CCNA

Yossef.E

2024-03

Table des matières

Introduction

Les cours de ce document sont extrait de la chaîne Youtube de Jeremy's IT Lab de son vrai nom Jeremy McDowell. Les cours etant très bien expliqués et réalisés ceux ci ont été retranscrit dans un format texte lisible traduit en Français. Cela permet une compréhension plus rapide et une mémorisation du contenu plus efficace, les commandes sont écrites et sont donc plus simple à copier. Voici le lien vers sa chaine Youtube : `https://www.youtube.com/@JeremysITLab`

Cours 1 : Appareils Réseaux

Ceci est un cours pour le CCNA nous allons voir tous les sujets qui concernent le CCNA.

Ce cours est dédié pour :

- Les personnes qui veulent passer l'examen du CCNA

- Les personnes qui veulent comprendre comment fonctionne un réseau

Nous allons comprendre d'abord les bases du réseau.

Donc qu'est ce qu'un Réseau ou Network en Anglais ?

« un réseau d'ordinateur est un réseau de télécommunication digital qui permet à des nœud de partager des ressources. »

Qu'est ce qu'un nœud ?

Pour comprendre nous allons prendre l'exemple de différents matériaux et de leurs symbole.

Voici un routeur :

Ceci est un Switch ou Commutateur :

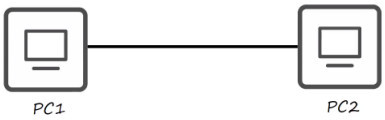
Ceci est un Firewall ou Pare feu :

Voici un serveur :

Un poste client :

Nous allons voir des exemples de différents types de réseaux.

Voici un premier exemple d'un réseau :



cela peut sembler être un petit réseau mais à partir du moment ou deux ordinateurs sont connectés en même temps cela constitue un réseau, et cela correspond bien à la définition d'un réseau qui indique qu'un réseau d'ordinateur permet de partager des ressources.

Un client peut être un téléphone, un ordinateur portable, un Mac, un ordinateur de bureau, etc..

Voici la définition d'un client :

« Un client est un appareil qui accède à un service rendu disponible par un serveur. »

Qu'est ce qu'un serveur ?

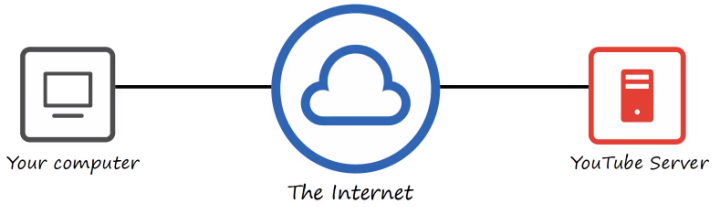
C'est tout simplement la définition inverse d'un client :

« Un appareil qui fournit des fonctions ou services pour des clients. »

Si l'on reprend l'exemple du réseau avec les deux PC on peut avoir le PC1 qui demande au PC2 de lui fournir des image jpg, le PC2 répond en fournissant ces images.

Ici le client est le PC1 et le PC2 est le serveur.

Voici un deuxième exemple d'un réseau avec un serveur et un client :



Le nuage est le symbole pour représenter Internet.

L'ordinateur demande la vidéo au serveur, le serveur Youtube envoie les données à travers le réseau.

Un exemple d'un réseau avec des IPhone :

L'IPhone qui fais la demande de vidéo est le client et l'autre est le serveur.

Il faut garder à l'esprit qu'un même appareil peut être un client dans certaines situation et un serveur dans d'autres situations.

Nous allons prendre un exemple plus large entre des clients se trouvant à New York et des serveurs se trouvant à Tokyo.

On ne connecte pas les clients directement entre eux la solution adapté est le Commutateur (ou Switch en Anglais) qui relie les appareils entre eux. Voici l'exemple d'un Switch qui permet de relier les clients et serveurs par ses interfaces. Voyons quelques caractéristiques de switchs :

- Les switchs ont plusieurs interfaces/ports réseau pour connecter les hôtes habituellement 24.

- Les switchs fournissent une connectivité pour les hôtes dans le même LAN (Local Area Network)

- Les switchs ne fournissent pas de connectivité entre des LAN différentes sur Internet.

Tous ces périphériques font partis du même réseau aussi appelé LAN pour Local Area Network.

Les switchs ne communiquent pas directement avec Internet pour partager leurs ressources entre LAN, il est utilisé pour cela le routeur qui permet le partage de ressources à travers Internet.

Voyons quelques caractéristiques des Switchs :

- Les routeurs ont moins d'interfaces réseau que les switchs.

- Les routeurs fournissent de la connectivité entre les LANs

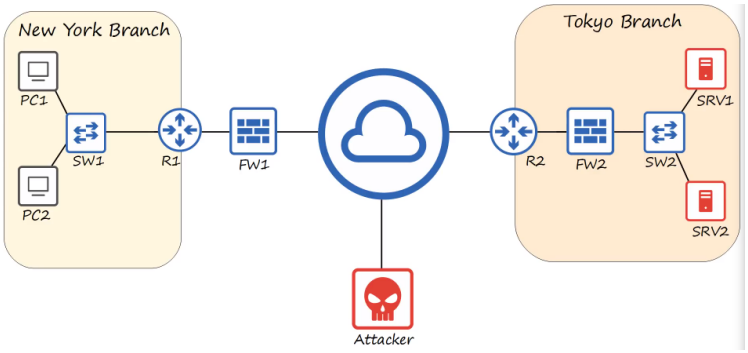
- Ils sont donc utilisés pour envoyer des données sur Internet.

Voici l'exemple du réseau schématisé :

Nous pouvons imaginer qu'il y ait un attaquant qui essaye de s'introduire dans notre réseau, la meilleure façon de se protéger contre cela est le Firewall.

Les Firewall sont spécialement conçus pour la sécurité du réseau et permettent de filtrer les entrées et sortis du réseau. Un Firewall peut être placé en dehors du réseau ou bien à l'intérieur du réseau.

Comme dans cette exemple :



L'essentiel est qu'ils protège les hôtes du dehors du réseau qui sont les appareils comme les ordinateurs.

Le Firewall est configuré avec des règles qui permettent d'autoriser quelle réseau sera permis et lequel sera renié. Il faut que les ordinateurs clients soient autorisés à passer le Firewall et puis que que les adresses externes qui pourraient être des attaquants soient bloqués en tentant de passer le Firewall.

Voici un exemple de Firewall :

Voici quelques caractéristiques des Firewalls :

- Il contrôlent et gèrent le trafique réseau basé sur les règles de configurés.

- Les Firewall peuvent être placés à l'intérieur et à l'extérieur du réseau

- Ils sont connus comme des pares feu de prochaine génération lorsqu'ils incluent des capacités de filtrage plus modernes.

Nous avons vu des "Firewall Hardware" qui filtrent le trafique entre les réseaux. Ce sont les Firewall dit "Réseaux".

Il existe aussi les "Host-Based Firewall" qui sont des applications qui filtrent le trafique entrant et sortant d'une machine hôte comme un ordinateur.

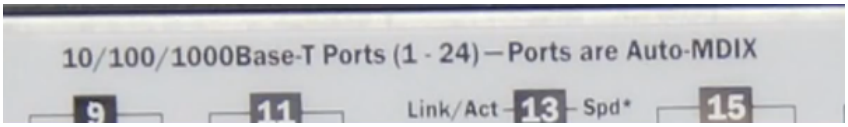
C'est une protection supplémentaire pour le réseau.

Cours 2 : Interfaces et Cables

Dans ce cours nous allons parler des interfaces et des câbles, comment connecter les appareils avec les câbles.

Cette photo montre les différentes interfaces d'un switch aussi appelés ports, c'est l'une des particularité des switchs, ils possèdent beaucoups d'interfaces/ports pour connecter les appareils.

Si l'on zoom sur l'écriture au dessus des interfaces on peut voir ceci que l'on expliquera plus tard :



Les interfaces sont appelés des ports RJ-45 (RJ = Registered Jack)

Un port RJ-45 ressemble à cela :

Ethernet est une collection de protocoles/standard réseau, dans ce cours nous verrons les types de câbles définis par les standard Ethernet.

Il y a différents types de ports et câbles pour différents types d'usages, ce sont des standard définis par l'Internet Protocol que toutes les marques suivent pour leurs appareils.

La connexion entre les appareils se fais à une certaine vitesse en Bits/secondes.

Qu'est ce qu'un Bits ?

L'ordinateur ne fonctionne que par des 0 et des 1 c'est à dire un langage Binaire que la machine interprète. Les données transmises à travers un câbles ne sont en fait que des données Binaires de 0 et de 1.

Qu'est ce qu'un Bytes (ou Octet en Français) ?

1 Byte (Octet) est égal à 8 bits soit huit 0 ou 1.

La vitesse est donc calculé en bits par seconde (Kbps, Mbps, Gbps, etc..), et non pas en Bytes par seconde.

Les données sur un Disque Dur par exemple sont mesurés en Bytes par secondes.

En résumé il y a :

- 1 Kilobit (Kb) = 1 000 bits

- 1 Megabit (Mb) = 1 000 000 bits

- 1 Gigabit (Gb) = 1 000 000 000 bits

- 1 Terabit (Tb) = 1 000 000 000 000 bits

En se qui concerne la vitesse on ne verra pas plus que la vitesse en Gbps

Après les Terabit il existe aussi les petabit, exabit, zettabit et yottabit, etc...

Les standard Ethernet sont définis dans le IEEE 802.3 standard en 1983

IEEE = Institute of Electrical and Electronics Engineers

Voici un tableau des standards qui proviennent du IEEE :

Vitesse	Nom commun	IEEE Standard	Nom informel	Longueur Maximale
10 Mbps	Ethernet	802.3i	10BASE-T	100m
100 Mbps	Fast Ethernet	802.3u	100BASE-T	100m
1 Gbps	Gigabit Ethernet	802.3ab	1000BASE-T	100m
10 Gbps	10 Gig Ethernet	802.3an	10GBASE-T	100m

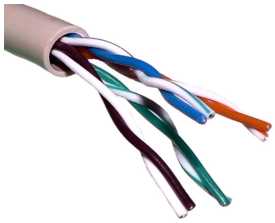
BASE se réfère au bandes de signalement. Le T est pour pair croisé.

Les câbles Ethernet utilisés sont des câbles UTP pour : Unshielded, Twisted, Pair.

"Unshielded" pour dire qu'il ne possède pas de protection en métal contre les interférences.

"Twisted" (croisés) puisque comme on le voit dans la photo les câbles sont croisés entre eux se qui permet de protéger contre les EMI (Electromagnetic Interference ou Interférence Magnétique en Français)

"Pair" se sont des pairs de câbles croisés entre eux.



Sur cette photo on peut voir qu'il y a 8 Pins qui correspondent aux 8 câbles de la photo précédente.

Ce ne sont pas tous les standards qui utilisent les 8 files :

10BASE-T et 100BASE-T utilisent 2 pairs (ou 4 files)

1000BASE-T et 10GBASE-T utilisent eux 4 pairs (ou 8 files)

Disons que l'on veut connecter un câble de standard 10BASE-T ou 100BASE-T entre un pc et un switch, on ne va utiliser que 2 pairs :

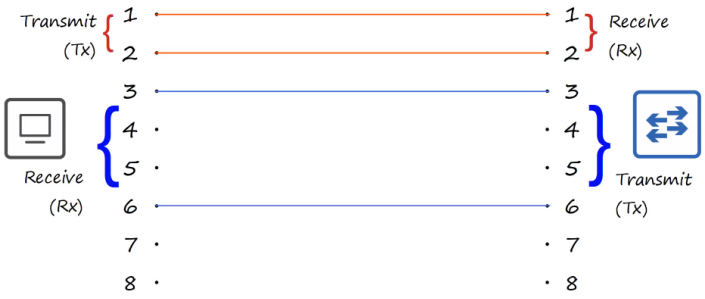
Pour la première pair les Pins vont aller du pin 1 vers le pin 1, et du pin 2 vers le pin 2.

Cette première pair est utilisé par l'ordinateur pour transmettre les données et par le switch pour réceptionner les données.

Pour la deuxième pair les Pin vont aller du pin 3 vers le pin 3 et du pin 6 vers le pin 6.

Cette seconde pair est utilisé par le switch pour transmettre les données et par l'ordinateur pour réceptionner les données.

C'est ce que l'on appelle une transmission Full-Duplex pour dire que les deux appareils peuvent recevoir et envoyer des données en même temps sans qu'il y ait des problème de collision.



La transmission des données ce fais de la même manière qu'entre un routeur et un switch, c'est à dire qu'il transmet les données avec les pins 1 et 2 et reçoit les données sur les ports 3 et 6.

Puisqu'ils reçoivent les données sur les même pin il n'y a pas vraiment de problème, ces câbles sont appelés : "Câbles droits" car les pins d'un sens se connectent directement au pin de l'autre sens.

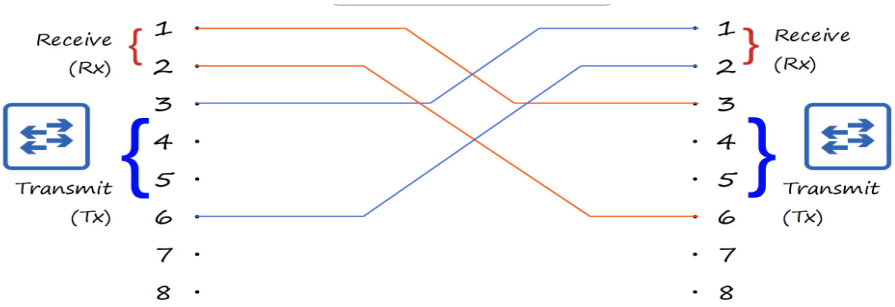
Mais que se passe t-il si l'on veut connecter deux matériaux les mêmes par exemple, un pc avec un pc, un switch avec un switch, un routeur avec un autre routeur ?

Cela ne fonctionnera pas avec des câble droits car pour les deux routeurs, les pins 1 et 2 ne font que transmettre les données pour les deux routeurs or il faut qu'il y ait d'un sens un transmetteur et puis de l'autre sens un récepteur.

Pour ce type de connexion il existe un autre type de câbles : les "câble croisés"

Avec ce type de câbles les pin sont inversés donc le pin 1 se connecte au pin 3 et le pin 2 au pin 6.

Le pin 3 d'un sens se connecte au pin 1 de l'autre appareil et le pin 6 au pin 2.



Si l'on connecte ce câble croisé entre un routeur et un ordinateur ils ne pourront pas recevoir et transmettre de données car les pin sont inversés.

Type d'appareils	Transmit (Tx) Pins	Receive (Rx) Pins
Router	1 et 2	3 et 6
Firewall	1 et 2	3 et 6
PC	1 et 2	3 et 6
Switch	3 et 6	1 et 2

Mais si les switchs reçoivent réellement leurs données avec les pins comme indiqué cela voudrait dire qu'il faut utiliser un câble croisé pour connecter un switch avec un autre switch ?

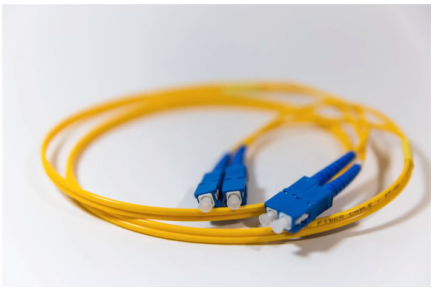
C'est pour cela qu'existe la fonction Auto MDI-X qui détecte quelles sont les pin qui reçoivent et ceux qui transmettent les données et adapte le fonctionnement pour que se soit adapté afin de pouvoir utiliser même des câbles normaux entre deux switchs.

Nous avons parlés des 10BASE-T et 100BASE-T mais qu'en est t-il des 1000BASE-T et 10GBASE-T ?

pour ce type de câble il y a les pin 4 et 5 qui sont connectés entre eux et les 7 et 8 entre eux.

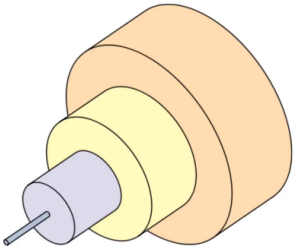
Une autre particularité est que chaque pair est bidirectionnel c'est à dire qu'elle peut recevoir et transmettre des données ce qui la rend plus rapide.

Dans certains Switchs peut se trouver des interfaces qui servent à insérer des SFP Transceiver (Small Form-Factor Pluggable) ce type de port sert à connecter des câble de fibre optique.



Il y a deux câbles un pour transmettre les données et le deuxième pour recevoir.

Les câbles de fibre optique ont différentes couches à l'intérieur :



1 : le verre de la fibre

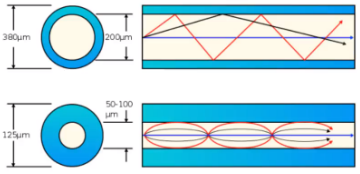
2 : une surface qui fais refléter la lumière contenu dans le câble

3 : Un espace de protection pour empêcher le verre de se casser

4 : le plastique au dessus du câble

Il existe deux modes pour les câbles de la fibre :

La fibre multimode :



Quelques caractéristiques de ce mode :

Le diamètre du verre est plus épais que pour le monomode. Permet que plusieurs angles pour la lumière d'entrer dans la couche de verre. Permet l'utilisation de câbles à plus longue distance que les UTP mais cela reste plus court que le mode fibre monomode.

Ils sont moins chère que les cables fibre monomode.

Voici l'exemple d'un câble monomode :

Quelques caractéristiques de ce mode :

Le diamètre du verre est plus réduit que pour le multimode.

La lumière entre par un seul angle depuis le transmetteur.

Il permet d'avoir de plus long câbles par rapport à UTP ou les cables fibres multimode.

Plus chère que le multimode à cause du laser-based SFP transmetters.

Nom Informel	IEEE Standard	Vitesse	Type de Câble	Longueur Maximal
1000BASE-LX	802.3z	1Gbps	Multimode ou Single-Mode	550m (MM) 5 km (SM)
10GBASE-SR	802.3ae	10Gbps	Multimode	400m
10GBASE-LR	802.ae	10Gbps	Sigle-Mode	10 km
10GBASE-ER	802.3ae	10Gbps	Sigle-Mode	30 km

Comparons à présent un câble UTP et un câble de fibre optique :

UTP :

- Les coûts sont moins chère que pour un câble de fibre optique

- Une distance moins grande que pour un câble de fibre optique (à peu près 100m)

- peut être vulnérable aux EMI (Electromagnetic Interference)

- Ports RJ45 utilisés avec UTP sont moins chère que pour un port SFP

- Il peut y avoir des fuite de données qui peuvent être copiés (risque de sécurité)

Fibre optique :

- Plus chère qu'un câble UTP

- Une longueur maximal plus grande que pour un câble UTP

- Pas de vulnérabilité aux EMI

- Des ports SFP sont plus chères que les ports RJ45

- N'émet pas de signal il n'y a donc pas de risque de vol de données (pas de risque de sécurité)

Cours 3 : Modèle OSI et TCP/IP

Ce cours est à propos des modèle de réseaux, le premier est le modèle OSI, le second le TCP/IP.

Tout d'abord, qu'est ce qu'un modèle de réseau ?

Les modèles de réseau catégorisent et fournissent une structure pour les protocoles de réseau et les standards.

Un protocole est un nombre de règles qui définit comment le réseau des périphériques et software devrait fonctionner.

Que se passerait il s'il n'y avait pas de standardisation des protocoles ?

Imaginons qu'il y ait plusieurs ordinateurs de la marque Dell dans une entreprise et puis des Mac dans une autre entreprise, les deux entreprises ne pourraient pas échanger puisque se sont des systèmes différents et ces deux systèmes parleraient deux langages différents.

Le modèle OSI est l'acronyme de « Open Systems Interconnection »

Il s'agit d'un modèle conceptuel qui catégorise et standardise les différentes fonctions d'un réseau.

Crée par l'« International Organisation for Standardization » (ISO)

Les fonctions sont divisés en 7 couches qui sont :

7. Application : Cette couche est la plus proche de l'utilisateur puisqu'elle interagit avec les applications comme par exemple : un navigateur web (Brave, Chrome, Firefox)

HTTP et HTTPS sont des protocoles de la couche 7.

L'application n'appartient pas à la couche réseau ce n'est que le protocole associé à cette application, pour un navigateur web : HTTP ou HTTPS

Les fonctions de la couche 7 incluent :

- Identification des partenaires de la communication

- Synchronisation des communications

Fonctionnement du modèle OSI : Lorsqu'un utilisateur interagit avec son système il utilise directement la couche 7 de l'application, l'information demandé sur cette couche va redescendre une à une les sept couches du modèle OSI jusqu'à la couche Physique puis arriver vers l'autre système dans lequel elle va recevoir l'information à partir de la couche 1 (Physique) puis remonter les 7 couches jusqu'à la couche 7 (Application).

Ce processus est appelé l'encapsulation puisque les informations sont encapsulés une à une en s'ajoutant en fonction de la couche réseau.

Lorsque le système reçoit l'information et remonte les couches une à une cela s'appelle la dé-encapsulation.

6. Présentation : Les données contenus dans la couche Application sont dans un format d'application. Il est nécessaire qu'elle soient traduites en un format différent pour être envoyé à travers le réseau. Le rôle de la couche de présentation est de traduire entre l'application le format du réseau. Par exemple, le cryptage de données lorsqu'il est envoyé, et le décryptage de données lorsqu'il est reçu.

Il traduit aussi entre différents formats de couches Application.

5. Session : Il contrôle les dialogues de session entre les hôtes de communication. Il établit gère et termine les connexions entre l'application local (par exemple : le navigateur web) et l'application à distance (par exemple : Youtube)

Les ingénieurs réseaux ne travaillent pas couvent avec les applications des couches : Application Présentation, Session. C'est le travail des développeurs d'application, qui font connecter leurs applications au réseau.

Lorsque les données traverse le modèle OSI elle traversent les couches : 7, 6 et 5 puis arrivé à la couche 4 elles ajoutent une entête L4. À ce stade de la transmission des données la donnée est appelé segment.

4. Transport : Segmente et réassemble les données pour la communication entre les hôtes.

Il casse de grosses part de pièce de données en de petits segments qui pourront être envoyés plus rapidement sur le réseau et sont moins sujet à provoquer des problèmes de transmission si une erreur apparaît. Il fournit la communication hôte à hôte (ou host-host)

Une fois la couche transport passé une nouvelle entête est ajouté qui correspond à la couche 3, à ce stade de l'encapsulation les données avec les deux entête est appelé un paquet. L'adresse IP est inclus dans cette entête.

3. Réseau : Fournit une connectivité entre l'hôte sur différents réseaux en dehors du LAN. Il fournit Un adressage logique (Adresse IP). Il fournit un chemin de sélection entre la source et la destination. Les routeurs fonctionnent à la couche 3.

Une fois la couche réseau passé une nouvelle entête est ajouté à la donnée transmise qui correspond à l'entête de la couche 2. A ce stade de l'encapsulation les données avec les 3 entêtes sont appelé une frame.

2. Liaison : Fournit une connectivité nœud à nœud et un transfert des données (par exemple, un PC vers un ordinateur, un switch vers un routeur, un routeur vers un routeur)

Il définit comment la donnée est formaté pour la transmission à travers le câble physique (par exemple, un câble UTP), Il détecte et corrige les erreurs probable de la couche physique.

Il utilise la couche 2 séparé de l'adressage de la couche 3.

Les switchs fonctionnent à la couche 2.

1. Physique : Définis les caractéristiques physique du moyen utilisé pour transféré les données entre les périphériques. Par exemple le niveau de voltage, la distance de transmission, les connecteurs physique, la spécification du câble, etc.

Les bits digitaux sont convertis en signaux électrique (pour une connexion cablé) ou radio (Pour une connexion sans fil).

Les informations du cours 2 sont inscrites dans la couche physique (Le câble, le pin, layouts, etc..)

Une fois toutes les couches traversés avec les entêtes ajoutés, il y a le processus de dé-encapsulation qui se passe sur l'autre système, les entêtes sont retirés une à une en fonction de la couche traversé jusqu'à atteindre les couches 5, 6 et 7 pour que les données soient visibles par le système.

Les différents termes associés aux différents ajouts des entêtes avec : La donnée, le segment, Le paquet, le Frame sont appelés des Protocol Data Units (PDUs)

Voyons à présent comment fonctionne le modèle TCP/IP.

Comme le modèle OSI, il s'agit un modèle conceptuel et de certaines utilisation de protocoles de communication utilisé sur internet et d'autres réseaux. Il est connu comme TCP/IP car ce sont deux protocoles fondamentaux dans la suite. Il a été développé par le United States Department of Defense par la DARPA (Defense Advanced Research Projects Agency)

Il a une structure similaire au modèle OSI mais avec moins de couches. C'est le modèle actuellement utilisé dans les réseaux modernes. Il est à noter que le modèle OSI a une influence sur comment les ingénieurs réseau pensent et parlent à propos du réseau.

Le modèle TCP/IP n'est constitué que de 4 couches :

Les 3 dernières couches du modèle OSI : Application, Présentation, session sont remplacé par une seule et même couche appelé : Application

les couches transport et réseau sont les même que pour le modèle OSI

Les 2 premières couches : Liaison et Physique sont fusionnée pour ne faire qu'une couche la couche : Liaison

Se qui fais donc en tout 4 couches :

1. Liaison – 2. Internet – 3. Liaison – 4. Application.

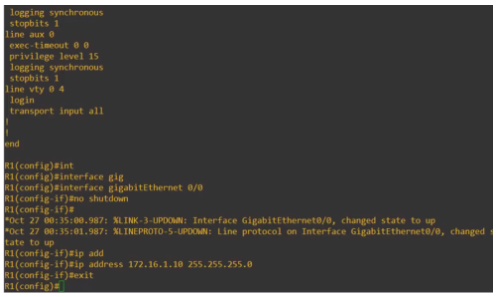
Cours 4 : Introduction au CLI de Cisco

Dans ce cours nous allons apprendre comment fonctionne le CLI sur l'IOS Cisco qui est le système d'exploitation de Cisco.

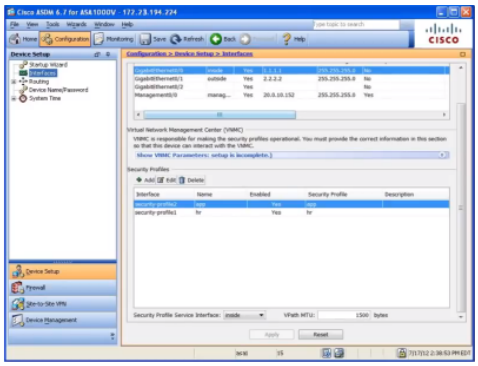
Qu'est ce qu'un CLI ?

C'est l'acronyme de Command-Line interface, c'est l'interface que l'on utilise pour configurer les appareils Cisco.

Voici une image de ce à quoi pourrait correspondre un CLI :

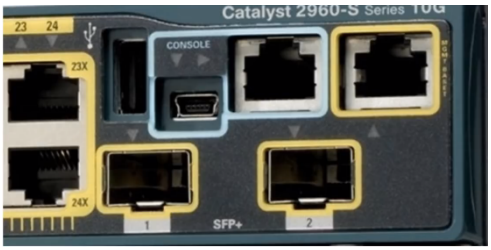


Il y a aussi le GUI qui est l'acronyme de Graphical User Interface pour que l'utilisateur ait une interface graphique :



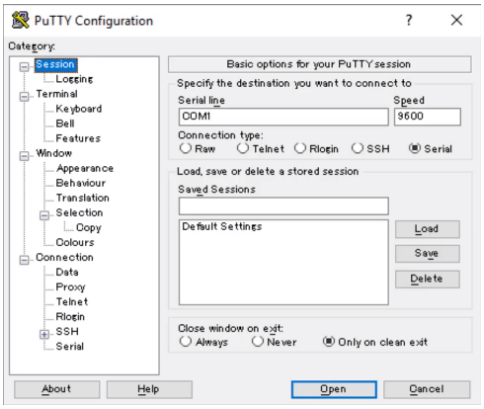
Comment connecter à un appareil Cisco pour le configurer avec le CLI ?

Il existe plusieurs méthode mais l'une d'entre elle est d'utiliser le port console qui est composé d'un port RJ45 et d'un port USB Mini-B comme sur cette photo :



Il faut utiliser ce type de câble qui est un câble console ou rollover cable :

Une fois que l'on connecté son ordinateur à l'appareil Cisco pour accéder au Terminal on peut utiliser l'application Putty. Voici une photo de l'interface utilisateur :



Une fois connecté à l'appareil on aura une interface comme celle ci :

:

```
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

Cisco CISCO2911/K9 (revision 1.0) with 491520K/32768K bytes of memory.
Processor board ID FTX152400KS
3 Gigabit Ethernet interfaces
DRAM configuration is 64 bits wide with parity disabled.
255K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)


         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!


Router>
```

Lors de la première connexion on est dans le mode user EXEC mode qui est caractérisé par le symbole « >» à côté du nom d'hôte.

Comme on peut le voir sur la photo précédente le nom utilisateur est Router avec le mode user EXEC mode : « >»

Le mode User EXEC mode est très limité, car les utilisateurs peuvent voir certaines choses mais ne peuvent faire aucun changement à la configuration.

Ce mode est aussi appelé le « User Mode ».

Lorsque l'on entre la commande :

```
Router>enable
```

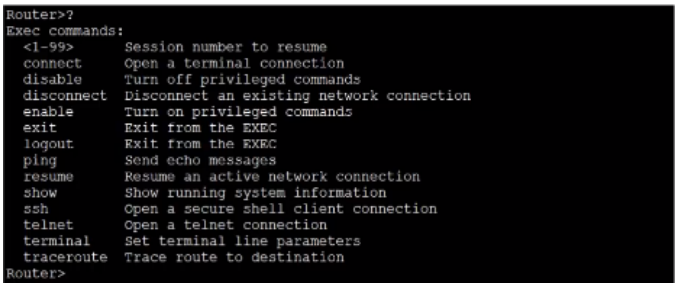
on entre dans le privileged EXEC mode sur ce mode le symbole « # » apparaît comme ceci :

```
Router#
```

Ce mode donne un accès complet pour voir la configuration de l'appareil, relancer l'appareil, etc..

Ce mode ne change pas la configuration, mais peut changer la date sur l'appareil, sauvegarder la configuration, etc..

Voici une liste des commandes en user EXEC mode à gauche et en Privileged EXEC Mode à droite :



Sur les routeurs Cisco il existe une fonction pour auto-compléter les commandes en appuyant sur la commande Tab, par exemple en appuyant sur Tab après avoir écrit « en » la commande va s'auto compléter en « enable ».

Une autre astuce est que sur les Cisco IOS CLI il n'est même pas nécessaire de compléter la commande pour qu'elle fonctionne, par exemple pour écrire la commande « enable », simplement d'écrire « en » suffit pour que la commande soit exécuté. Cela ne fonctionne seulement si c'est la seule commande qui est possible et commençant par les termes inscrit sinon ça ne fonctionne pas.

Il y a la possibilité de faire afficher les commandes disponibles en ajoutant un « ? ». Par exemple si je veux connaître les commandes disponible qui commencent avec un « e » je peux taper la commande « e ? » et les commandes enable, exit apparaissent pour indiquer que se sont les deux commandes possible commençant par un « e ».

Pour entrer dans le mode de changement de la configuration il faut entrer dans le mode Global Configuration Mode avec les commandes :

```
Router>enable
Router#configure terminal
Router(config)#
```

Une abréviation pour cette commande est « conf t » pour « configure terminal ».

On peut protéger le privileged EXEC Mode avec un mot de passe comme ceci :

```
Router(config)#enable password CCNA
```

Ici le mot de passe pour entrer dans le mode privileged EXEC Mode est « CCNA »

On peut tester tout de suite le mot de passe en quittant le mode Configuration en tapant la commande :

```
Router(config)#exit
Router#exit
```

Lorsque l'on essaye d'entrer dans le User Mode, le mot de passe nous est demandé une fois le mot de passe indiqué on peut entrer dans le mode configuration :

```
Router>enable
Password:
Router#
```

Il y a deux fichier de configuration gardés dans l'appareil en même temps.

Il y a le « Running-Config » qui correspond à la configuration active sur l'appareil. Lorsque l'on entre une commande dans le CLI, on affiche une configuration active.

Il y a aussi le « Startup-config » qui correspond à la configuration qui est chargé lorsque l'on redémarre l'appareil.

On utilise la commande :

```
Router#show running-config
```

Pour afficher le fichier de configuration en cours.

Pour afficher la configuration de lancement on utilise :

```
Router#show startup-config
```

Il faut sauvegarder la configuration pour que celle ci soit affichée à chaque redémarrage sinon c'est la configuration de la Running config qui sera chargée.

Il y a trois manière de sauvegarder la configuration « running configuration » pour qu'elle soit sauvegardé dans la « startup configuration » voici les 3 commandes possibles :

```
Router#write
Router#write memory
Router#copy running-config startup-config
```

Avec la commande : « show startup config » on a pu voir qu'il était possible de voir la configuration actuelle et les commandes qui étaient active, ce qui signifie qu'il est possible de voir le mot de passe rien qu'en tapant cette commande, ceci peut présenter des risques de sécurité, c'est pour cela qu'il est possible de crypter le mot de passe en ajoutant la commande :

```
Router#conf t
Router(config)#service password-encryption
```

à présent lorsque l'on tape la commande : « show running-config » il n'est plus possible de voir le mot de passe en clair, il est crypté avec un algorithme de Cisco, mais reste possible à craquer.

Pour plus de sécurité on peut utiliser la commande :

```
Router(config)#enable secret Cisco
```

Avec cette dernière commande le mot de passe sera crypté avec un cryptage MD5 qui est plus sécurisé.

Si cette commande est activé Ce n'est que le cryptage MD5 qui sera valide pour mot de passe le mot de passe crypté par Cisco avec l'indicatif 7 ne sera pas valide.

A présent voyons comment annuler une commande que l'on a ajouté en plus, cela se fais en ajoutant devant la commande le terme : « no »

par exemple si l'on tape la commande :

```
Router(config)#no service password-encryption
```

Les mots de passe que l'on ajoutera ne serons plus cryptés.

Cours 5 : Ethernet LAN switching (Partie 1)

Dans ce cours nous allons apprendre se qu'est l'Ethernet LAN Switching, c'est à dire comment fonctionne la connectivité dans un réseau local en LAN (Local Area Network).

Tout d'abord revoyons se que nous avons vu précédemment sur le modèle OSI.

Le modèle OSI définit les caractéristiques du moyen utilisé pour transférer des données entre appareils.

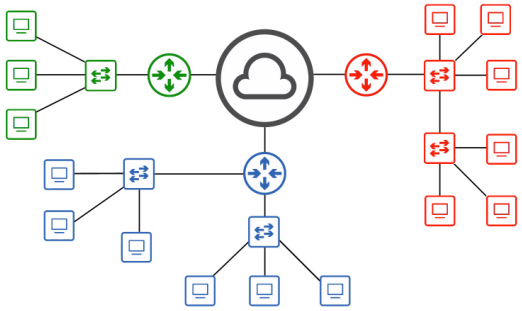
Par exemple avec la couche physique c'est le niveau de voltage, la distance maximal de transmission, la spécification du câble, etc.. Les Bits sont convertis en signaux électriques pour des connexion filaire et en signaux radio pour une connexion sans fil. Les informations données sur le cours 2 sont en rapport avec la couche physique.

Pour la couche 2 Liaison cela fournit une connectivité de transfert de données (par exemple, PC à switch, switch à routeur, routeur à routeur). Il définit comment les données sont formatées pour la transmission à travers un moyen physique (par exemple un câble UTP, etc..) Il détecte et corrige les erreurs de la couche physique. Il utilise une couche 2 d'adressage séparé de la couche 3 d'adressage.

Les switchs fonctionnent à la couche 2.

Dans ce cours nous verrons surtout la couche 2 qui est en rapport avec les switchs puis plus tard la couche 3 en rapport avec les routeurs.

Voici un schéma de se que pourrait être un LAN :

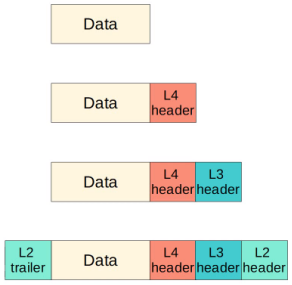


Nous allons apprendre petit à petit se qu'est le réseau mais avec une définition simple se pourrait être un réseau contenu dans une petite zone. Par exemple une réseau de la maison ou d'une entreprise. Les routeurs sont utilisés pour connecter des LAN séparés par exemple sur le schéma précédent on peut voir 4 différentes LAN liés aux 3 différents routeurs.

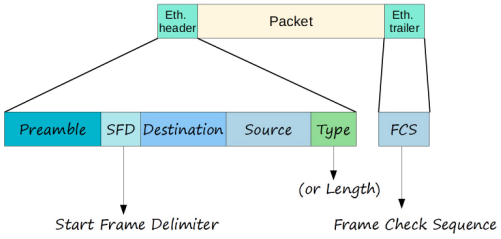
Le routeur vert constitue 1 LAN le rouge un autre LAN et il y a 2 LAN qui constituent le routeurs bleue puisqu'il y a 2 switchs connectés au routeur sur deux interfaces différentes pour 2 différents LAN.

Nous avons aussi vu précédemment qu'il y avait 5 parties dans un PDU (Protocol Data Units)

La donnée, le Segement, Le paquet, la Frame.



Voici le contenu d'un Ethernet Frame :



Le préambule est composé ainsi :

- Une longueur de 7 bytes (56 bits)

- Alterne entre les 1 et 0 ce qui fait 01010101 * 7

- Il permet aux appareils de synchroniser leurs horloges.

Le SFD est composé de :

- Acronyme de « Start Frame Delimiter »

- D'une longueur de 1 byte (8 bits)

- 10101011

- il marque la fin du préambule et commence le reste du Frame.

La destination et source ont le rôle de :

- Indiquer à l'appareil qui envoie et réceptionne le frame.

- Consiste à la destination et la source l'« adresse MAC »

- MAC est l'acronyme de Media Access Control

- L'adresse MAC est de 6 byte (48-bit) qui est l'adresse d'un appareil physique

Le type ou la longueur du field sont de :

- 2 byte (16 bit) de longueur

Il peuvent être utilisés pour indiquer le type du paquet encapsulé ou bien la longueur du paquet encapsulé.

- Une valeur de 1500 ou moins dans ce paquet indique la longueur du paquet encapsulé (en bytes)

- Une valeur de 1536 ou plus indique le Type de paquet encapsulés et la longueur est déterminé par d'autres méthodes.

Par exemple un IPv4 = 0x0800 (hexadécimal) est égal à 2048 en décimal.

Une Ipv6 = 0x86DD (hexadécimal) est égal à 34525 en décimal.

Donc en résumé les longueurs en bytes pour chaque partie de l'entête Ethernet sont égal à :

Préambule = 7 bytes

SFD = 1 bytes

Destination = 6 bytes

Source = 6 bytes

Type = 2 byte

Voyons à présent la partie présente dans le Ethernet Trailer :

Il est composé du FCS qui est l'acronyme de « Frame Check Sequence »

- Il est constitué de 4 bytes (32bits) de longueur

- Il détecte les données corrompus en lançant un algorithme « CRC » au travers des données reçus.

- CRC est l'acronyme de « Cyclic Redundancy Check »

L'Ethernet Frame fais donc au total 26 bytes lorsque l'on inclut l'entête Ethernet et Ethernet Trailer.

Voyons plus précisément de quoi est composé une adresse MAC :

- 6 byte (48 bit) d'adresse physique assigné à l'appareil lorsqu'il est conçu

- A.K.A « Burned-In Address » (BIA)

- L'adresse est unique c'est à dire qu'il n'y en a pas deux identique dans le monde.

- Les 3 premiers bytes sont le OUI (Organizationally Unique Identifier) qui est assigné à l'entreprise qui fabrique l'appareil.

- Les 3 derniers bytes sont unique à l'appareil lui même

- Il est écrit en 12 caractères hexadécimal

Le décimal est composé de 10 chiffres possibles : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

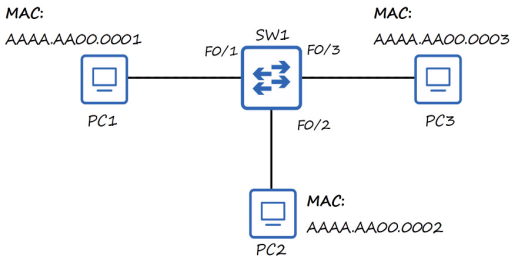
L'Hexadécimal utilise 16 chiffres possibles : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Ici le A vaut 10 le B : 11, le C : 12, le D : 13, le E : 14, et le F : 15

Voici un tableau qui compte le décimal et l'Hexadécimal jusqu'à 31 pour mieux comprendre :

DEC.	HEX.	DEC.	HEX.	DEC.	HEX.	DEC.	HEX.
0	0	8	8	16	10	24	18
1	1	9	9	17	11	25	19
2	2	10	A	18	12	26	1A
3	3	11	B	19	13	27	1B
4	4	12	C	20	14	28	1C
5	5	13	D	21	15	29	1D
6	6	14	E	22	16	30	1E
7	7	15	F	23	17	31	1F

Voyons à présent un exemple dans lequel sont distribué des adresses MAC dans un réseau :



Un Frame Unicast est destiné pour aller vers une seule destination

Imaginons que le PC1 veut envoyer une donnée au PC2, la donnée va passer par le switch qui va ajouter son adresse MAC en mémoire et l'ajoute à sa table d'adresse MAC ce qui est connus comme le Dynamic MAC Address puisque c'est le switch qui l'apprend lui même.

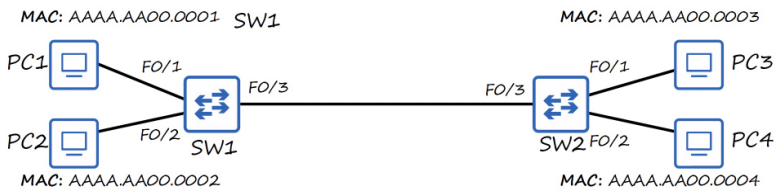
Le switch ne connais pas l'adresse de destination de l'adresse MAC du PC2 c'est pour cela qu'il va « inonder » ses interfaces excepté l'interface source, les interfaces reçoivent donc les deux le paquet de destination mais le PC3 va ignorer le paquet puisque son adresse MAC n'est pas la même que l'adresse demandé de destination.

Le PC2 quant à lui reçoit le paquet et l'utilise.

Pour que l'adresse MAC du PC2 soit ajouté à la table d'adressage MAC du switch il faut que le PC2 envoie un frame vers un autre PC.

Il est à noter que les adresses MAC sont retirés après 5 minutes d'inactivité sur les switchs Cisco

A présent voyons se qu'il se passe avec 2 switch qui communiquent entre eux comme sur ce schéma :



Imaginons que le PC1 veut envoyer des données au PC3, le même processus ce passe : le switch1 reçoit le frame et ajoute l'adresse MAC à sa table d'adressage après cela il va inonder ses interfaces excepté l'interface de réception donc ici les interface F0/2 et F0/3, le PC2 reçoit le frame mais ne le lit pas car l'adresse de destination pas à son adresse MAC. Le Switch 2 reçoit le frame et ajoute lui aussi l'adresse MAC du PC1 à sa table d'adressage MAC mais correspondant à l'interface F0/3.

Le Switch 2 va inonder lui aussi ses interfaces donc ici les interfaces F0/1 et F0/2 pour distribuer le frame, le PC4 ne lit pas la frame car ça ne correspond pas à son adresse MAC, mais le PC3 lit la trame car cela correspond à son adresse MAC.

C'est ainsi que les données sont distribuées.

Cours 6 : Ethernet LAN Switching (Partie 2)

Dans ce cours nous allons apprendre comment fonctionne le réseau localemement dans un LAN (Local Area Network). Le préambule et le SFD ne sont pas considérés comme faisant partie de l'entête Ethernet.

De plus la taille de l'entête Ethernet avec le trailer est de 18 bytes (6 + 6 + 2 + 4)

Il y a aussi une taille minimal pour un frame Ethernet (Entête + Payload + Trailer) qui est de 64 bytes.

Les 64 bytes – 18 bytes (header + taille du trailer) = 46 bytes

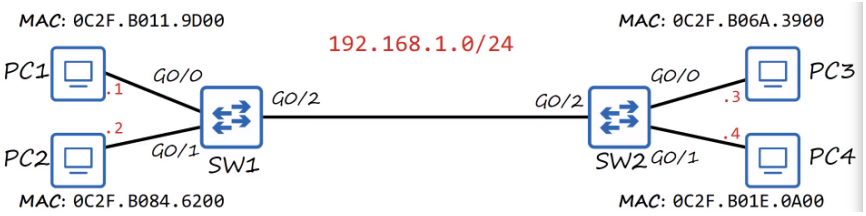
La taille minimal pour la taille d'un payload (paquet) est de 46 bytes.

Si le payload est de moins de 46 bytes d'autres bytes sont ajoutés.

Par exemple si l'on a 34 byte de paquet + 12 byte de padding = 46 bytes.

Dans le schéma suivant nous allons assigner des adresses IP aux différents PC sur l'adresse réseau :

192.168.1.0/24



Lorsqu'un appareil veut communiquer avec un autre appareil il n'inclut pas seulement son adresse MAC comme on a pu le voir dans le cours précédent.

Dans l'entête sera inclus l'adresse IP source, l'adresse IP de destination, l'adresse MAC source mais le PC1 ne connais actuellement pas l'adresse MAC de destination.

Lorsque l'on veut envoyer une donnée c'est l'adresse IP qui est indiquée.

Imaginons que le PC1 veut envoyer une donnée vers le PC3, pour se faire il doit tout d'abord découvrir l'adresse MAC du PC correspondant à la destination. Pour se faire il va utiliser le protocole ARP (Address Resolution Protocol) qui va découvrir les adresses MAC présente sur le réseau.

- ARP est l'acronyme de « Address Resolution Protocol »

- il est utilisé pour découvrir la couche 2 (l'adresse MAC) à partir d'une adresse de couche 3 connu (l'adresse IP)

- Le protocole consiste en 2 messages : la requête ARP et la réponse ARP.

- La requête ARP est envoyé en Broadcast (envoyé à tous les hôtes sur le réseau)

- La réponse ARP est envoyé en Unicast (envoyé seulement à un seule hôte, celui qui envoie la requête)

L'adresse utilisé en Broadcast pour connaître les adresse MAC est : FFFF.FFFF.FFFF

PC1 envoie la requête ARP au SW1, le SW1 ajoute l'adresse à la table d'adresse MAC, de cette manière l'adresse MAC est apprise dynamiquement. La requête ARP est envoyé au interface, le PC2 reçoit la requête mais puisque la requête ne lui est pas adressé il n'accepte pas la requête.

Le SW2 reçoit aussi la requête et ajoute l'adresse MAC du PC1 à sa table d'adresse MAC.

Le SW2 redistribue la requête sur ses interfaces, le PC4 reçoit la requête mais ne l'accepte pas, le PC3 accepte la requête puisque c'est son adresse IP qui correspond à la requête.

Une fois la requête reçu le PC3 va renvoyer une réponse au PC1 en Unicast pour confirmer la réception. La réponse va s'ajouter aux table d'adresse MAC du SW2 et du SW1, la réponse est ensuite reçu pas le PC1.

On peut utiliser la commande *arp -a* pour voir la table ARP (Windows, MacOS, Linux)

```
C:\Users\user>arp -a

Interface: 169.254.146.29 --- 0x9
  Internet Address      Physical Address      Type
  169.254.255.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.0.167 --- 0xd
  Internet Address      Physical Address      Type
  192.168.0.1           98-da-c4-dd-a8-e4     dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

- L'adresse Internet est l'adresse IP (l'adresse de couche 3)

- L'adresse Physique est l'adresse MAC (l'adresse de couche 2)

- Si le Type est en statique cela signifie que c'est une entrée par défaut.

- Si le Type est en dynamic cela signifie qu'il est appris par ARP

Il est possible d'utiliser un logiciel pour simuler le réseau appelé GNS3 mais aussi un logiciel Cisco appelé Packet Tracer plus simple d'utilisation.

A présent que l'adresse du PC1 est connu il est possible de tester la connexion en lançant une requête ping.

Le ping est très utile il est utilisé pour tester la connectivité.

- Il mesure le temps que la requête prend pour être réceptionné

- Il utilise deux messages : Une requête Echo ICMP, Une réponse Echo ICMP

C'est similaire au requêtes et réponses ARP

- La commande pour utiliser ping est : *ping (ip address)*

voici la capture d'écran d'une requête ping :

```
PC1#
PC1#ping 192.168.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 20/20/22 ms
PC1#
```

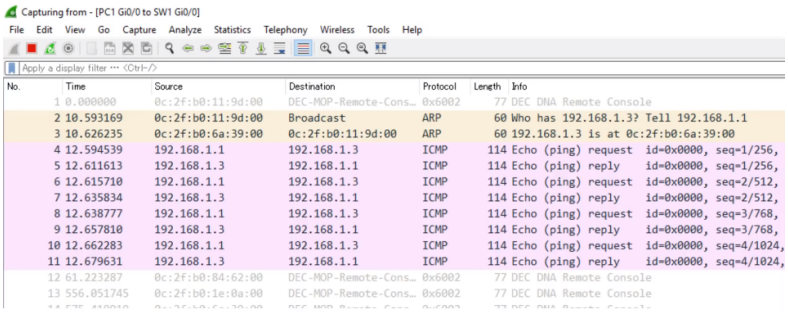
Pourquoi la y a t-il une requête qui n'a pas fonctionné sur les 5 envoyés ?

C'est parce que la première requête était une requête ARP pour connaître l'adresse MAC

La commande pour voir la table ARP sur le switch est *show arp*

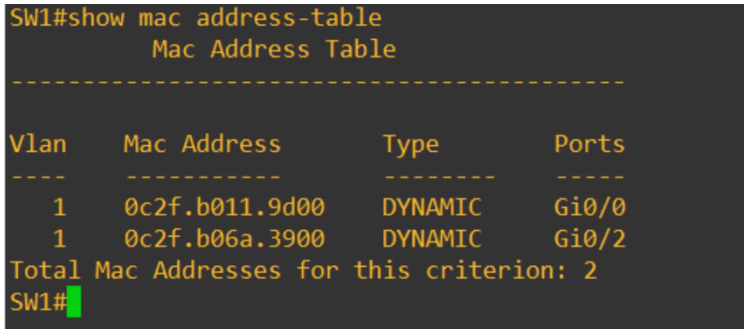
```
PC1#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  192.168.1.1              -   0c2f.b011.9d00  ARPA   GigabitEthernet0/0
Internet  192.168.1.3             34   0c2f.b06a.3900  ARPA   GigabitEthernet0/0
PC1#
```

Voici une capture d'écran sur Wireshark sur laquelle on peut voir la requête ARP envoyé avec la réponse ARP.



Voici à présent la commande pour voir la table d'adresse MAC sur un Switch Cisco :

```
SW1#show mac address-table
```



Comme on peut le voir il y a la section Vlan pour indiquer le numéro de Vlan, l'adresse MAC, le type, et le numéro de port/interface

Les adresses MAC se retirent automatiquement après 5 minutes d'inactivité c'est aussi appelé « aging ».

Il est aussi possible de supprimer manuellement l'adresse MAC de la table en utilisant la commande :

```
SW1#clear mac address-table dynamic
```

ou bien si l'on veut supprimer une seule adresse MAC :

```
SW1#clear mac address-table dynamic address 0c2f.b011.9d00
```

Il est aussi possible de supprimer les adresses MAC se trouvant sur une seule interface :

```
SW1#clear mac address-table dynamic interface Gi0/0
```

Cours 7 : Adressage IPv4 (Partie 1)

Dans les cours précédent nous avons vu comment fonctionnait un réseau local avec les PC connectés aux switchs. Dans ce cours nous allons voir comment le trafique réseau est orienté en dehors du LAN à l'extérieur du réseau local vers d'autres LAN.

Pour cela la couche 3 est utilisé.

Revoyons donc rapidement quelques caractéristiques de la couche 3 (couche Réseau) :

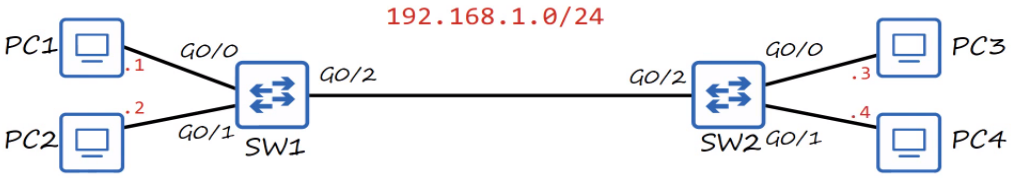
- Fournit une connectivité entre les hôtes de différents réseaux (en dehors du LAN)

- Fournit un adressage logique (IP adresses)

- Fournit une sélection de chemins entre la source et la destination, car sur internet il existe plusieurs chemins possibles pour atteindre une destination

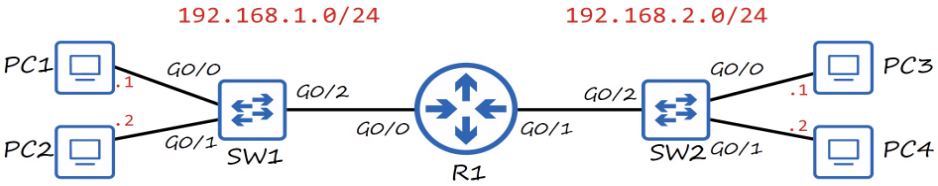
- Les routeurs fonctionnent à la couche 3

Prenons l'exemple du schéma suivant :



Nous avons vu auparavant que lorsqu'un PC veut connaître les adresses d'un réseau local il va envoyer des requêtes en Broadcast sur tout le réseau.

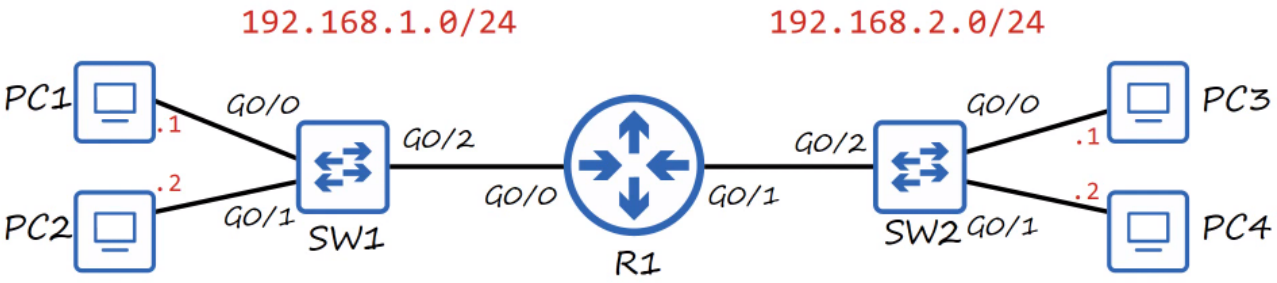
Mais que se passe t-il si l'on ajoute un routeur entre les deux switchs ?



A présent au lieu d'y avoir 1 seul réseau le réseau est divisé en 2 avec deux adresses sur deux réseaux différents : 192.168.1.0/24 ; 192.168.2.0/24

Le /24 est pour indiquer quelle est l'adresse fixe du réseau qui ne peut pas être changé avec : 192.168.1 ; le 0 est le chiffre qui change

Une dernière chose à ajouter sur le schéma est l'adresse IP des interfaces du routeur. Pour G0/0 se sera : 192.168.1.254 et pour G0/1 se sera : 192.168.2.254



Cette fois si le PC1 envoie une requête sur tout le réseau la trame sera distribuée sur toutes les interfaces du switchs jusqu'à l'interface du routeur sur son même réseau, mais la requête n'ira pas sur l'interface local de l'autre routeur.

Ceci est une capture d'écran d'une trame réseau prise sur Wikipédia :

IPv4 Header Format

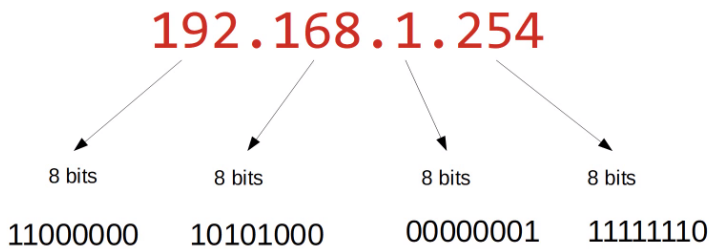
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags				Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

On voit bien que la longueur d'une adresse IP est de 32 Bits de longueur (ou 4 bytes)

Si l'on prend l'exemple de l'adresse 192.168.1.254 :

Chaque numéro représente 8 bits.

Il est possible de convertir ces chiffres de décimal en binaire ce qui donnerait les chiffres suivants :



Avant de voir comment fonctionnent les binaire revoyons comment fonctionne le décimal :

Il y a 10 chiffres dans le décimal : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ont dis aussi que c'est un comptage en base 10.

Il y a 16 chiffres en Hexadécimal : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

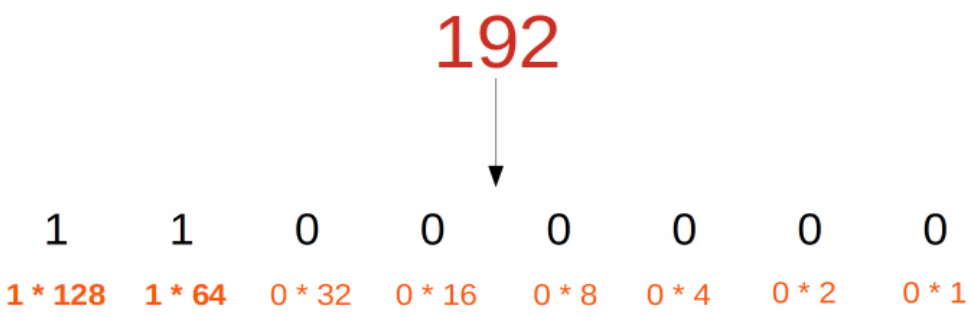
donc si l'on fais par exemple D * 16 (D = 13) se sera égal à 208 en décimal

Si l'on fais C * 256 (C = 12) se sera égal à 3072 en décimal

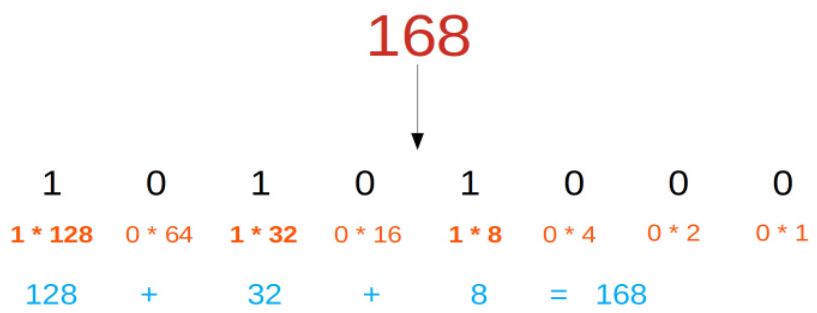
Si l'on additionne C + D se sera égal à 208 + 12 = 220

Voyons à présent comment fonctionne le Binaire.

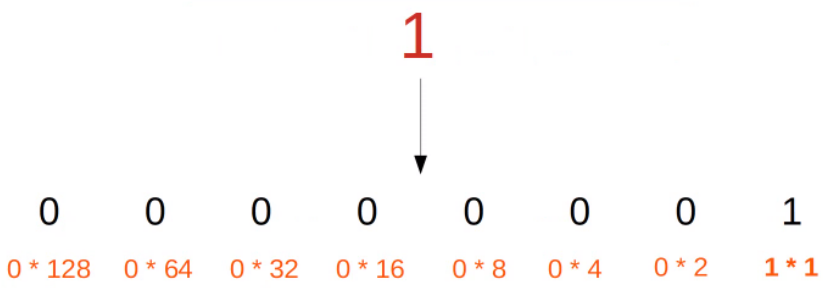
Pour convertir 1100000 en décimal il faut utiliser la méthode suivante : Multiplier par 2 à chaque fois en partant de droite et multiplier au nombre correspondant ainsi :



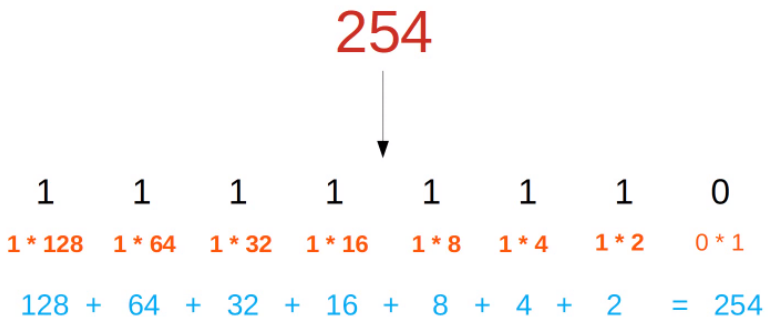
Pour 168 cela donnera :



Pour 1 cela donnera :



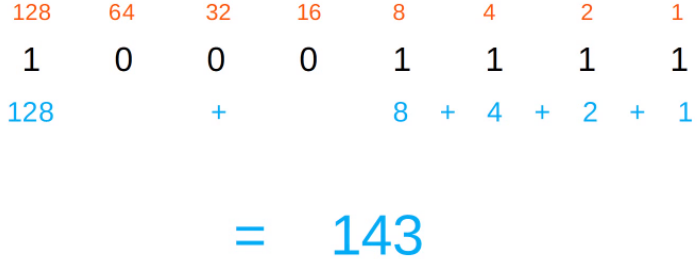
Pour 254 cela sera égal à :



Essayons de convertir 10001111 en décimal se sera égal à :

Il faut multiplier par deux la position de chaque chiffre pour qu'il soit converti en décimal :

Une fois que l'on a fais cela il faut juste additionner les numéro sur lesquelles il y a un 1 ce qui donne :



Un autre exemple on veut convertir 01110110 en décimal.

Il faut d'abord multiplier par deux la position de chaque chiffre pour qu'il soit converti en décimal :

128	64	32	16	8	4	2	1
0	1	1	1	0	1	1	0

On peut à présent additionner les numéro sur lesquelles il y a un 1 ce qui donne :

128	64	32	16	8	4	2	1
0	1	1	1	0	1	1	0
	64 +	32 +	16 +		4 +	2	

= 118

Essayons un 3ème exemple pour 11101100.

On multiplie par deux à chaque numéro et on additionne le résultat ce qui donne :

1	1	1	0	1	1	0	0
128 +	64 +	32 +		8 +	4		

= 236

A présent essayons de convertir du décimal vers le binaire :

Il faut ajouter déjà les 8 chiffres possible en binaire en multipliant à chaque fois par 2 en partant de la droite puis soustraire à chaque fois jusqu'à réduire à 0 le chiffre décimal donc pour 221 ça fera :

128 64 32 16 8 4 2 1

221 93 28 12 4

-128 -64 -16 -8 -4

=93 =28 =12 =4 =0

On met un « 1 » à chaque endroit sur lequel il y a eu une soustraction se qui fera :

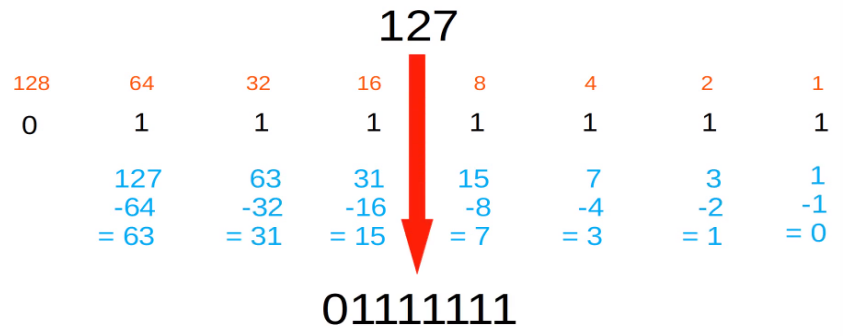
221

128	64	32	16	8	4	2	1
1	1	0	1	1	1	0	0
221 -128 = 93	93 -64 = 28		28 -16 = 12	12 -8 = 4	4 -4 = 0		

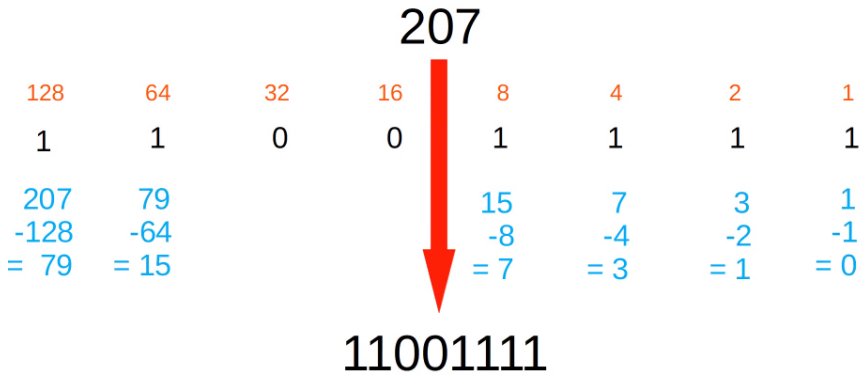
11011100

Un autre exemple pour convertir 127 en binaire.

Il faut déjà multiplier par deux à chaque chiffre décimal, puis ajouter le chiffre « 1 » à chaque endroit ou il y a eu une soustraction se qui sera égal à :

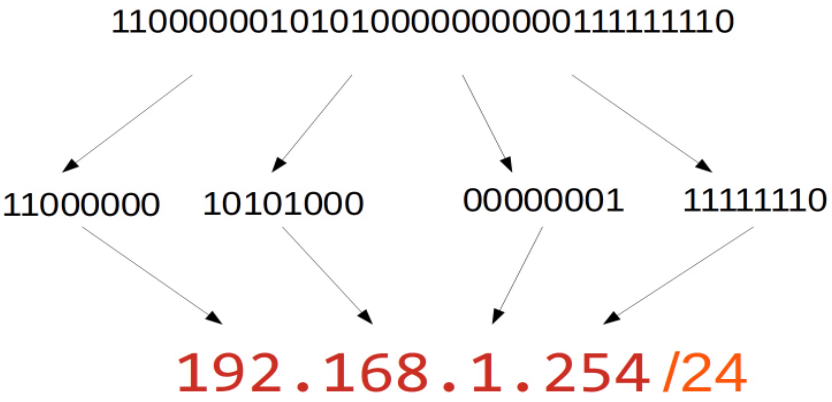


Faisons un dernière exemple pour convertir 207 en binaire :



Il est à noter que 00000000 est égal à 0 en décimal et que 11111111 est égal à 255 en décimal.

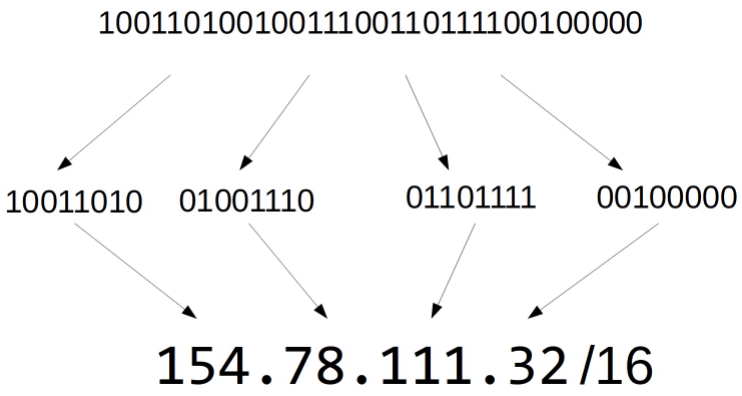
A présent que nous avons vu comment convertir des chiffres voyons comment convertir des adresses IP. Par exemple pour l'adresse : 11000000101010000000000111111110 il faut déjà séparer l'adresse en 4 parties de 8 puis la convertir se qui fera :



Ici le /24 représente les 24 bits qui font partie du réseau et qui sont fixes les 8 bits restants (24 + 8 = 32) correspondent à l'adresse de l'hôte.

Donc ici 254 est l'adresse d'hôte du réseau.

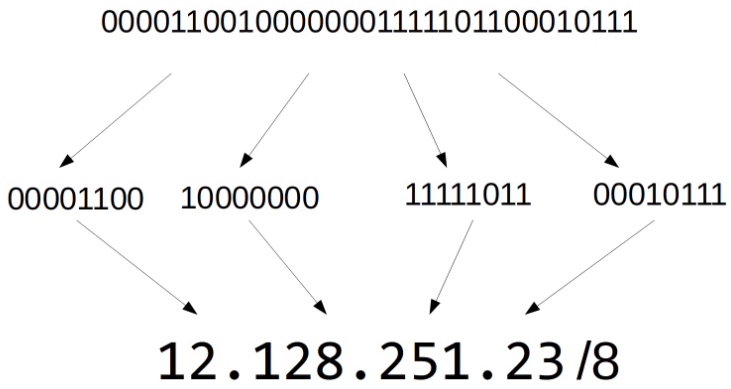
Pour convertir l'adresse 10011010010011100110111100100000 on sépare déjà l'adresse en 4 partie de 8 chiffres puis on convertie en décimal se qui fera :



Ici le /16 indique que les 16 premiers chiffres binaires représente l'adresse du réseau fixe.

Donc ici l'adresse réseau est 154.78

Un dernier exemple pour convertir 00001100100000000111110110001011 1 cela donne :



Ou 12 est l'adresse réseau puisque ici il y a un /8

Voyons à présent les différentes classes d'adresses IP qui sont définis en fonction du premier octet :

Classe	Premier Octet	Premier Octet avec un classement numérique
A	0xxxxxxx	0-127
B	10xxxxxx	128-191
C	110xxxxx	192-223
D	1110xxxx	224-239
E	1111xxxx	240-255

Les classes principales sont les classes A, B, et C

Les adresses des classes D sont utilisées pour les adresses Multicast

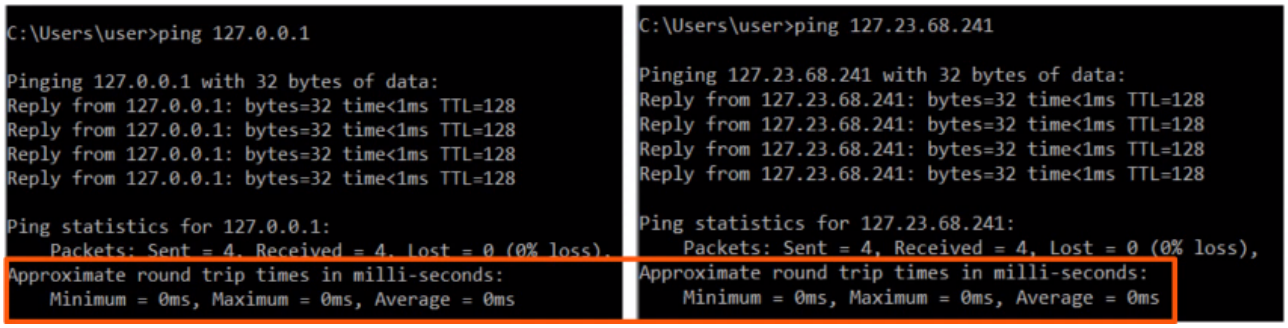
Les adresses des classes E sont réservées pour une utilisation expérimental

La fin de l'adresse de classe A est terminé par 126 et non pas 127 voici pourquoi :

Il y a une adresse de loopback qui à pour classement : 127.0.0.0 – 127.255.255.255

Ces adresses sont utilisés pour tester sa connectivité réseau sur l'appareil local.

Par exemple si l'on ping sur les adresses 127.0.0.1 et 127.23.68.241 on obtient les résultats suivants qui sont égal à 0 puisque le ping est fais sur son propre appareil.



On peut classer le réseau de cette façon :

Classes	Premier Octet	Premier Octet avec classement numérique	Longueur du préfixe
A	0xxxxxxx	0-127	/8
B	10xxxxxx	128-191	/16
C	110xxxxx	192-223	/24

Voici une capture d'écran Wikipédia pour mieux comprendre comment sont répartis les classes :

Class	Leading bits	Size of *network number* bit field	Size of *rest* bit field	Number of networks	Addresses per network
Class A	0	8	24	128 (2^7)	16,777,216 (2^{24})
Class B	10	16	16	16,384 (2^{14})	65,536 (2^{16})
Class C	110	24	8	2,097,152 (2^{21})	256 (2^8)

Donc les classes sont répartis avec les masques suivants :

Classes A : /8 255.0.0.0

Classes B : *16* 255.255.0.0

Classes C : /24 255.255.255.0

Si la portion d'hôte de l'adresse est 0 cela signifie qu'il s'agit de l'adresse réseau par exemple 192.168.1.0/24 cette adresse ne peut pas utiliser pour un hôte la première adresse est donc 192.168.1.1

Si la portion de l'adresse est 1 cela signifie qu'il s'agit de l'adresse de Broadcast, cette adresse ne peut pas être assigné à un hôte, ici c'est 192.168.1.254

Cours 8 : Adressage IPv4 (Partie 2)

Ce cours est la suite du cours précédent sur l'adressage IP.

Commençons par calculer le nombre maximal d'hôtes par réseau.

Pour l'adresse 192.168.1.0/24 l'adresse la plus élevé du réseau est 192.168.1.255/24

ce qui fais un total de 8bit donc $2^8 = 256$

mais comme l'adresse qui finit par 0 n'est pas assignable à un hôte puisque c'est l'adresse du réseau (192.168.1.0/24) et que la dernière adresse du réseau est l'adresse de Broadcast n'est pas non plus assignable à un hôte (192.168.1.255) il faut soustraire 2 donc ce qui fait que le nombre maximal d'hôte par réseau est de 2^8 - 2 = 254

Pour l'adresse réseau 172.16.0.0/16 l'adressage maximal est 172.16.255.255/16 ce qui fait que l'adressage de la partie hôte est de 16 bits = 2^{16} = 65 536 – 2 = 65 534

Pour le réseau 10.0.0.0/8 l'adressage maximal est 10.255.255.255/8 ce qui fait que l'adressage de la partie hôte est de 24 bits = 2^{24} = 16 777 216 – 2 = 16 777 214

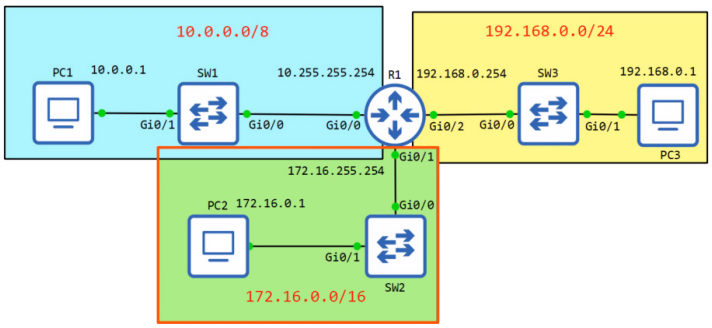
Donc la formule pour connaître le nombre maximal d'hôte par réseau est égal à 2 puissance n nombres d'hôtes – 2

Donc pour le réseau 192.168.1.0/24 la première adresse est 192.168.1.1/24 la dernière adresse est 192.168.1.254/24

Pour le réseau 172.16.0.0/16 la première adresse est 172.16.0.1/16 la dernière adresse est 172.16.255.254/16

Pour le réseau 10.0.0.0/8 la première adresse est 10.0.0.1/8 la dernière adresse est 10.255.255.254/8

A présent disons que l'on veut assigner des adresses sur le schéma suivant :



Voici la commande pour confirmer le statut de chaque interface pour vérifier l'adresse IP :

```
R1>en
R1#show ip interface brief
```

Voici le résultat d'une commande lorsque les interfaces n'ont pas été configurés :

```
R1>en
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         unassigned      YES unset  administratively down down
GigabitEthernet0/1         unassigned      YES unset  administratively down down
GigabitEthernet0/2         unassigned      YES unset  administratively down down
GigabitEthernet0/3         unassigned      YES unset  administratively down down
R1#
```

Pour configurer une interface il faut utiliser les commandes suivantes :

```
R1#conf t
R1(config)#interface gigabitethernet 0/0
R1(config-if)#
```

Une fois que l'on est entré dans la configuration pour configurer l'interface on peut utiliser les commandes suivantes pour assigner l'adresse IP et allumer l'interface :

```
R1(config-if)#ip address 10.255.255.254 255.0.0.0
R1(config-if)#no shutdown
```

```
R1(config-if)#ip address 10.255.255.254 ?
  A.B.C.D  IP subnet mask

R1(config-if)#ip address 10.255.255.254 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#
*Dec  7 08:29:08.937: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Dec  7 08:29:09.938: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/
0, changed state to up
R1(config-if)#
```

Voici également une capture d'écran de la commande pour voir un aperçu des interfaces :

```
R1(config-if)#do sh ip int br
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         10.255.255.254  YES manual up                    up
GigabitEthernet0/1         unassigned      YES unset  administratively down down
GigabitEthernet0/2         unassigned      YES unset  administratively down down
GigabitEthernet0/3         unassigned      YES unset  administratively down down
R1(config-if)#
```

Nous allons à présent assigner l'adresse de l'interface g0/1 :

```
R1(config-if)#int g0/1
R1(config-if)#ip add 172.16.255.254 255.255.0.0
R1(config-if)#no shut
R1(config-if)#
*Dec  7 08:51:42.648: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
*Dec  7 08:51:43.649: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/
1, changed state to up
R1(config-if)#do sh ip int br
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         10.255.255.254  YES manual up                    up
GigabitEthernet0/1         172.16.255.254  YES manual up                    up
GigabitEthernet0/2         unassigned      YES unset  administratively down down
GigabitEthernet0/3         unassigned      YES unset  administratively down down
R1(config-if)#
```

Et de l'interface g0/2 :

```
R1(config-if)#int g0/2
R1(config-if)#ip add 192.168.0.254 255.255.255.0
R1(config-if)#no shut
R1(config-if)#
*Dec  7 09:05:41.505: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
R1(config-if)#
*Dec  7 09:05:42.505: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/
2, changed state to up
R1(config-if)#do sh ip int br
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         10.255.255.254  YES manual up                    up
GigabitEthernet0/1         172.16.255.254  YES manual up                    up
GigabitEthernet0/2         192.168.0.254   YES manual up                    up
GigabitEthernet0/3         unassigned      YES unset  administratively down down
R1(config-if)#
```

Voici quelques autres commandes utiles pour voir les interfaces :

```
R1#show interface g0/0
```

Avec cette commandes on peut voir des informations concernant les couches 1, 2 et 3 de l'interface.

Une autre commande utile est :

```
R1#show interfaces description
```

Il est possible d'ajouter une description pour chaque interface qui sera :

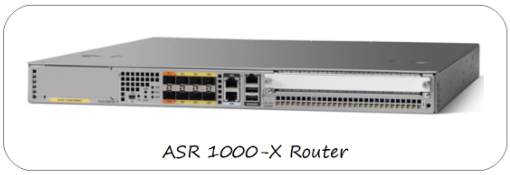
```
R1(config)#int g0/0
R1(config-if)#description ## to SW1 ##
```

Pour voir la description de chaque interface on peut lancer la commandes :

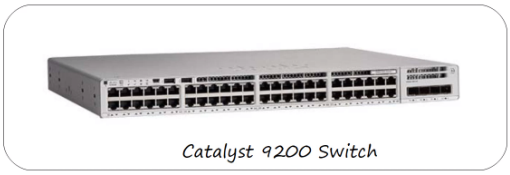
```
R1(config-if)#do show interface description
```

Cours 9 : Switch Interfaces

Dans ce cours nous verrons quelles sont les particularités des interfaces Switchs et quelles sont ses différences avec les interfaces des routeurs. Nous parlerons de la vitesse des interfaces et du duplex pour savoir si un appareil peut recevoir et envoyer des données en même temps, nous verrons ensuite se qu'est l'autonegotiation ne parlerons du statut des interfaces. Nous compterons les interfaces et erreurs.



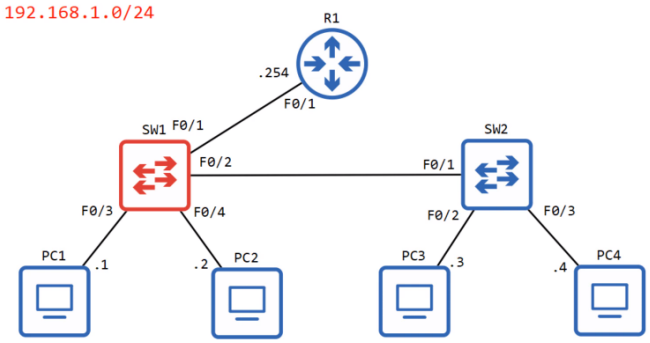
ASR 1000-X Router



Catalyst 9200 Switch

Le routeur n'a que 8 interfaces SFP tandis que le switch a en tout 48 interfaces.

Dans ce cours nous utiliserons la topologie suivante :



Nous allons commencer par configurer le SW1 en lançant les commandes suivantes :

```
SW1>en
SW1#sh ip int br
```

On peut voir que certaines interfaces sont éteintes puisque la commande shutdown est appliqué par défaut donc le statut sera automatiquement down/down par défaut.

Les interfaces Switch n'auront pas la commande « shutdown » appliqué par défaut le statut sera up/up s'il est connecté à un autre appareil ou en down/down si non connecté à un autre appareil.

Une autre commande pour connaître le statut des interfaces est :

```
SW1#show interfaces status
```

Avec cette commande ont peut voir le Numéro de port, le statut de l'interface, la Vlan associé, si l'interface est en Duplex ou non, la vitesse de l'interface, et le type d'interface.

Voici les commandes pour modifier la vitesse d'une interface et le duplex de l'interface :

```
SW1#conf t
SW1(config)#int f0/1
SW1(config-if)#speed 100
SW1(config-if)#duplex full
SW1(config-if)#description ## to R1 ##
```

Ici l'interface à été configuré avec une vitesse de 100mbps et en full duplex avec une description.

Nous configurons les autres interfaces de la même manière en correspondance avec le schéma

Il est possible d'utiliser les commandes rapidement depuis le mode config global avec les commandes :

```
SW1(config)#interface range f0/5-12
SW1(config-if-range)#description ## not in use ##
SW1(config-if-range)#shutdown
```

Ici les interface allant de f0/5 à f0/12 ont été allumé en une seule commande.

Nous allons à présent expliquer rapidement se qu'est Full et Half Duplex.

- Half Duplex signifie que l'appareil ne peut pas envoyer ni recevoir des données en même temps. S'il reçoit une trame il doit attendre avant d'envoyer la trame.

- Full Duplex signifie que l'appareil peut envoyer et recevoir des données en même temps. Il n'a pas besoin d'attendre.

Full Duplex et la manière préféré à utiliser.

Voyons à présent se qu'est que la collision de données.

Sur un Hub lorsqu'un appareil veut envoyer une donnée le Hub inonde la requête vers les autres interfaces, si la requête est envoyé vers une interface et que l'autre interface répond au même moment, il va y avoir ce que l'on appelle une « collision domain » dans lesquelles les données ne seront pas acheminées puisqu'il y a eu une collision.

Pour parvenir à résoudre ce type de problématique il y a une fonction intégrée sur les Switch appelé CI/CD (Carrier Sense Multiple Access)

Cela fonctionne de la manière suivante :

- Avant d'envoyer la trame les appareils écoutent le collision domain jusqu'à se qu'ils détectent qu'aucun autre appareil n'envoie de données.

- Si une collision se passe tout de même, l'appareil envoie un signal pour informer les autres appareils qu'une collision s'est passé.

- Chaque appareil va attendre une petite période aléatoire de temps avant de renvoyer une nouvelle trame.

- Puis le processus se répète.

Les Hubs sont surtout des répéteurs de signal sans vraiment qu'il y ait de filtres il sont surtout familiers avec la couche 2, les Switch sont plus sophistiqués pour filtrer les trames, ils sont familiers avec la couche 2.

Sur les Switch les données peuvent passer et envoyer les données en Full Duplex et ne vérifient pas si d'autres données se distribuent sur le réseau. Il y a donc très peu de problèmes de collision.

Parlons à présent de la vitesse/Duplex autonegotiation

Les interfaces qui peuvent être lancés à différentes vitesses (10/100 ou 10/100/1000) ont des paramètres par défaut avec speed auto et duplex auto.

Les interfaces avertissent leurs capacités aux appareils voisins et négocient la meilleure vitesses possible en paramètres duplex.

Que se passe t-il si autonegotiation est désactivé sur l'appareil connecté au Switch ?

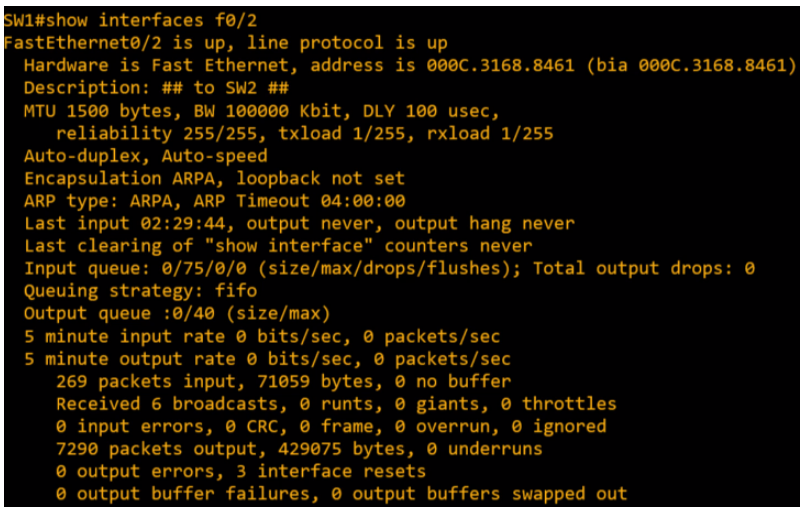
Pour la vitesse, il essaiera de détecter la vitesse sur laquelle l'autre appareil fonctionne.

S'il ne réussie pas à détecter la vitesse il utilisera la vitesse la plus lente.

Pour le paramètre de Duplex, Si la vitesse est de 10 ou 100Mbps le switch va utiliser le half duplex.

Si la vitesse est de 1000Mbps ou plus il utilisera le full duplex.

Avec la commande « show interface f0/2 » il est possible d'afficher différentes statistiques de la configuration.



Ont peut voir dans les dernières lignes quelques paramètres avec ces fonctions :

- Runts : Ce sont les trames plus petites que la taille de trame minimal (64 bytes)

- Giants : Ce sont les trames plus grandes que la taille de trame maximal (1518 bytes)

- CRC : Ce sont les trames qui ont une vérification de CRC qui échoue (dans le Ethernet FCS)

- Frame : Ce sont les trames qui ont un format incorrect à cause d'une erreur

- Input errors : Ce sont le total de différents compteurs comme les quatre autres

- Output errors : Ce sont les trames que le switch essaie d'envoyer, mais ne parvient pas à cause d'une erreur.

Cours 10 : Entête IPv4

Dans ce cours nous allons apprendre à voir de quoi est composé l'entête IPv4, c'est utilisé dans la couche 3 pour envoyer des données entre des réseaux séparés, même sur des réseaux séparés de par le monde géographiquement. C'est connu sous le nom de « routing ».

Nous commencerons par voir la structure des paquet IPv4, puis les parties de l'entête IPv4.

Revoyons rapidement de quoi est composé le modèle OSI. Tout d'abord il y a la donnée, puis le segment composé de la donnée avec l'entête IPv4 de couche 4. Ce segment est ensuite encapsulé par une entête de couche 3, le segment est appelé Paquet avec la nouvelle entête ajouté.

Après cela est encapsulé l'entête de la couche 2 et le trailer de la couche 2, le paquet s'appelle Trame (Frame en anglais) avec son entête et trailer ajouté.

Tous ces noms (Donnée, Segment, Paquet, Trame) sont appelé Protocol Data Units (PDUs)

Dans ce cours nous nous concentrerons sur l'entête de couche 3.

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

Nous allons détailler chaque champs qui composent l'entête :

1. Version : avec une longueur de 4 bits, il permet d'identifier la version de l'IP utilisé.

IPv4 = 4 (0100)

IPv6 = 6 (0110)

2. IHL (Internet Header Lenght) : avec une longueur de 4 bits, puisque la longueur de la partie de l'entête IPv4 varie, cette partie est nécessaire pour indiquer la longueur total de l'entête.

La valeur minimal pour cette partie est de 5 (= 20 bytes), la valeur maximal est de 15 (15 * 4 bytes = 60 bytes)

La longueur minimal pour une entête IPv4 est de 20 bytes, la longueur maximal pour une entête IPv4 est de 60 Bytes.

3. le DSCP (Differentiated Services Code Point) : d'une longueur de 6 bits, il est utilisé pour le QoS (Quality of Service), il est utilisé pour prioriser le délai de données sensibles (le streaming de voix, la video, etc.)

4. ECN (Explicit Congestion Notification) : avec une longueur de 2 bits, il fournit une notification entre deux extrémité d'un réseau en congestion sans perte de paquet.

C'est un champ optionnel qui requière les deux extrémité, comme les connaissances de l'infrastructure réseau pour le supporter.

5. La longueur total : d'une longueur de 16 bits, il indique la longueur total d'un paquet (la couche de l'entête 3 avec le segment de couche 4), il est mesuré en bytes. La valeur minimal de ce champ est 20 (= l'entête IPv4 sans les données encapsulés).

La valeur maximal est 65 535 (avec une valeur maximal de 16 bit)

6. Identification : d'une longueur de 16 bits, si un paquet est fragmenté car trop large, ce champ est utilisé pour identifier quelle paquet le fragment appartient.

Tous les fragments du même paquet ont leurs propre entête IPv4 avec la même valeur dans ce champ.

Les paquets sont fragmentés s'il sont plus large que le MTU (Maximum Transmission Unit)

Le MTU est de 1500 bytes, les fragments sont réassemblés par l'hôte de réception.

7. Flag : d'une longueur de 3 bits il est utilisé pour contrôler/identifier le fragment. Le Bit 0 est réservé toujours pour 0. Le Bit 1 (DF bit) ne fragmente pas il est utilisé pour indiquer qu'un paquet ne devrait pas être fragmenté. Le Bit 2 (MF bit) il est placé à 1 s'il y a plus de fragments dans le paquet, placé à 0 pour le dernier fragment.

8. Fragment Offset : d'une longueur de 13 bits, ce champs est utilisé pour indiquer la position du fragment sans l'original, paquet IP non fragmenté. Il permet aux paquets fragmenté d'être réassemblés même si les fragments arrivent dans le désordre.

9. Time To Live : d'une longueur de 8 bits. Un routeur ne retient pas un paquet avec un TTL de 0. Il est utilisé pour empêcher les boucles infinis et éviter que le paquet ne soit renvoyés plusieurs fois cela permet d'éviter les erreurs.

Ce champs a été conçu pour indiquer la durée de vie maximal d'un paquet en secondes.

En pratique cela indique un « compte des bond », chaque fois que le paquet arrive à un routeur, le routeur diminue le TTL de 1. Le TTL recommandé par défaut est 64.

10. Protocol : d'une longueur de 8 bits il indique le protocole du PDU de couche 4. ce peut être :

- Valeur de 6 pour TCP

- Valeur de 17 pour UDP

- Valeur de 1 pour ICMP

- Valeur de 89 pour OSPF (Dynamic routing Protocol)

11. Header Checksum : d'une longueur de 16 bits il est utilisé pour vérifier s'il y a des erreurs dans l'entête IPv4. Lorsqu'un routeur reçoit un paquet il calcule le checksum de l'entête et le compare à un dans ce champs de l'entête, s'il ne sont pas de même taille le routeur ne reçoit pas le paquet.

Il est utilisé pour vérifier des erreurs seulement dans l'entête IPv4. Les IP dépendent du protocole encapsulé pour détecter des erreurs dans la donnée encapsulé.

Le TCP et UDP ont leurs propres champs de checksum pour détecter les erreurs dans les données encapsulés.

12. Source/Destination IP adresse : chacun de ces champs est d'une longueur de 32 bits car c'est la longueur pour une adresse IPv4.

La source de l'adresse IP indique l'adresse IPv4 de l'envoyeur.

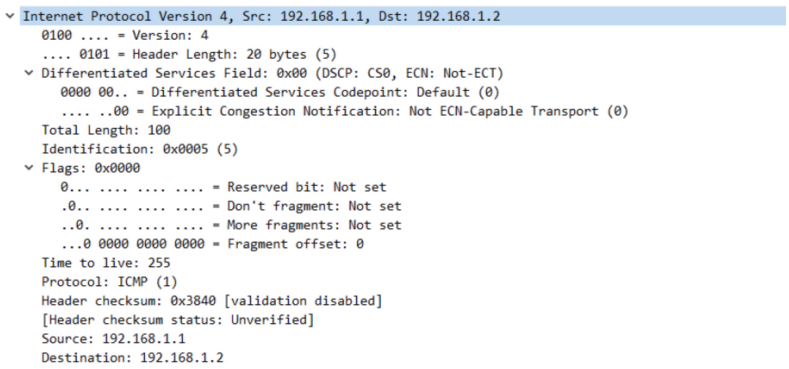
La destination de l'adresse IP indique l'adresse IPv4 du receveur.

13. Options : sa longueur peut varier de 0 à 320 bits de longueur, il est rarement utilisé, si le champs IHL est de plus de 5 cela signifie que l'Options est présent.

Field	Size (bits)	Description
Copied	1	Set to 1 if the options need to be copied into all fragments of a fragmented packet.
Option Class	2	A general options category. 0 is for "control" options, and 2 is for "debugging and measurement". 1 and 3 are reserved.
Option Number	5	Specifies an option.
Option Length	8	Indicates the size of the entire option (including this field). This field may not exist for simple options.
Option Data	Variable	Option-specific data. This field may not exist for simple options.

Il est possible d'utiliser le logiciel Wireshark pour mieux analyser le trafique réseau avec les paquet capturés dans le réseau.

Voici la capture d'écran d'un paquet sur Wireshark :



On peut voir sur cette image tous les différents champs analysés de l'entête vu auparavant.

Il s'agit d'une requête de Ping avec le protocole ICMP.

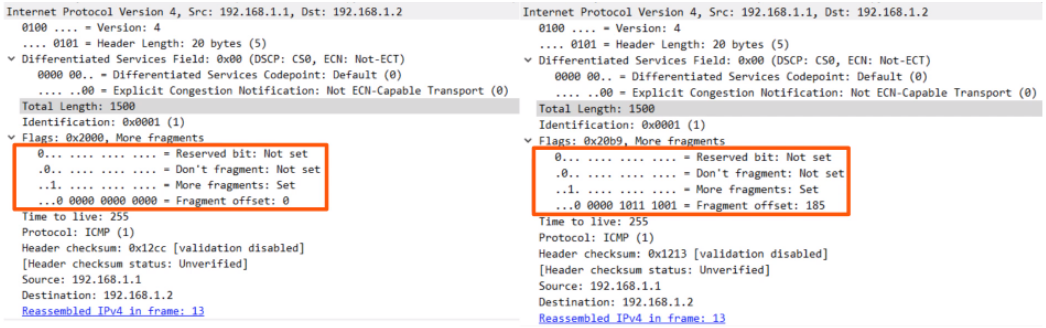
Si l'on lance la commande :

```
R1#ping 192.168.1.2 size 10000
```

La requête de ping sera fragmenté en plusieurs partis comme on pourra le voir sur la capture Wireshark :

7	17.411175	192.168.1.1	192.168.1.2	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=0001) [Reassembled in #13]
8	17.412827	192.168.1.1	192.168.1.2	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=0001) [Reassembled in #13]
9	17.414347	192.168.1.1	192.168.1.2	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=2960, ID=0001) [Reassembled in #13]
10	17.415913	192.168.1.1	192.168.1.2	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=4440, ID=0001) [Reassembled in #13]
11	17.417560	192.168.1.1	192.168.1.2	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=5920, ID=0001) [Reassembled in #13]
12	17.419203	192.168.1.1	192.168.1.2	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=7400, ID=0001) [Reassembled in #13]
13	17.420793	192.168.1.1	192.168.1.2	ICMP	1134	Echo (ping) request id=0x0000, seq=1/256, ttl=255 (reply in 20)

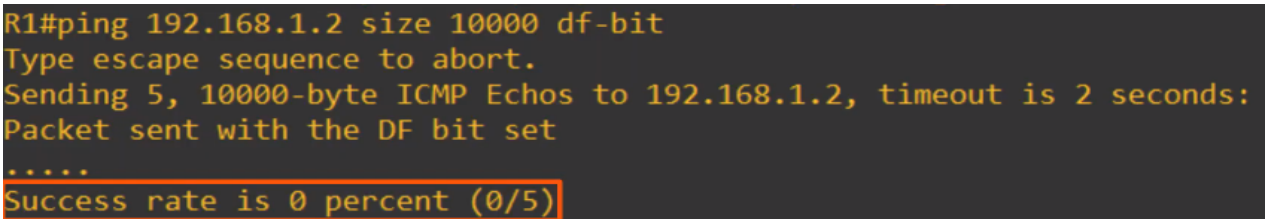
Il est indiqué « réassemblé dans le #13 » cela correspond à la requête 13 qui est fragmenté en plusieurs partis. Voici plus en détail ce que contiennent ces parties fragmentées sur Wireshark :



Il est possible d'ajouter une option pour empêcher cette fragmentation en utilisant la commande :

```
R1#ping 192.168.1.2 df-bit
```

Si cette commande est utilisée pour ping avec une taille de 10000 bit, le ping ne fonctionnera pas il est il est obligatoire que le paquet soit fragmenté pour être distribué.

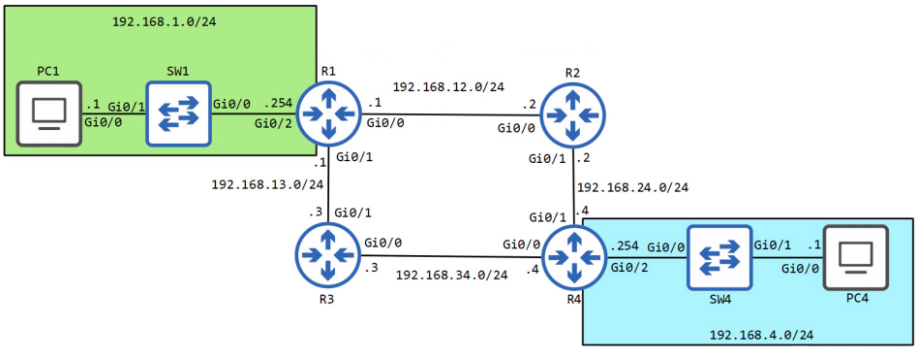


Cours 11 : Routage statique

Dans ce cours nous verrons comment les routeurs redistribuent le trafique sur les différents réseaux.

Nous verrons le processus du routage IP, nous examinerons la table de routage d'un routeur Cisco, puis nous ferons la configuration d'un routage statique.

Voici la topologie réseau que nous utiliserons pour ce cours :



Il y a deux LAN l'une connecté au R1 et l'autre connecté avec le R4, les quatre routeurs au centre forment un WAN (Wide Area Network).

Imaginons que le PC1 veut envoyer une requête au PC4, pour cela il va vérifier si le PC est sur son réseau, comme l'adresse de destination n'est pas sur le même réseau (192.168.4.1) le PC1 va envoyer le paquet à la passerelle par défaut (default gateway), dans le schéma la passerelle par défaut du LAN est ici l'interface de R1. R1 est en possession du paquet et doit le distribuer vers le PC4. R1 va commencer par comparer l'adresse IP de destination avec la table de routage puis va distribuer le paquet à R2 qui correspond à la destination avec l'interface 192.168.12.2

R2 va faire le même fonctionnement en comparant l'adresse avec sa table de routage et le distribuer à R4. R4 en possession du paquet va juste redistribuer le paquet puisqu'il se trouve sur le bon réseau.

On utilise la commande suivante pour afficher la table de routage :

```
PC1#show ip route
```

```
PC1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
PC1#
```

On peut voir l'adresse 192.168.1.1/32 le L signifie Local c'est pour cela qu'il s'agit d'une adresse en /32 cela signifie qu'il n'y a qu'une seule adresse sur le réseau.

Pour configurer la passerelle d'un routeur Cisco doit être configuré la route par défaut. Une route par défaut est une route qui peut joindre toutes les destinations possible. Il est utilisé si une route spécifique ne correspond pas avec la table de routage.

La route par défaut est spécifique avec :

Adresse IP : 0.0.0.0

Masque de sous réseau : 0.0.0.0

Il faut donc configurer l'adresse de routage à 0.0.0.0/0 qui correspond à toute les adresses possible

L'adresse 0.0.0.0/0 peut joindre toute les adresses puisqu'on a dis auparavant que le masque de sous réseau était là pour fixer l'adresse du réseau. Par exemple avec un 192.168.1.0/24 ce sont les 24 premiers octet qui sont fixes

donc ici 192.168.1 le reste de l'adresse n'est pas fixe. Donc sur le même raisonnement avec un masque de sous réseau de 0.0.0.0 se sont toute les adresse qui sont non fixes et donc joignable sur le réseau.

Pour l'adresse 192.168.1.1/32 se sont tous les chiffres qui sont fixes puisque le masque est 255.255.255.255 donc ça n'est que cette adresse qui est joignable avec elle même.

On utilise la commande suivante pour configurer un routage statique :

```
PC1#conf t
PC1(config)#ip route {destination-address mask next-hop}
```

Nous configurons donc sur le PC1 le routage avec la commande suivante :

```
PC1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
PC1(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.254
```

Voici à présent le résultat de la commande show ip route :

```
PC1(config)#do sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is 192.168.1.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.168.1.254
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
PC1(config)#
```

On peut voir que l'adresse de routage par défaut est 0.0.0.0 et que le code sur le côté est différent avec un "S*" qui signifie static et l'asterisc signifie qu'il est par défaut.

Voici le résultat de la commande show ip route sur le routeur R1 :

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/2
L        192.168.1.254/32 is directly connected, GigabitEthernet0/2
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, GigabitEthernet0/0
L        192.168.12.1/32 is directly connected, GigabitEthernet0/0
      192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.13.0/24 is directly connected, GigabitEthernet0/1
L        192.168.13.1/32 is directly connected, GigabitEthernet0/1
R1#
```

Les switch inondent de trames avec des destinations inconnus. Les routeurs quant à eux lâchent le paquet avec des destinations inconnus.

Nous allons donc configurer la route par défaut pour le routeur R1 avec la commande :

```
R1#conf t
R1#ip route {destination-address mask exit-interface}
```

Sur R1 ça donnerais ce résultat :

```
R1(config)#ip route 192.168.4.0 255.255.255.0 g0/0
```

On peut à présent voir la table de routage de R1 avec la commande « show ip route » sur le R1 :

```
R1(config)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/2
L        192.168.1.254/32 is directly connected, GigabitEthernet0/2
S     192.168.4.0/24 is directly connected, GigabitEthernet0/0
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, GigabitEthernet0/0
L        192.168.12.1/32 is directly connected, GigabitEthernet0/0
      192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.13.0/24 is directly connected, GigabitEthernet0/1
L        192.168.13.1/32 is directly connected, GigabitEthernet0/1
```

Le paquet est donc distribué vers R2 mais pour que R2 lui redistribue le paquet vers R4 il faut configurer le routage avec les commandes suivantes :

```
R2(config)#ip route 192.168.4.0 255.255.255.0 192.168.24.4
R2(config)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

S     192.168.4.0/24 [1/0] via 192.168.24.4
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, GigabitEthernet0/0
L        192.168.12.2/32 is directly connected, GigabitEthernet0/0
      192.168.24.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.24.0/24 is directly connected, GigabitEthernet0/1
L        192.168.24.2/32 is directly connected, GigabitEthernet0/1
```

Le paquet est bien distribué à R4.

Nous n'avons pas configuré de routage statique par défaut est ce que cela sera un problème ?

Commençons par vérifier la table de routage de R4 :

```
R4#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected, GigabitEthernet0/2
L        192.168.4.254/32 is directly connected, GigabitEthernet0/2
      192.168.24.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.24.0/24 is directly connected, GigabitEthernet0/1
L        192.168.24.4/32 is directly connected, GigabitEthernet0/1
      192.168.34.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.34.0/24 is directly connected, GigabitEthernet0/0
L        192.168.34.4/32 is directly connected, GigabitEthernet0/0
```

Puisque l'adresse 192.168.4.0/24 est ajoutée sur la table de routage (car l'interface est directement connecté au réseau) il n'est pas nécessaire d'ajouter l'adresse pour que le paquet soit distribué au PC4

On peut à présent lancer un ping pour vérifier que le paquet est bien distribué jusqu'au PC4 :

```
PC1#ping 192.168.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
PC1#
```

Le ping n'a pas bien fonctionné, alors quelle est le problème ?

En faite le problème est que le PC4 ne peut pas renvoyer une réponse au ping du PC1 puisque son adresse n'est pas joignable sur le réseau (« One way reacheability » ou « une manière de joindre » en anglais), pour que le rendre joignable il faut configurer le routage par défaut du PC4, R2 et R4

Voici donc les tables de routage des appareils à présent configuré :

```
PC4(config)#ip route 0.0.0.0 0.0.0.0 192.168.4.254
PC4(config)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is 192.168.4.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.168.4.254
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected, GigabitEthernet0/0
L        192.168.4.1/32 is directly connected, GigabitEthernet0/0
```

```
R4(config)#ip route 192.168.1.0 255.255.255.0 192.168.24.2
R4(config)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

S     192.168.1.0/24 [1/0] via 192.168.24.2
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected, GigabitEthernet0/2
L        192.168.4.254/32 is directly connected, GigabitEthernet0/2
      192.168.24.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.24.0/24 is directly connected, GigabitEthernet0/1
L        192.168.24.4/32 is directly connected, GigabitEthernet0/1
      192.168.34.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.34.0/24 is directly connected, GigabitEthernet0/0
L        192.168.34.4/32 is directly connected, GigabitEthernet0/0
```

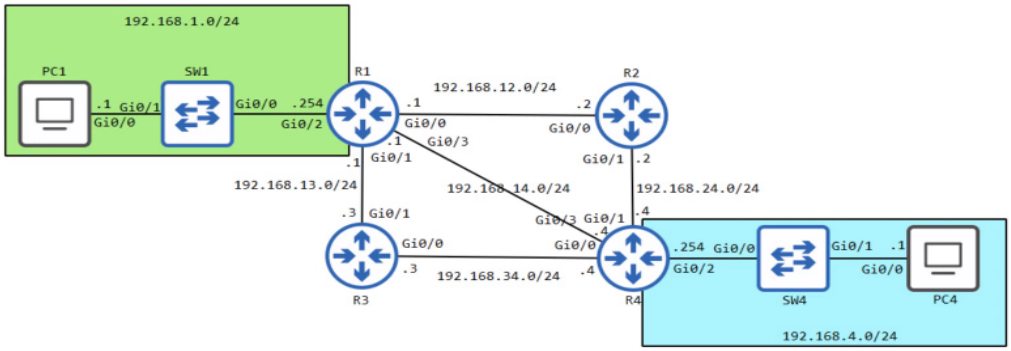
```
R2(config)#ip route 192.168.1.0 255.255.255.0 192.168.12.1
R2(config)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

S     192.168.1.0/24 [1/0] via 192.168.12.1
S     192.168.4.0/24 [1/0] via 192.168.24.4
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, GigabitEthernet0/0
L        192.168.12.2/32 is directly connected, GigabitEthernet0/0
      192.168.24.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.24.0/24 is directly connected, GigabitEthernet0/1
L        192.168.24.2/32 is directly connected, GigabitEthernet0/1
```

Nous pouvons à présent réessayer le ping et voir qu'il fonctionne cette fois ci :

```
PC1#ping 192.168.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 26/31/41 ms
PC1#
```

Lorsqu'un routeur vérifie l'adresse IP de destination dans sa table de routage il vérifie le routage le plus spécifique c'est à dire qui à le masque le plus haut (/32 > /24 > /16 > /8 > /0)

Pour mieux comprendre nous allons ajouter une interface au routeur R1 et R4 :



Voici la table de routage R1 nouvellement configuré :

```
S       192.0.0.0/8 [1/0] via 192.168.13.3
        192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C          192.168.1.0/24 is directly connected, GigabitEthernet0/2
L          192.168.1.254/32 is directly connected, GigabitEthernet0/2
        192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
S          192.168.4.0/24 is directly connected, GigabitEthernet0/0
S          192.168.4.1/32 [1/0] via 192.168.14.4
        192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C          192.168.12.0/24 is directly connected, GigabitEthernet0/0
L          192.168.12.1/32 is directly connected, GigabitEthernet0/0
        192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C          192.168.13.0/24 is directly connected, GigabitEthernet0/1
L          192.168.13.1/32 is directly connected, GigabitEthernet0/1
        192.168.14.0/24 is variably subnetted, 2 subnets, 2 masks
C          192.168.14.0/24 is directly connected, GigabitEthernet0/3
L          192.168.14.1/32 is directly connected, GigabitEthernet0/3
```

Disons que l'on veut faire un ping vers 192.168.4.1 quelle serait l'adresse la plus spécifique que le routeur pourrait utiliser ?

L'adresse la plus spécifique serait : 192.168.4.1/32

Mais il y a d'autres adresses joignables qui sont : 192.168.4.0/24 et 192.0.0.0/8

Autre exemple pour le ping vers 192.168.4.254 quelle serait l'adresse la plus spécifique que le routeur pourrait utiliser ?

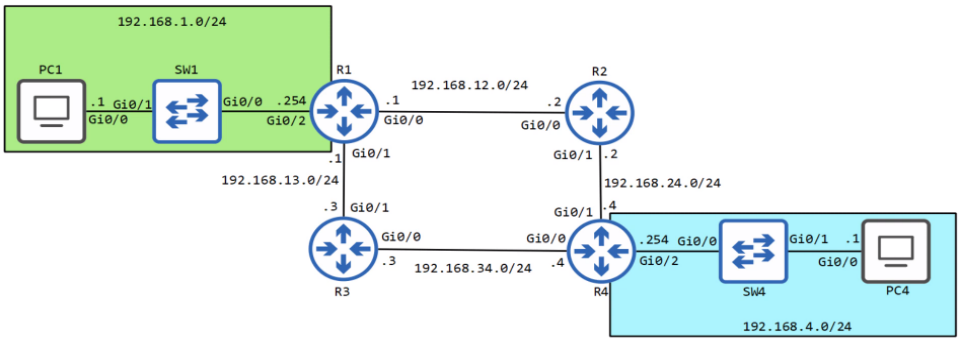
L'adresse la plus spécifique serait : 192.168.4.0/24

Mais l'autre adresse joignable est aussi : 192.0.0.0/8

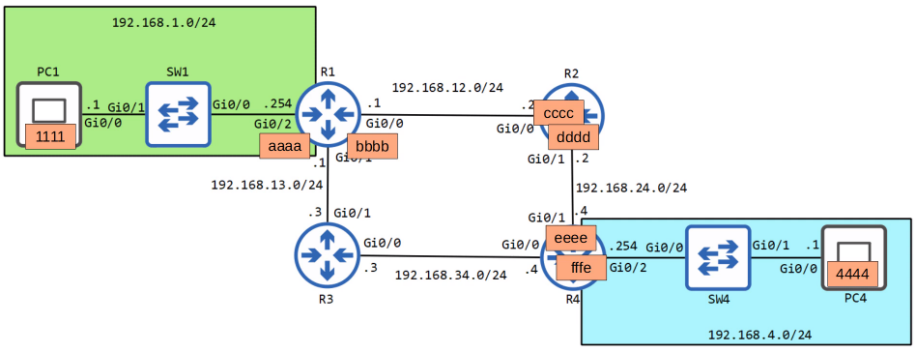
Cours 12 : Vie d'un paquet

Dans ce cours nous verrons le processus d'envois d'un paquet vers une destination, cela inclut ARP, l'encapsulation, de-encapsulation, etc..

Voici la topologie réseau que nous utiliserons dans ce cours :



Disons que la configuration a été faite sur chaque matériel et que le paquet va cheminer en passant par PC1, R1, R2, R4 pour arriver au PC4. Ajoutons seulement des adresses MAC pour chaque interfaces :



Disons que le PC1 veut envoyer un paquet au PC4.

PC1 n'a encore jamais envoyé de paquet, il faut donc d'abord qu'il utilise le protocole ARP (Address Resolution Protocol) pour identifier les adresses.

La requête ARP va être envoyé avec pour informations :

- Src IP : 192.168.1.1

- Dst IP : 192.168.1.254

- Dst MAC : ffff.ffff.ffff

- Src MAC : 1111

Ici la destination MAC est ffff.ffff.ffff car l'adresse MAC n'est pas connu.

Le R1 répond à la requête ARP car il reconnaît sa propre adresse IP et renseigne l'adresse MAC au PC1. L'adresse de requête est envoyé en Broadcast, tandis que la réponse ARP est envoyé en Unicast.

A présent que PC1 connaît l'adresse MAC de R1, PC1 envoie la requête ARP à R2 avec pour informations :

- Src IP : 192.168.12.1

- Dst IP : 192.168.12.2

- Dst MAC : ffff.ffff.ffff

- Src MAC : bbbb

L'adresse MAC de destination est ffff.ffff.ffff car l'adresse n'est pas connu.

R2 répond à la requête ARP en renseignant son adresse MAC à PC1.

PC1 connais à présent les adresses MAC de R1 et R2, il lui faut connaître l'adresse de R4.

R2 va envoyer cette fois la requête ARP pour connaître l'adresse MAC de R4 avec ces informations :

- Src IP : 192.168.24.2

- Dst IP : 192.168.24.4

- Dst MAC : ffff.ffff.ffff

- Src MAC : dddd

R4 répond à la requête ARP en renseignant son adresse MAC.

R2 distribue donc le paquet à R4 en unicast.

R4 possède à présent le paquet, il lui faut connaître l'adresse MAC de PC4 pour distribuer le paquet.

C'est cette fois R4 qui envoie la requête ARP au PC4 avec ces informations :

- Src IP : 192.168.4.254

- Dst IP : 192.168.4.1

- Dst MAC : ffff.ffff.ffff

- Drc MAC : fffe

PC4 répond à la requête et renseigne son adresse MAC à R4.

R4 distribue donc le paquet au PC4 en unicast, le paquet est arrivé à destination.

Il est à noter que le paquet n'a pas changé d'adresses IP dans ce processus cela reste les adresses du PC1 :

- Src : 192.168.1.1

- Dst : 192.168.4.1

A présent disons que PC4 veut envoyer une réponse à PC1, pour cela le paquet va repasser par R4, R2, R1, PC1. Mais la différence majeur avec l'envoie PC1 vers PC4 est que cette fois ci le paquet est directement acheminé vers PC1 sans requête ARP car les adresses MAC sont déjà ajouté.

Cours 13 : Sous Réseau (Partie 1)

Dans ce cours nous verrons ce qu'est le CIDR (Classless Inter-Domain Routing), puis le processus d'adressage réseau.

Tout d'abord commençons par rappeler les différentes classes d'adresse IP :

Classes	Premier octet (Binaire)	Premier octet (Décimal)
A	0xxxxxxx	0 – 127
B	10xxxxxx	128 – 191
C	110xxxxx	192 – 223
D	1110xxxx	224 – 239
E	1111xxxx	240 – 255

Seulement les adresses de classes A, B et C peuvent être assigner à un appareil :

Classes	Premier Octet (Binaire)	Premier Octet (Décimal)	Prefix Length
A	0xxxxxxx	0 – 127	/8
B	10xxxxxx	128 – 191	/16
C	110xxxxx	192 – 223	/24

Comment une entreprise obtient son propre nombre d'adresse IP à utiliser ?

C'est l'IANA (Internet Assigned Numbers Authority) qui assigne les adresses IPv4/réseau aux entreprises basé en fonction de leurs taille.

Par exemple une grande entreprise recevra plutôt un réseau de classe A ou classe B tandis qu'une petite entreprise recevra plutôt un réseau de classe C.

Imaginons à présent qu'il y ait 2 routeurs connectés entre eux avec dans chacune des LAN différents switchs connectés au routeur.

Les deux routeurs connectés entre eux sont connus comme des réseau point à point.

Par exemple disons qu'il s'agit d'un routeur basé à San francisco et d'un autre à New York.

Comme il s'agit d'un petit réseau point à point une adresse de classe C est assigné avec pour adresse : 203.0.113.0/24

Il y a donc en tout 256 adresses dont :

- 1 adresse réseau (203.0.113.0)

- 1 adresse de Broadcast (203.0.113.255)

- 1 adresse d'interface pour R1 (203.0.113.1)

- 1 adresse d'interface pour R2 (203.0.113.2)

il reste donc en tout ici 252 adresses restantes ce qui n'est pas idéal pour un réseau pouvant avoir 252 adresses.

Par exemple si une entreprise X à besoin d'adressage pour 5000 hôte. Les réseaux de classe C ne fourniront pas assez d'adresses, donc un réseau de classe B doit être assigné.

Cela résultera que 60000 adresses seront restantes et non utilisées.

Lorsque Internet à été crée pour la première fois, les créateurs n'ont pas prévu qu'Internet deviendra aussi grand qu'aujourd'hui. Cela résulte qu'il y a des adresses restantes à chaque fois.

Le IETF (Internet Engineering Task Force) introduit CIDR en 1993 pour remplacer l'adresse des systèmes. Avec CIDR il a été mis en place que Classe A serait égal à /8, Classe B à /16 et classe C à /24 les autres classes ont été supprimés.

Cela permettait à de grands réseaux d'être divisé en de petits réseaux, pour permettre une grande efficacité.

Ces petits réseaux sont appelés sous réseaux.

Donc pour l'adresse de point à point de nos deux routeurs les adresses assignées sont :

203.0.113..0 avec un masque de sous réseau de 255.255.255.0

Pour calculer le nombre d'adresses utilisable on fais le calcul suivant :

$2^8 - 2 = 254$ adresses utilisables.

Le puissance 8 correspond aux adresses des bits hôtes et le – 2 sont les adresses réseau et adresse de Broadcast.

CIDR permet d'utiliser plusieurs longueur utilisable il n'est pas obligatoire que ce soit un /24

Il est par exemple possible d'assigner les adresses réseaux de cette manière :

203.0.113.0/25 ; 203.0.113.0/26 ; 203.0.113.0/27 ; 203.0.113.0/28 ; 203.0.113.0/28 ; 203.0.113.0/29 ; 203.0.113.0/30 ; 203.0.113.0/31 ; 203.0.113.0/32.

Donc pour calculer le nombre d'adresse utilisable ont fait :

Pour 203.0.113.0/25 = $2^7 - 2 = 126$ adresses utilisables.

Pour 203.0.113.0/26 = $2^6 - 2 = 62$ adresses utilisables.

Pour 203.0.113.0/27 = $2^5 - 2 = 30$ adresses utilisables.

Pour 203.0.113.0/28 = $2^4 - 2 = 14$ adresses utilisables.

Pour 203.0.113.0/29 = $2^3 - 2 = 6$ adresses utilisables.

Pour 203.0.113.0/30 = $2^2 - 2 = 2$ adresses utilisables.

Donc au lieu d'utiliser un réseau en /24 pour les deux routeurs connectés en point à point mieux vaut utiliser le réseau en /30 car il n'y a que 2 adresses à utiliser.

Il existe tout de même une autre possibilité d'adressage avec le sous réseau /31 ainsi :

Pour 203.0.113.0/31 = $2^1 - 2 = 0$ adresses utilisables.

Ici il n'y a pas d'adresses utilisables mais pour une connexion point à point il est possible d'utiliser ce masque de sous réseau exceptionnellement avec pour un routeur avec l'adresse 203.0.113.0/31 et pour le second routeur avec l'adresse : 203.0.113.1/31

Pour 203.0.113.0/32 = $2^0 - 2 = -1$ adresses utilisables ?

Il y a -1 adresses adressable pour un masque en /32 il n'y a jamais besoin d'utiliser un masque en /32 pour configurer une interface cependant il y a une utilité pour ce type d'adresse dans certains cas.

Voici un tableau qui inscrit une notation en décimal et en CIDR :

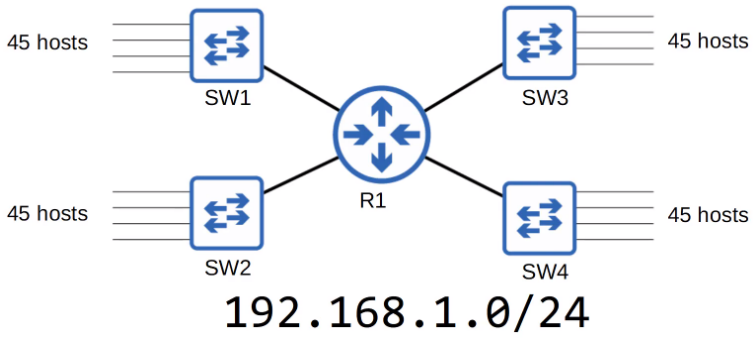
Decimal	Notation CIDR
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28
255.255.255.248	/29
255.255.255.252	/30
255.255.255.254	/31
255.255.255.255	/32

Cours 14 : Sous Réseau (Partie 2)

Dans le cours précédent nous avons vu pourquoi le sous réseau est important.

Dans ce cours nous commencerons par répondre à des questions de sous réseau pratique (comme les réseaux de classes C), nous verrons ensuite des cas pratique avec des sous réseaux de classe B.

Imaginons que l'on ait un réseau avec la topologie suivante :



Il faut ici diviser le réseau 192.168.1.0/24 en 4 sous réseaux qui correspondent aux nombre d'hôte requis.

Il faut déjà déterminer le sous réseau le plus approprié qui permettra d'assigner les 45 hôtes.

Avec un sous réseau en /26 il nous est permis d'avoir en tout 62 adresses utilisable, ce qui correspond à notre besoin ici pour assigner 45 adresses aux hôtes.

Nous commençons donc par assigner l'adresse au premier sous réseau : 192.168.1.0/26

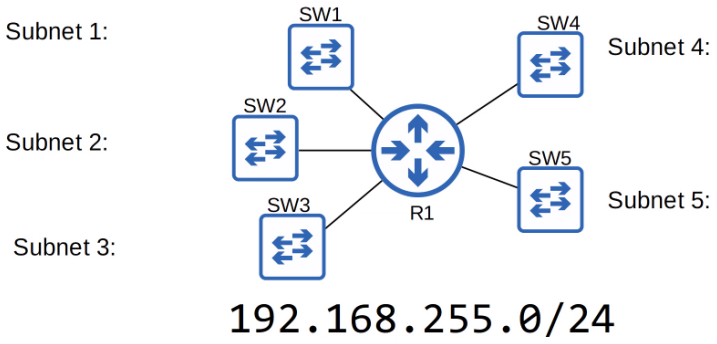
- Pour le réseau 1 192.168.1.0/26 les adresses vont de 192.168.1.0 à 192.168.1.63

à la suite il faut augmenter de 1 l'adresse pour arriver au réseau 2 :

- Pour le réseau 2 l'adresse du réseau est donc : 192.168.1.64/26, les adresses vont de 192.168.1.64 à 192.168.1.127

- Pour le réseau 3 l'adresse du réseau est donc : 192.168.1.128/26, les adresses vont de 192.168.1.128 à 192.168.1.191

- Pour le réseau 4 l'adresse du réseau est donc : 192.168.1.192/26, les adresses vont de 192.168.1.192 à 192.168.1.255



Nouvelle exercice avec cette fois la topologie suivante dans laquelle il faut assigner les adresses des 5 réseaux différents.

Il faut déterminer quelle est le sous réseau approprié qui permettra d'avoir 5 adresses de réseau pour cela ont peut faire rapidement le calcul et trouver le bon masque de sous réseau :

192.168.255.0/27 permettra d'avoir en tout 8 sous réseau puisque $2^3 = 8$ sous réseau possibles.

Les adresses seront

- Le réseau 1 commencera par : 192.168.255.0/27

- Le réseau 2 commencera par : 192.168.255.32/27

- Le réseau 3 commencera par : 192.168.255.64/27

- Le réseau 4 commencera par : 192.168.255.96/27

- Le réseau 5 commencera par : 192.168.255.128/27

Une autre question possible, quelle est le réseau auquel appartient l'adresse 192.168.5.57/27 ?

Pour répondre on peut déjà commencer par convertir l'adresse en binaire et ainsi identifier l'adresse du réseau :

192.168.5.57 = 11000000.10101000.00000101.00111001

255.255.255.224 = 11111111.11111111.11111111.11100000

il suffit à présent de changer à 0 tout les bits se trouvant après l'adresse du masque se qui donne :

192.168.5.32 = 11000000.10101000.00000101.00100000

255.255.255.224 = 11111111.11111111.11111111.11100000

l'adresse du réseau auquel appartient l'adresse 192.168.5.57/27 est donc 192.168.5.32

Répondons à la même question mais cette fois pour savoir à quelle réseau appartient l'adresse 192.168.29.219/29

Pour répondre on convertit déjà l'adresse en binaire puis on peut identifier l'adresse du réseau :

192.168.29.219 = 11000000.10101000.00011101.11011011

255.255.255.248 = 11111111.11111111.11111111.11111000

on change les bits à 0 se trouvant après l'adresse du masque et le convertit en décimal se qui donne :

192.168.29.216 = 11000000.10101000.00011101.11011011

255.255.255.248 = 11111111.11111111.11111111.11111000

L'adresse du réseau auquel appartient l'adresse 192.168.29.219/29 est donc : 192.168.29.216/29

Voici un tableau pour identifier le nombre de réseau et d'hôte possible pour un réseau de classe C :

Longueur du préfixe	Nombre de sous réseau	Nombre d'hôtes
/25	2	126
/26	4	62
/27	8	30
/28	16	14
/29	32	6
/30	64	2
/31	128	0(2)
/32	256	0(1)

Le processus de fonctionnement des sous réseau est exactement le même pour une classe B que pour une classe C.

Nous pouvons donc répondre à la question : Nous avons un réseau en 172.16.0.0/16. Il nous est demandé de créer 80 réseau pour l'entreprise. Quelle longueur de préfixe devrons nous utiliser ?

Pour répondre nous allons faire les calculs comme auparavant pour trouver le nombre de réseau correspondant :

$2^7 - 2 = 126$

172.16.0.0 = 10101100.00010000.00000000.00000000

255.255.254.0 = 11111111.11111111.11111110.00000000

Il est donc nécessaire un masque en /27 qui puisse correspondre au nombre de réseaux associés.

Les réseaux possible seront donc 172.16.0.0 puis 172.16.2.0 puis 172.16.4.0 etc..

Répondons à une question similaire avec cette fois le réseau 172.22.0.0/16. Il nous est demandé de diviser le réseau en 500 sous réseau séparés. Quelle est la longueur de préfixe devrons nous utiliser ?

Faisons le calcul pour trouver le nombre de sous réseaux demandés :

$2^9 - 2 = 510$

172.22.0.0 = 10101100.00010110.00000000.00000000

255.255.255.128 = 11111111.11111111.11111111.10000000

Il est donc nécessaire un masque en /25 qui puisse correspondre au nombre de réseaux associés.

Une Troisième questions avec cette fois le réseau 172.18.0.0/16. Il nous est demandé de diviser le réseau en 250 sous réseaux séparés avec à chaque fois le même nombre d'hôte par sous réseau.

Quelle est la longueur du préfixe pouvons nous utiliser ?

Faisons le calcul pour trouver le nombre de sous réseaux demandés :

2^8 - 2 = 254

172.22.0.0 = 10101100.00010010.00000000.00000000

255.255.255.128 = 11111111.11111111.11111111.00000000

Il est donc nécessaire un masque en /24 qui puisse correspondre au nombre de réseaux associés.

Autre question : à quelle réseau appartient l'adresse : 172.25.217.192/21 ?

172.25.217.192 = 11000000.10101000.00011101.11011011

255.255.248.0 = 11111111.11111111.11111000.00000000

On change tous les bits à 0 se trouvant après l'adresse du masque se qui donne :

172.25.216.0 = 11000000.10101000.00011000.00000000

255.255.248.0 = 11111111.11111111.11111000.00000000

L'adresse 172.25.217.192 appartient donc à l'adresse du réseau 172.25.216.0/21

Voici un tableau pour identifier le nombre de réseau et d'hôte possible pour un réseau de classe B :

Longueur du préfixe	Nombre de sous réseau	Nombre d'hôtes
/17	2	32766
/18	4	16382
/19	8	8190
/20	16	4094
/21	32	2044
/22	64	1022
/23	128	510
/24	256	254
/25	512	126
/26	1024	62
/27	2048	30
/28	4096	14
/29	8192	6
/30	16384	2
/31	32768	0(2)
/32	65536	0(1)

Cours 15 : Sous réseau (Partie 3)

Dans ce cours nous verrons comment fonctionne les sous réseau dans des réseaux de classe A, puis nous aborderons le concept de VLSM (Variable-Length Subnet Masks).

Tout d'abord commençons par répondre à la question suivante :

Il nous a été donné l'adresse de réseau 172.30.0.0/16. L'entreprise a besoin de 100 sous réseau à configurer avec 500 hôtes par sous réseau. Quelle est la longueur de préfixe que l'on devrait utiliser ?

Il suffit de faire le calcul suivant : $2^7 = 128$ réseaux possibles.

Donc en ajoutant 7 bit à l'adresse du masque donnera l'adresse de masque suivant :

172.30.0.0 = 10101100.00011110.00000000.00000000

255.255.255.254 = 11111111.11111111.11111110.00000000

Si l'on veut calculer à présent le nombre d'hôte possible par réseau il suffit de faire le calcul suivant en mettant le nombre de 0 du masque puissance 2 se qui donne :

$2^9 - 2 = 510$ adresses utilisables.

Cela correspond à la demande de 500 utilisateurs par réseau.

Donc le masque sera de /23 ou 255.255.255.254

Répondons à présent à la question suivante :

A quelle réseau appartient l'hôte 172.21.111.201/20 ?

Pour répondre il faut d'abord convertir l'adresse en binaire puis changer à 0 tous les 1 se trouvant après le masque de sous réseau en /20 et reconvertir le tout en décimal se qui donne :

172.21.111.201 = 10101100.00010101.01100000.00000000

255.255.240.0 = 11111111.11111111.11110000.00000000

se qui fais que l'adresse appartient au réseau : 172.21.96.0/20

Répondons à la question : Quelle est l'adresse de Broadcast de l'adresse 192.168.91.78/26 ?

Dans ce cas il faut remplacer tous les Bits étant après le masque par des 1 ce qui donne :

192.168.91.127 = 11000000.10101000.01011011.01111111

255.255.255.172 = 11111111.11111111.11111111.11000000

L'adresse de Broadcast pour l'adresse 192.168.91.78/26 est donc 192.168.91.127/26

Autre question : Le réseau 172.16.0.0/16 a été divisé en 4 sous réseau de taille égal. Il faut identifier l'adresse de réseau et de Broadcast de l'adresse du réseau 2.

Pour répondre nous allons déjà ajouter les 4 réseaux à l'adresse en faisant le calcul : $2^2 = 4$ réseaux.

Se qui donne l'adresse 172.16.0.0/18

Pour trouver l'adresse du réseau 2 il faut ajouter 1 à l'adresse des sous réseau se qui donne l'adresse :

172.16.64.0 = 10101100.00010000.01000000.00000000

Il nous faut à présent trouver l'adresse de Broadcast du réseau 2 pour cela on remplace tous les bits par 1 se trouvant après l'adresse du masque ce qui donne :

172.16.0.0/18 = 10101100.00010000.01111111.11111111

L'adresse du réseau 2 est donc : 172.16.64.0 et l'adresse de Broadcast est 172.16.127.255

Dernière question : Le réseau 172.30.0.0/16 à été divisé en réseau de 1000 hôtes chacun. Combien de sous réseau est il possible de faire ?

Pour répondre il faut faire le calcul suivant : $2^10 - 2 = 1022$ hôtes

Ce qui correspond à dire qu'il y a 6 bits restant dans la partie réseau qui correspondent à nos sous réseau il faut donc faire le calcul suivant pour déterminer le nombre de sous réseau : $2^6 = 64$ adresses de sous réseau possible.

Voyons à présent le processus de sous réseau de classe A.

Les fonctionnement est exactement le même pour des réseau de classe A, B ou C.

Répondons donc à la question : Il nous à été donné l'adresse de réseau : 10.0.0.0/8. Il doit être crée 2000 sous réseaux qui seront distribués dans différentes entreprises. Quelle est la longueur de préfixe que l'on doit utiliser ? Combien d'adresse d'hôte seront sur chaque sous réseau ?

Pour trouver le bon masque de réseau il nous faut savoir comment atteindre un total de 2000 sous réseaux. Pour cela on fais le calcul suivant : $2^11 = 2048$

Ce qui donnerait l'adresse suivante :

10.0.0.0 = 00001010.00000000.00000000.00000000

255.255.224.0 = 11111111.11111111.11100000.00000000

Pour déterminer le nombre d'hôte par réseau il faut juste mettre en puissance de 2 le nombre de 0 se trouvant dans le masque se qui donne : $2^13 - 2 = 8190$ hôtes par sous réseau

La longueur de préfixe que l'on doit utiliser est donc /19 et il y aura 8190 hôtes par réseau.

Répondons à la question : PC1 à une adresse IP de 10.217.182.223/11, identifier le réseau pour le PC1 et le nombre d'hôte de l'adresse.

On fais le calcul suivant pour trouver le nombre d'hôtes : $2^21 - 2 = 2\ 097\ 150$ hôtes par réseau

Voici donc les réponses aux question :

1) Adresse du réseau : 10.192.0.0/11
2) Adresse de Broadcast : 10.223.255.255/11 3) Première adresse utilisable : 10.192.0.1/11 4) Dernière adresse utilisable : 10.223.255.254/11 5) Nombre d'hôtes par réseau : 2 097 150

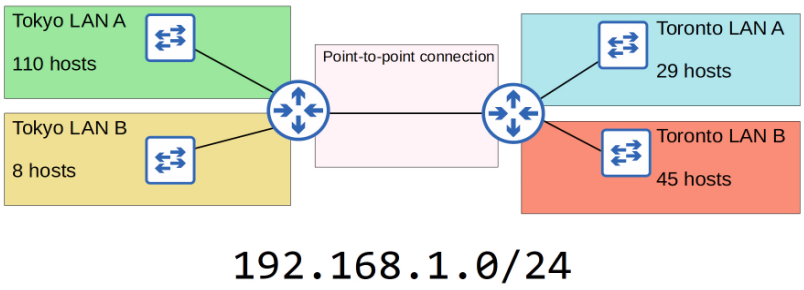
Voyons à présent le concept de VLSM (Variable-Length Subnet Masks).

Jusqu'à présent nous avons utilisé le réseau en pratiquant la méthode FLSM (Fixed-Length Subnet Masks) cela signifie que chaque sous réseau utilise la même longueur de préfixe.

VLSM est le processus de créer des sous réseau de différentes taille pour rendre le réseau d'adresse plus efficace.

VLSM est plus compliqué que FLSM mais il suffit de bien suivre chaque étape.

Voici donc un exemple d'une topologie :



Voici donc les étapes pour faire un réseau en utilisant VLSM :

1) Assigner le plus grand réseau utilisant le plus d'hôtes au départ de l'espace d'adresse réseau

2) Assigner le second plus grand réseau utilisant le plus d'hôtes

3) Répéter le processus jusqu'à ce que chaque réseau soit assigné.

Dans le cas de notre topologie Nous commencerons donc par le réseau Tokyo LAN A avec les 110 hôtes puis le Toronto LAN B avec les 45 hôtes puis le Toronto LAN A avec les 29 hôtes puis le Tokyo LAN B avec les 8 hôtes puis finir par la connexion entre les deux routeurs en point à point.

Donc commençons par l'adresse de Tokyo LAN A :

Adresse du réseau : 192.168.1.0/25

Adresse de Broadcast : 192.168.1.127/25

Première adresse utilisable : 192.168.1.1/25

Dernière adresse utilisable : 192.168.1.126/25

Nombre total d'adresse hôte utilisable : 126

Puis pour Toronto LAN B :

Adresse du réseau : 192.168.1.128/26

Adresse de Broadcast : 192.168.1.191/26

Première adresse utilisable : 192.168.1.129/26

Dernière adresse utilisable : 192.168.1.190/26

Nombre total d'adresse hôte utilisable : 62

Pour Toronto LAN A :

Adresse du réseau : 192.168.1.192/27

Adresse de Broadcast : 192.168.1.223/27

Première adresse utilisable : 192.168.1.193/27

Dernière adresse utilisable : 192.168.1.222/27

Nombre total d'adresse hôte utilisable : 30

Pour Tokyo LAN B :

Adresse de réseau : 192.168.1.224/28

Adresse de Boradcast : 192.168.1.239/28

Première adresse utilisable : 192.168.1.225/28

Dernière adresse utilisable : 192.168.1.238/28

Nombre total d'adresse hôte utilisable : 14

Pour la connexion entre routeurs point à point :

Adresse du réseau : 192.168.1.240/30

Adresse de Broadcast : 192.168.1.243/30

Première adresse utilisable : 192.168.1.241/30

Dernière adresse utilisable : 192.168.1.242/30

Nombre total d'adresse hôte utilisable : 2

Cours 16 : Vlans (Partie 1)

Dans ce cours nous parlerons des Vlan (Virtual Local Area Network) voici les choses que nous verrons dans ce cours : nous commencerons déjà par revoir ce qu'est un LAN, puis qu'est qu'un domaine de Broadcast, après cela nous verrons ce qu'est le Vlan et son but puis nous verrons comment configurer un Vlan sur un switch Cisco.

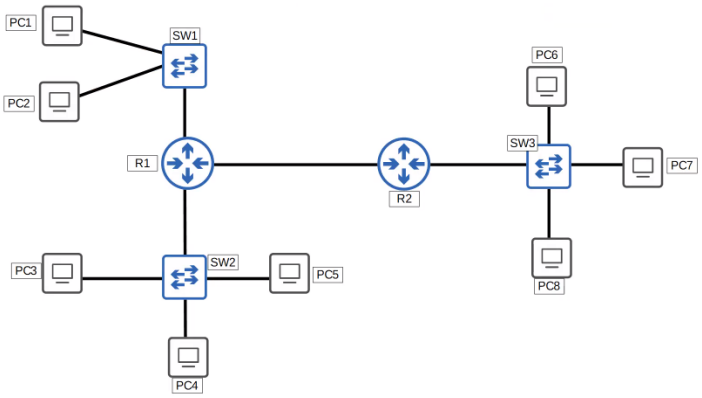
Qu'est ce qu'un LAN ?

Un LAN est un groupe d'appareil (PC, Serveurs, routeurs, switchs, etc..) qui est placé dans une seule localisation (maison, entreprise, etc..)

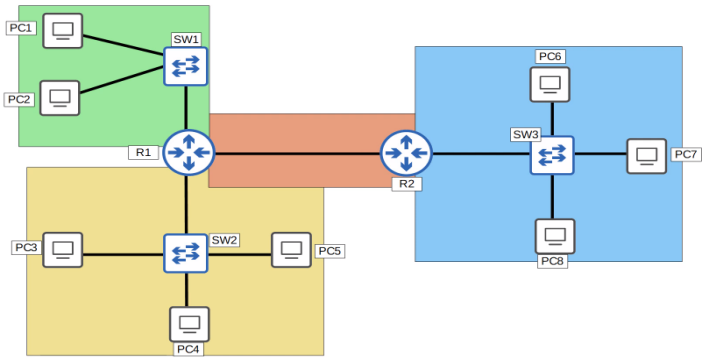
Une définition plus spécifique du LAN est qu'il a un seul domaine de broadcast qui inclus tous les appareils dans ce domaine de broadcast.

Un domaine de broadcast est un groupe d'appareils qui reçoit des trames de broadcast (destination MAC FFFF.FFFF.FFFF) envoyé par un autre membre du même groupe d'appareil.

Si l'on prend la topologie suivante, quelle est le nombre de domaines de Broadcast présent ?



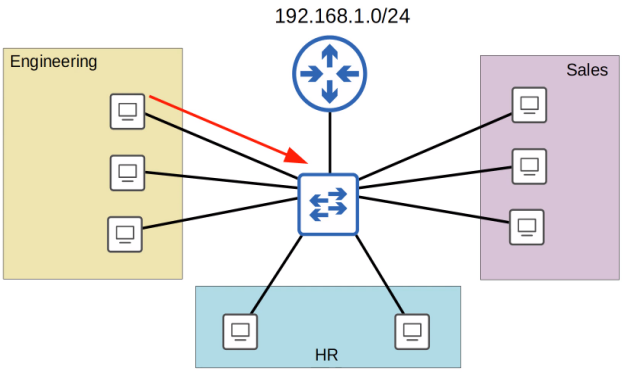
Il y a en tout 4 domaines de Broadcast dans lesquelles les appareils qui envoient des requêtes de Broadcast du même groupe membre :



1.

2.

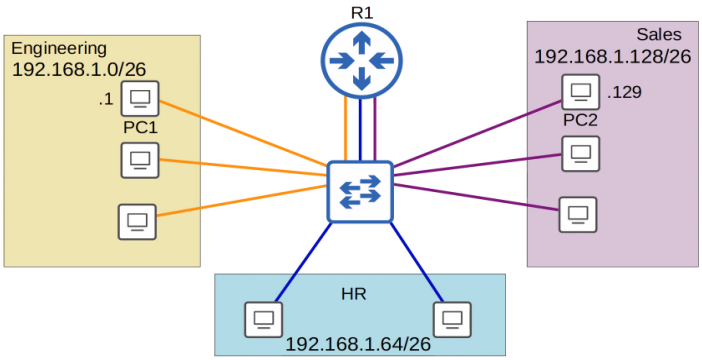
A présent voyons de qu'est qu'un Vlan, pour cela utilisons la topologie suivante :



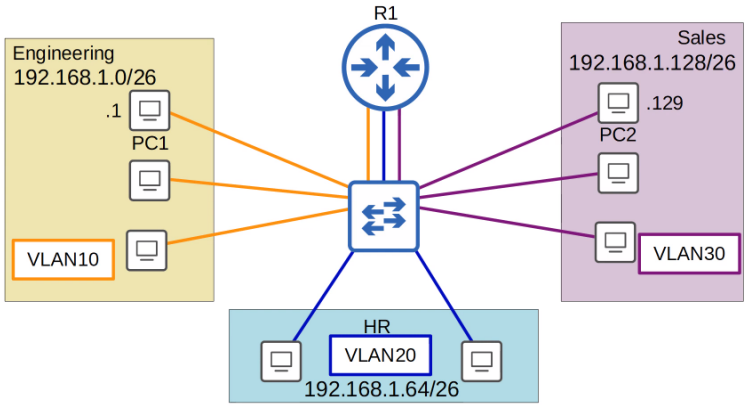
Lorsqu'un appareil sur cette topologie veut envoyer une requête vers le routeur se sont tous les appareils du service qui vont recevoir la requête envoyé en Broadcast puisqu'il s'agit du même domaine de Broadcast. Cela peut causer des réduction des performances du réseau mais aussi causer des problèmes de sécurité car même en configurant des politique de sécurité cela ne sera pas bien fonctionnel car la sécurité est surtout pour régler le trafique provenant de l'extérieur du réseau et non pas la sécurité du même LAN.

La solution la plus adapté est de séparer chaque réseau en plusieurs Vlans.

Nous avons donc séparé les trois départements de l'entreprise en 3 sous réseaux de la manière suivante :



Les 3 départements de l'entreprise sont à présent séparés en 3 sous réseaux mais ils restent toujours connectés au même domaine de Broadcast ce qui fais que lorsqu'un PC envoie une requête c'est toujours les autres départements qui reçoivent la requête. Nous allons donc créer différentes Vlan pour simuler les LAN virtuellement de cette manière :



De cette manière les Switch sont séparés et ne peuvent plus partager le trafique entre Vlans ce qui inclus des requêtes Broadcast/Unicast, les requêtes passent obligatoirement vers le Routeur qui redistribue le trafique vers le Vlan concerné.

En résumé les Vlan sont configuré sur un switch par une interface basique.

Les Vlan sont séparé à la couche 2. Les switch ne partagent pas le trafique directement entre hôtes et différentes Vlans.

A présent voici comment se fais la configuration des Vlans sur un Switch :

Commençons par vérifier quelle est la configuration par défaut sur un Switch en lançant la commande :

```
SW1#show vlan brief
```

Voici les commandes pour assigner une interface à un Vlan :

```
SW1(config)#interface range g1/0-3
```

Cette commande est utilisé pour sélectionner plusieurs interface d'un seul coup.

```
SW1(config-if-range)#switchport mode access
```

Cette commande est utilisé pour lancer le mode access, le mode access est un port qui désigne un seul Vlan et connecte plusieurs hôtes comme les PC.

```
SW1(config-if-range)#switchport access vlan 10
```

Cette commande est utilisé pour faire accéder la ou les interface vers le Vlan voulus ici le vlan 10.

Le vlan est crée automatiquement s'il n'avait pas été crée auparavant.

Il est aussi possible d'utiliser une commande pour créer la vlan manuellement :

```
SW1(config)#vlan 10
```

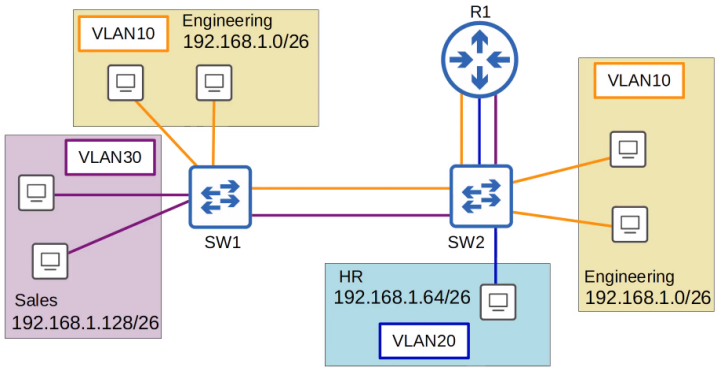
Une fois la vlan crée il est aussi possible de lui donner un nom avec la commande par exemple :

```
SW1(config)#name ENGINEERING
```

Cours 17 : Vlans (Partie 2)

Dans ce cours nous allons continuer l'apprentissage à propos des Vlan (Virtual Local Area Network). Tout d'abord nous verrons se qu'est qu'un port Trunk et son but, puis se qu'est l'encapsulation 802.1Q, nous verrons ensuite comment configurer un port Trunk et se qu'est « Router on a Stick » (ROAS).

Voici la topologie réseau que nous utiliserons dans ce cours :



Il n'y a pas de lien en Vlan20 entre SW1 et SW2. Ceci car il n'y a pas de PC dans le Vlan 20 connectés au SW1. Les PC dans le Vlan 20 peuvent tout de même joindre les PC connectés au SW1. R1 fonctionne avec un routage inter-vlan c'est à dire que lorsqu'un pc veut communiquer avec un pc se trouvant sur un autre vlan c'est le routeur qui va rediriger le trafique vers le vlan adapté.

Dans un petit réseau avec peu de Vlan il est possible d'utiliser des interfaces déparés pour chaque Vlan lorsque l'on connecte des switchs entre eux, et les switchs aux routeurs.

Lorsqu'il y a beaucoup de vlan il n'est plus vraiment possible de connecter une interface par vlan car il n'y aura plus assez d'interface pour chaque Vlan.

Une solution est d'utiliser le Trunk pour que le trafique soit multiplié en plusieurs Vlans sur une seule interface.

Imaginons qu'un PC de Vlan 10 veuille envoyer un paquet vers un autre PC sur la Vlan 10, le trafique ne passera pas par le routeur obligatoirement il peut passer directement vers le SW1 pour distribuer le paquet mais une fois au SW1 comment le Switch saura t-il sur quelle interface distribuer le paquet ?

La réponse est en marquant la trame envoyé sur le lien trunk cela permet d'identifier et savoir à quelle Vlan la trame appartient.

Il existe deux principaux protocoles de trunk : ISL (Inter-Switch Link) et IEEE 802.1Q

ISL est un ancien protocole Cisco propriétaire crée avant le standard industrie : IEEE 802.1Q

IEEE 802.1Q est un protocole de standard industrie crée par le IEEE (Institute of Electrical and Electronics Engineers). On utilisera probablement jamais ISL dans le monde réel même si les Cisco moderne le supporte.

Le marquage du type de protocole de trunking est inséré dans l'entête Ethernet entre l'adresse IP source et le Type de paquet/longueur du paquet :

Le tag est de 4 bytes (32 bits) de longueur ; ce marquage consiste en deux parties principales :

- Tag Protocol Identifier (TPID) et le Tag Control Information (TCI)

Le TCI consiste en 3 sous parties.

Voici une capture du format 802.1Q :

802.1Q tag format

16 bits	3 bits	1 bit	12 bits
TPID	TCI		
	PCP	DEI	VID

- Le TPID est de 16 bits (2 bytes) de longueur, la valeur indiqué est toujours : 0x8100 en hexadécimal. Cela indique la trame marqué par le 802.1Q.

- Le PCP (priority Code Point) est de 3 bits de longueur. Et utilisé pour le Class of Service (COS) qui priorise le trafique important dans un réseau.

- Le DEI (Drop Eligible Indicator) est de 1 bit de longueur et utilisé pour indiquer la trame qui peut être lâché si le trafique réseau est trop important.

- Le VID est d'une longueur de 12 bits, il permet d'identifier le Vlan et à quelle trame il appartient. 12 bits de longueur = 4096 Vlan en tout (2^{12}), se qui fait un classement de 0 – 4095

Les Vlan 0 et 4095 sont réservés et ne peuvent pas être utilisés, le classement des Vlan se fais donc de 1 – 4094.

Le protocole ISL Cisco à aussi un classement Vlan de 1 – 4094

Le classement des Vlan (1- 4094) est divisé en deux sections :

Normal Vlan : 1 – 1005

Vlan étendus : 1006 – 4094

Certains anciens appareils n'utilisent plus le classement Vlan, même s'il est plus sûre de s'attendre qu'un Switch moderne va supporter un classement Vlan étendu.

Donc si l'on reprend la question du départ avec notre topologie réseau et qu'un PC se trouvant dans le Vlan 10 veut envoyer une donnée à un autre PC se trouvant dans le Vlan 10, il va directement passer par le Switch sans passer par le routeur et le SW1 va distribuer le trafique directement au Vlan adapté grâce au marquage inscrit dans le l'entête Ethernet.

802.1Q à une fonctionnalité appelé native VLAN (ISL ne possède pas cette fonction) Le Vlan Natif est le Vlan 1 par défaut sur tous les ports trunk, il peut être configuré manuellement sur chaque port Trunk. Le Switch n'ajoute pas un tag 802.1Q à la trame native du Vlan. Lorsqu'un switch reçoit une trame non marqué dans un port trunk, il l'attribue au Vlan natif.

Il est donc important que les Vlan natif soient correspondant entre switchs sinon le paquet risque d'être mal distribué.

Voici à présent les commandes nécessaire pour la configuration d'un interface en mode trunk :

```
SW1(config)#interface g0/0
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
```

Pour manuellement configurer une interface comme port trunk, il faut d'abord placé l'encapsulation en 802.1Q ou ISL. Sur les switchs qui supportent uniquement 802.1Q cela n'est pas nécessaire.

Une fois l'encapsulation mis en place il faut configurer l'interface en trunk.

On peut ensuite utiliser la commande suivante pour vérifier la configuration de l'interface en mode trunk :

```
SW1#show interfaces trunk
```

Voici les commandes pour ajouter la permission d'une vlan pour un trunk :

```
SW1(config)#int g0/0
SW1(config-if)#switchport trunk allowed vlan add 20
```

Pour supprimer la permission d'une Vlan sur un trunk on lance la commande :

```
SW1(config)#int g0/0
SW1(config-if)#switchport trunk allowed vlan remove 20
```

Pour ajouter jouter la permission de toutes les vlan pour le trunk on lance la commande :

```
SW1(config-if)#switchport trunk allowed vlan all
```

On peut aussi excepter certaines Vlans de la permission à toutes les Vlans en lançant la commande :

```
SW1(config-if)#switchport trunk allowed vlan except 1-5,10
```

On peut empêcher toute les Vlan d'utiliser le mode trunk en lançant la commande :

```
SW1(config-if)#switchport trunk allowed vlan none
```

Pour plus de sécurité il est conseillé de changer le Vlan natif à un Vlan inutilisé tout en étant sûre que la vlan native est la même entre les switchs.

La commande pour changer le Vlan Natif est :

```
SW1(config-if)#switchport trunk native vlan 1001
```

Au lieu d'utiliser 3 interface différentes pour connecter le switch au routeur 1 comme dans la topologie, il est possible d'utiliser un seul câble interface et de configurer des « sub-interface » afin que le trafique soit redirigé vers les trois Vlan et fonctionner comme 3 interfaces séparés.

Voici les commandes pour configurer des « sub-interface » sur le routeur1 :

```
R1(config)#interface g0/0
R1(config-if)#no shutdown
R1(config-if)#interface g0/0.10
R1(config-if)#encapsulation dot1q 10
R1(config-if)#ip address 192.168.1.62 255.255.255.192
```

On peut vérifier la configuration des sub-interface avec la commande :

```
R1#show ip interface brief
```

ROAS est utilisé pour faire le routage entre plusieurs Vlan en utilisant une seule interface dans un routeur et un switch. L'interface switch est configuré comme un trunk regular. L'interface de routage est configuré utilisant des subinterfaces. On configure le marquage Vlan et l'adresse IP sur chaque subinterface. Le routeur à la trame qui arrive avec un certain marquage Vlan configuré avec le tage Vlan. Le routeur va marquer la trame envoyé vers la sub-interface avec le marquage Vlan configuré sur la sub-interface.

Cours 18 : Vlans (Partie 3)

Dans ce cours nous verrons quelques exemple de native Vlan sur un routeur, puis nous ferons des analyse Wireshark. Nous verrons ensuite plus en détail la couche 3 avec les Switch/multicouche Switch.

Il existe 2 méthodes pour configurer un native Vlan sur un routeur :

- utiliser la commande :

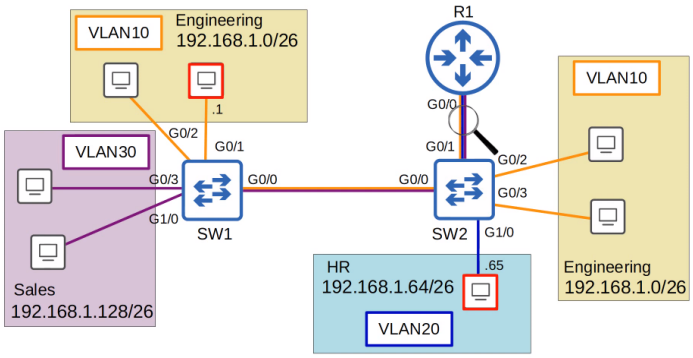
```
SW1(config)#int g0/0.10
SW1(config-subif)#encapsulation dot1q {vlan-id} native
```

- configurer l'adresse IP pour le Vlan natif sur un routeur avec l'interface physique (la commande encapsulation dot1q vlan id n'est pas nécessaire)

Voici les commandes nécessaires pour configurer avec la deuxième méthode :

```
R1(config)#no interface g0/0.10
R1(config)#interface g0/0
R1(config-if)#ip address 192.168.1.62 255.255.255.192
```

utilisons à présent la topologie suivante :



Imaginons que le PC du vlan 20 veuille envoyer un paquet au PC du vlan 10. Voici une capture Wireshark prise lors de l'envoie de la trame :

La trame est d'abord envoyé au Routeur 1 :

```
> Frame 104: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
v Ethernet II, Src: 0c:bd:ad:00:70:00 (0c:bd:ad:00:70:00), Dst: 0c:bd:ad:c5:08:00 (0c:bd:ad:c5:08:00)
  > Destination: 0c:bd:ad:c5:08:00 (0c:bd:ad:c5:08:00)
  > Source: 0c:bd:ad:00:70:00 (0c:bd:ad:00:70:00)
    Type: 802.1Q Virtual LAN (0x8100)
v 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 20
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0001 0100 = ID: 20
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.1.65, Dst: 192.168.1.1
> Internet Control Message Protocol
```

Puis le routeur 1 redirige le trafique au SW2 :

```
> Frame 105: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
v Ethernet II, Src: 0c:bd:ad:c5:08:00 (0c:bd:ad:c5:08:00), Dst: 0c:bd:ad:84:0a:00 (0c:bd:ad:84:0a:00)
  > Destination: 0c:bd:ad:84:0a:00 (0c:bd:ad:84:0a:00)
  > Source: 0c:bd:ad:c5:08:00 (0c:bd:ad:c5:08:00)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.1.65, Dst: 192.168.1.1
> Internet Control Message Protocol
```

Nous avons vu jusqu'à présent des Switch de couche 2 qui redistribuaient le trafique, voyons à présent les Switch de couche 3.

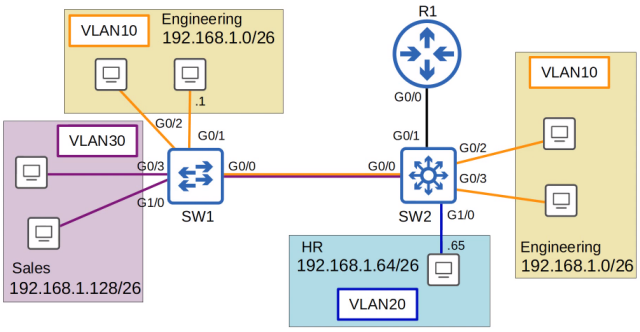
Layer 2 switch

Layer 3 switch

Un switch multicouche est capable de faire du switch et du routage.

On peut assigner une adresse IP à son interface comme un routeur. On peut aussi créer une interface virtuel pour chaque Vlan et assigner une adresse IP à ces interfaces. On peut configurer le routage sur ceux ci comme un routeur. Il peut également être utilisé pour du routage Inter-Vlan.

Changeons à présent la topologie réseau du SW2 par un Switch multicouche :



Pour que le PC du Vlan 20 puisse envoyer une trame, il a été dis jusqu'à présent qu'il fallait qu'il passe par le Routeur 1. Mais comme un routeur de couche 3 est utilisé il est possible de l'utiliser comme routeur et donc éviter que le trafique ne passe par le routeur 1.

Pour cela il faut utiliser SVI (Switch Virtual Interface) se sont des interfaces virtuels que l'on peut assigner une adresse IP à un switch multi-couche.

Configurer chaque PC pour utiliser un SVI (non pas un routeur) comme adresse de passerelle.

Pour envoyer le trafique sous différents sous réseau/Vlans, les PC vont envoyer le trafique au Switch qui va le router ou rediriger le trafique.

Nous allons donc reconfigurer la topologie réseau pour que le trafique soit diriger vers le Switch de couche 3 :

Commençons par le routeur 1 en retirant les sub interfaces :

```
R1(config)#interface g0/0
R1(config-if)#ip address 192.168.1.194 255.255.255.252
```

Puis en configurant le SW2 :

```
SW2(config)#default interface g0/1
SW2(config)#ip routing
SW2(config)#interface g0/1
SW2(config-if)#no switchport
SW2(config-if)#ip addresse 192.168.1.193 255.255.255.252
```

On configure ensuite le routage avec les commandes :

```
SW2(config-if)#exit
SW2(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.194
```

Pour configurer les SVI sur un Switch multicouche on lance les commandes :

```
SW2(config)#interface vlan10
SW2(config-if)#ip address 192.168.1.62 255.255.255.192
SW2(config-if)#no shutdown
SW2(config-if)#interface vlan 20
SW2(config-if)#ip address 192.168.1.126 255.255.255.192
SW2(config-if)#no shutdown
SW2(config-if)#interface vlan 30
SW2(config-if)#ip address 192.168.1.190 255.255.255.192
SW2(config-if)#no shutdown
```

Il faut d'abord créer le Vlan sur le Switch pour que celui ci fonctionne correctement.

Le Switch doit avoir au moins un access port dans le Vlan en état up/up ; et/ou un port trunk qui autorise le Vlan en l'état up/up. Le vlan ne doit pas être éteint. Le SVI ne doit pas être éteint non plus pour que le trafique soit correctement dirigé (Les SVI sont désactivés par défaut)

A présent que la configuration est faite, lorsqu'un PC veut envoyer une trame vers un autre PC se trouvant dans un autre Vlan, le trafique passera seulement par le Switch multicouche et non plus sur le routeur 1.

Cours 19 : DTP/VTP

Dans ce cours nous parlerons de deux protocole propriétaire : DTP (Dynamic Trunking Protocol) et VTP (Vlan Trunking Protocol). Ce sont des protocoles qui ne fonctionnent que sur du matériel Cisco.

DTP est un protocole propriétaire Cisco qui permet aux Switchs Cisco de dynamiquement déterminer l'état des interface (access ou trunk) sans faire de configuration manuelle.

DTP est activé par défaut sur toutes les interfaces des Switchs Cisco.

A départ on configurait les modes en utilisant les commandes : *switchport mode access* ou *switchport mode trunk*.

Pour des raisons de sécurité la configuration manuelle est recommandé. DTP devrait être désactivé sur tous les ports switch.

Voici les commandes nécessaires pour activer le mode DTP :

```
SW2(config-if)#switchport mode dynamic negotiation parameter to AUTO
```

Ou bien :

```
SW2(config-if)#switchport mode dynamic negotiation parameter to DESIRABLE
```

Un switchport en dynamic desirable va former un trunk si d'autres Switchs sont connecté à une interface dans les modes suivants :

- switchport mode trunk - switchport mode dynamic desirable - switchport mode dynamic auto

Si l'un des deux switch est en mode dynamic desirable et que celui connecté est en mode dynamic auto, le trunk va se former aussi car c'est le mode desirable qui s'active et le mode auto va suivre.

Si l'un des deux Switch est en mode dynamic desirable et que celui connecté est en mode access c'est le mode access qui est privilégié.

Si l'un des deux Switch est en mode trunk et que celui connecté est en mode access le trafique réseau sera non effectif.

Lorsque le mode dynamic auto est activé le switch ne va pas essayer de former un trunk avec d'autres appareils Cisco, il va former le trunk seulement si l'autre appareil connecté veut en former un, et donc si l'interface connecté est dans un des deux modes suivant :

switchport mode trunk

switchport mode dynamic desirable

Voici une commande pour vérifier les modes activés sur l'interface du Switch :

```
SW1#show interfaces g0/0 switchport
```

Sur d'ancien Switch le mode *switchport mode dynamic desirable* est le mode d'administration par défaut. Sur les nouveau Switch *switchport mode dynamic auto* est le mode par défaut.

Il est possible de désactiver le mode DTP sur une interface en lançant la commande :

switchport nonegotiate

Configurer une interface avec le mode *switchport mode access* désactive aussi une interface en mode DTP.

Les Switch qui supportent 802.1Q et ISL trunk encapsulation peuvent utiliser DTP pour négocier la négociation qu'ils utilisent.

Cette négociation est activé par défaut, le mode trunk par défaut est :

switchport trunk encapsulation negotiate

ISL est favori par rapport à 802.1Q donc si les deux switch supportent ISL c'est ISL qui sera sélectionné.

Les trames DTP sont envoyés dans le Vlan1 lorsque ISL est utilisé, ou le Vlan natif lorsque c'est 802.1Q qui est utilisé (le vlan natif est Vlan1)

Voyons à présent comment fonctionne VTP (Vlan Trunking Protocol), VTP permet de configurer les Vlans sur un serveur VTP et les autres Switchs vont synchroniser leurs base de données de Vlan au serveur.

Il est conçu pour de grands réseaux contenant beaucop de Vlan, donc il n'est pas nécessaire de configurer chaque Vlan sur chaque Switch.

Il est rarement utilisé et il est recommandé de ne pas l'utiliser.

Il existe 3 versions de VTP : 1, 2 et 3

Il y a 3 mode de VTP : server, client, transparent

Les Switch Cisco fonctionnent avec le mode VTP server par défaut.

Le mode VTP server :

Ce mode peut ajouter/modifier/supprimer une Vlans. Il stocke les base de données Vlan dans la ram non-volatile (NVRAM). Il augmente le nombre de révision chaque fois qu'une Vlan est ajouté/modifié/supprimé. Il avertis la dernière version de la base de donnée de Vlan sur une interface trunk, et le client VTP synchronise ses bases de données VLAN au serveur.

Les serveurs VTP fonctionnent aussi en tant que VTP clients. Par contre un serveur VTP va synchroniser à un autre serveur VTP qui est à un plus grand niveau de révision.

Le mode VTP clients :

Ne peut pas ajouter/modifier/supprimer de Vlans. Ne stocke pas la base de donnée Vlan dans la NVRAM (dans VTPv3 il le fais). Il synchronise les bases de données Vlan au serveur avec le plus haut numéro de révision dans le domaine VTP. Il avertis les base de données Vlan et redirige les avertissements VTP aux autres clients à travers leurs ports trunk. Pour activer le mode client :

```
SW2(config)#vtp mode client
```

Voici la commande pour vérifier l'état du VTP :

```
SW1#show vtp status
```

Voici la commande pour faire joindre un switch vers un nom de domaine VTP :

```
SW1(config)#vtp domain cisco
```

Si l'on configure le Switch avec les commandes suivantes

```
SW1(config)#vlan 10
SW1(config-vlan)#name engineering
SW1(config-vlan)#exit
```

et que l'on relance la commande :

```
SW1#show vtp status
```

On remarquera que le nombre de révision à augmenté.

Lorsque plusieurs Switch sont connectés entre eux et que l'un des switch à cette configuration d'active avec le nom de domaine VTP etc.. il y a une synchronisation qui est faite automatiquement avec les autres Switch, le nom de domaine est automatiquement configuré et les vlan sont automatiquement ajoutés.

Il existe tout de même des dangers à utiliser VTP : Si l'on connecté un ancien Switch avec un haut numéro de révision au réseau, tous les Switch du domaine vont synchroniser leurs base de données de Vlan à ce Switch et tous les Switch connectés perdront leurs configuration automatiquement car ils se synchroniseront avec le Switch nouvellement connecté.

Le mode VTP transparent :

Ne participe pas au domaine VTP (et ne synchronise pas les base de données de Vlan)

Il maintient ses propres bases de données Vlan dans le NVRAM. Il peut ajouter/modifier/supprimer les Vlans, mais ils ne seront pas avertis aux autres Switch.

Il va rediriger les avertissements VTP qui sont du même domaine que lui.

Pour utiliser le mode transparent on lance la commande :

```
SW2(config)#vtp mode transparent
```

Le mode VTP V2 n'est pas vraiment différent que la version VTP V1. La différence majeur est que la version 2 supporte le Token Ring Vlan.

Cours 20 : STP (Partie 1)

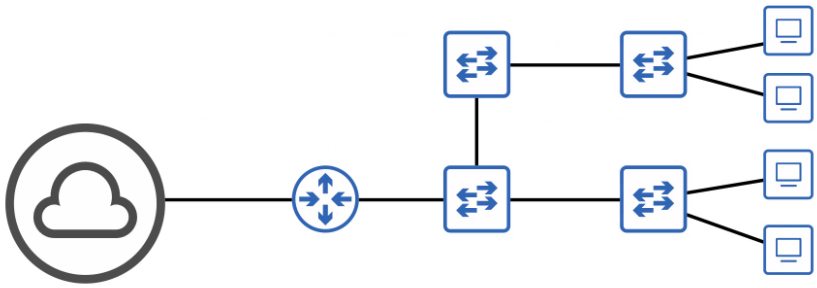
Dans ce cours nous apprendrons ce qu'est le protocole réseau STP (Spanning Tree Protocole)

Nous verrons d'abord le sujet des redondances dans le réseau.

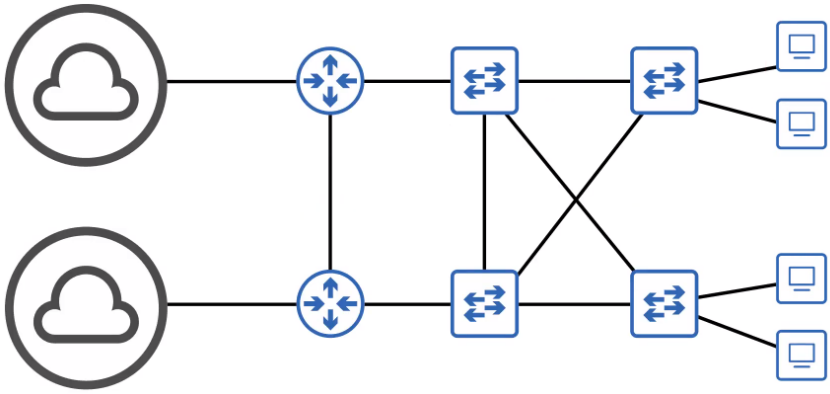
La redondance est une partie essentiel d'un réseau, les réseaux modernes sont sensés fonctionner 24/7/365. Même une simple arrêt temporaire peut être désastreux pour une entreprise.

Si un composant d'un réseau n'est plus fonctionnel, il faut s'assurer que d'autres composants prennent le relais en un minimum de temps. Il faut ajouter de la redondance autant que possible sur le réseau.

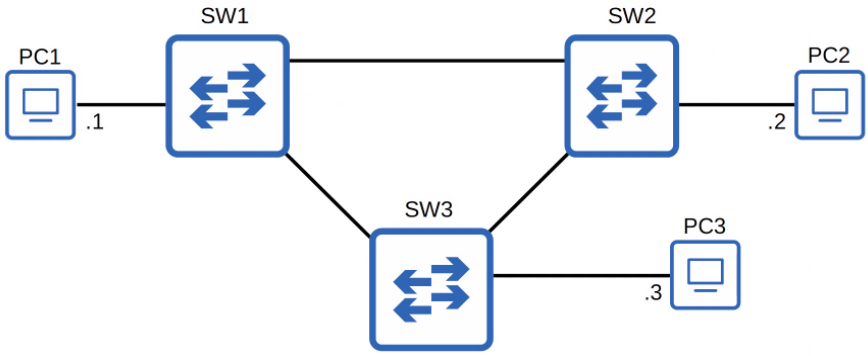
Voici une topologie qui n'est pas très fiable en terme de redondance car si une connexion est interrompue il n'y a pas de relais pour faire fonctionner le réseau



Voici donc une autre topologie plus efficace et qui permet une meilleure redondance lorsqu'une connexion est interrompue :



Cette topologie permet un bonne redondance mais présente de même un risque appelé : « les tempêtes de Broadcast » comprenons mieux ce risque avec ce schéma :



Sur ce schéma lorsque PC1 envoie une requête au PC2, celle ci est diffusé en Broadcast sur toutes les interfaces des Switchs afin de faire connaître son adresse MAC, une fois la requête reçue une requête Unicast est renvoyé au PC1. Le problème sur cette topologie est que même après la requête Unicast reçus, il y a toujours des requêtes qui sont échangés entre le SW2 et le SW3 et donc les requêtes en Broadcast continuent d'être envoyés sur le réseau indéfiniment, c'est ce que l'on appelle une tempête de Broadcast.

La congestion du réseau n'est pas le seul problème. Chaque fois qu'une trame arrive sur un port Switch, le

Switch utilise la source MAC pour apprendre l'adresse MAC et mettre à jour sa table d'adresse MAC. Lorsque la trame avec la même source MAC arrive sur différentes interface, le Switch va continuellement mettre à jour son interface dans sa table d'adresse MAC. C'est connu sous le nom de MAC Address Flapping.

Le protocole Spanning Tree est une solution efficace à ce type de problème.

Le protocole Spanning tree classique est pris du standard : IEEE 802.1D

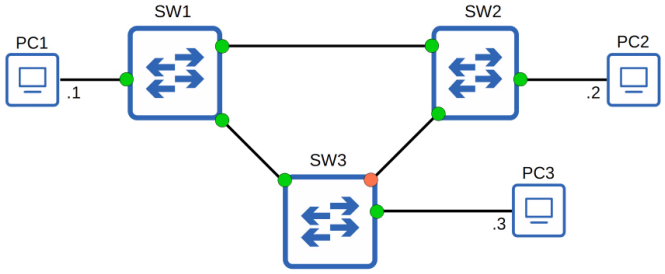
Les switch de tous les vendeurs lancent STP par défaut. STP empêche les boucles de couche 2 en bloquant un port, principalement en désactivant une interface.

Cette interface fonctionne ensuite comme Backup qui peut entrer dans un état actif si une interface active ne fonctionne plus.

Les interfaces en état actif fonctionnent normalement. Les interfaces en état bloqués ou désactivés envoient ou reçoivent seulement des messages STP (appelés BPDUs = Bridge Protocol Data Units)

Avant que ne viennent les Switch, les Hub étaient utilisés avec un pont pour partager les données aux appareils voulus.

Voici à quoi pourrait ressembler la topologie une fois le spanning tree activé, on peut voir qu'il y a l'état bloqué en orange pour l'interface du switch 3, cela permet d'empêcher les tempêtes de Boadcast.



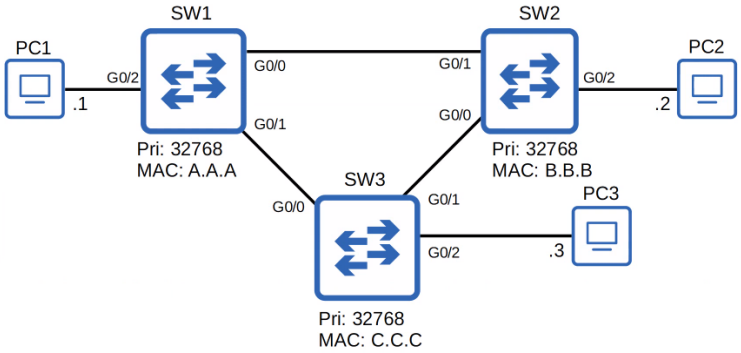
Allons plus en détail à présent pour comprendre le protocole Spanning tree, en sélectionnant quelle port va partager le réseau et quelle port sera bloqué, STP crée un seul chemin vers/depuis chaque point du réseau. Cela empêche de provoquer des boucles de couche 2.

Il y a un tas de processus que STP utilise pour déterminer quelle port devrait partager et lequelle devrait être bloqué. STP active les Switch pour envoyer/recevoir des requête Hello BPDU en dehors de toutes ses interfaces avec un temps par défaut de 2 secondes (Le Switch envoie des BPDU sur toutes les interfaces toutes les deux secondes). Si un Switch reçoit un BPDU Hello sur une interface, il sait que cette interface est connecté à un autre Switch (les routeurs, PC etc.. n'envoient pas de BPDU)

Les Switchs utilisent une seule partie dans les STP BPDU, la section Bridge ID, pour élire un « root bridge »pour le réseau. Le Switch avec le Bridge ID le plus bas devient le root Bridge.

Tous les ports du root bridge sont dans l'état de partage et les autres switch dans la topologie doivent avoir le chemin pour joindre le root Bridge.

Le bridge de priorité par défaut est 32768 sur tous les Switch, donc par défaut l'adresse MAC est utilisé pour déterminer quel switch est le root bridge (L'adresse MAC la plus basse devient le root bridge)

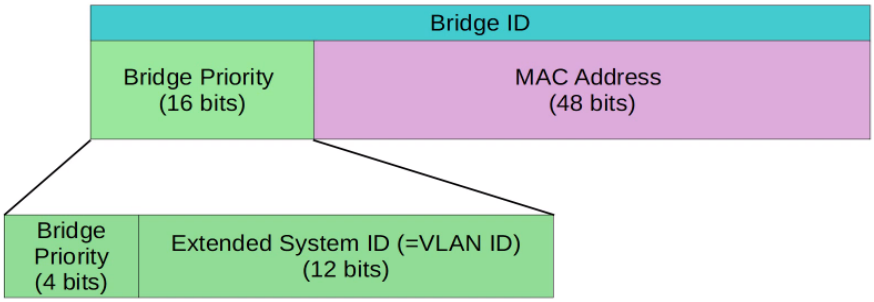


Voici donc la topologie une nouvelle fois avec cette fois les Bridge ID et le numéro de priorité.

Sur cette topologie tous les Switch ont une priorité égal qui est 32768, mais l'adresse la plus basse est celle du SW1, c'est donc le SW1 qui devient le root bridge.

Les Switch Cisco utilisent une version de STP appelé PVST (Per-Vlan Spanning Tree) PVST lance des instances STP séparés sur chaque Vlan, donc chaque interface de Vlan différentes peut être partagé/bloqué.

Voici à quoi devrait ressembler un Bridge ID sur une trame Ethernet :



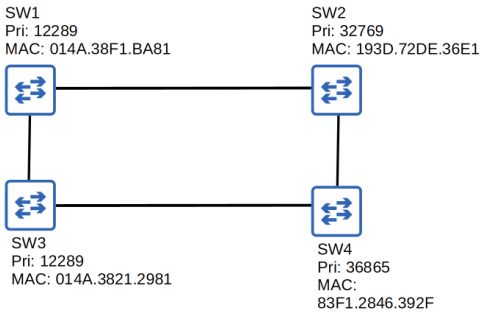
Le Bridge priority contient 4 bits, par défaut le Vlan par défaut est 32769 et il est à 1 pour le premier Bits :

Bridge Priority				Extended System ID (VLAN ID)											
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

La partie de Bridge Priority est contenu en 2 parties le Bridge Priority + Extended System ID ce qui fais que le Bridge Priority ne peut être changé en unités de 4096. La valeurs que l'on peut configurer pour le Bridge Priority sont donc : 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344 ou 61440.

Il est possible aussi de faire en sorte d'assigner un root bridge pour un Vlan déterminé.

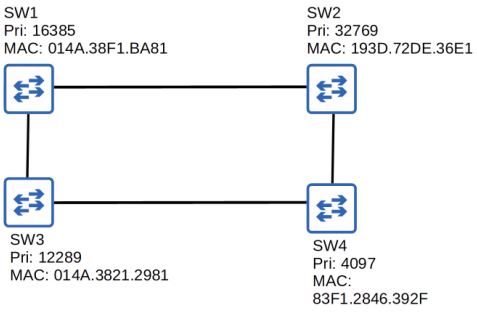
Lorsqu'un switch est allumé il assume devenir le root bridge. Il laisse sa position s'il reçoit un BPDU supérieur (avec un bridge ID plus bas) Une fois que la topologie à convergé et que tous les Switch sont d'accord avec le root bridge, seulement le root bridge envoie des BPDU. Les autres Switch du réseau vont partager ces BPDU, mais ne vont pas générer leurs propres BPDU.



Essayons de répondre à la question suivante à présent, sur la topologie suivante, quelle Switch devient le root Bridge ?

La bonne réponse est le Switch 3, car il possède la plus basse Priorité avec l'adresse MAC la plus basse.

Essayons de déterminer à présent pour cette topologie quelle Switch sera le root Bridge ?



La bonne réponse est le Switch 4 car il possède la plus basse priorité.

Jusqu'à présent nous avons vu la première étape du processus de Spanning tree :

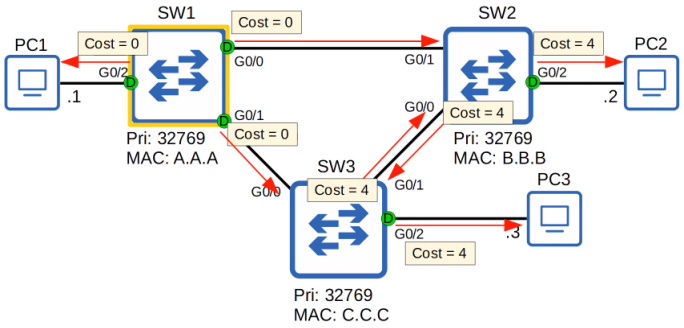
1) Le Switch avec la plus basse Bridge ID est élu comme root Bridge. Tous les ports de ses interfaces sont activés.

2) Chaque autre Switch sélectionne une de ses interface pour être le port root. L'interface avec le coût de root le plus bas sera le root port. Les ports root sont aussi dans un état de partage.

Voici les paramètres pour établir quelle sera le coût pour le Spanning tree :

Vitesse	Coût STP
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2

Si l'on prend l'exemple de cette topologie :

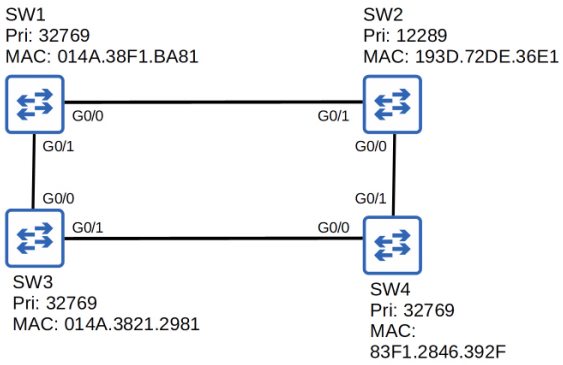


:

l'interface qui a le coût le moins élevé pour le SW2 est l'interface G0/1 avec 0 pour le coût du SW1 et de 4 pour son propre coût ce qui fais un cout total de 4. c'est donc cette interface qui devient le port root.

Il existe une autre possibilité pour qu'une interface soit sélectionnée comme port root, en choisissant l'adresse MAC voisine d'interface le plus bas par exemple lorsque deux interfaces de deux Switch différents ont le même coût, l'interface connectée au Switch avec la plus basse adresse MAC devient le port root.

Prenons un autre exemple pour mieux comprendre :



Sur cette topologie, quelle sera le root bridge ?

La bonne réponse est le SW2, car il a la plus basse priorité. Les deux interfaces sont donc en mode Designated.

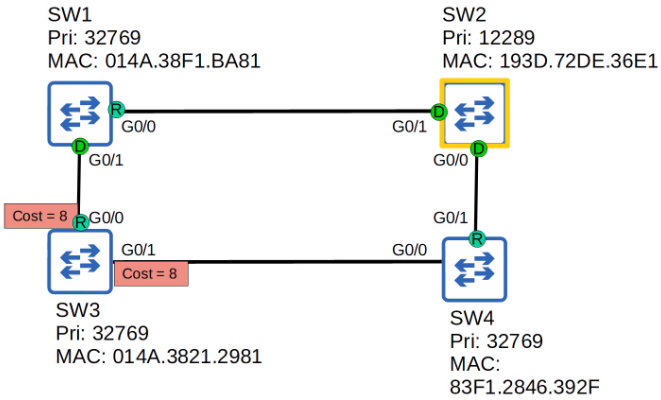
Quelle sera l'interface choisi comme port root entre les deux interfaces Switch 1 et 4 ?

La bonne réponse est l'interface G0/0 du SW1 et G0/1 car ils ont la même valeur.

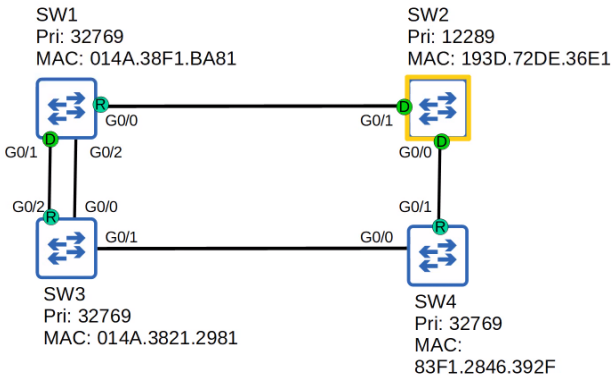
Pour ce qui est du Switch 3 quelle sera alors le port root ?

Puisque sur les deux interface G0/0 et G0/1 les coût interface sont égal cela fais que l'interface connecté à l'adresse MAC la plus basse est sélectionnée donc ici l'interface G0/0.

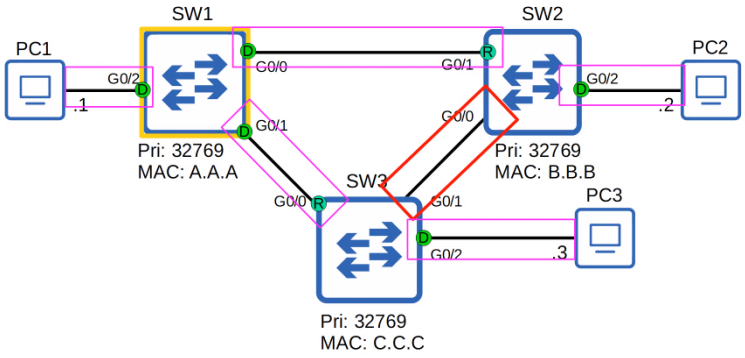
Ce qui fais que la topologie sera semblable à cela :



Il existe encore une possibilité pour sélectionner un port root lorsque deux interface ont la même adresse MAC, par exemple lorsque deux câbles Ethernet sont connectés à deux interfaces d'un seule et même Switch, c'est le port avec le numéro d'interface le plus bas qui est sélectionné par exemple sur cette topologie, c'est l'interface G0/0 qui devient le port root :



Il y reste tout de même la troisième étape à compléter, qui est le blocage de l'une des deux interface non utilisé dans le processus, par exemple dans cette topologie :



La connexion entre SW3 et SW2 n'est pas effective il faut néanmoins déterminer quelle interface sera en mode Designated ou en mode Bloqué.

Pour cela il est déterminé quelle switch avec le coût root est le plus bas ce sera le port désigné.

Si le coût de root est le même c'est le Bridge ID le plus bas qui sera le port Designated, les autres interfaces sont bloqués.

Dans l'exemple c'est l'interface G0/0 qui a le coût de root le plus bas, l'autre interface sera donc bloqué.

Cours 21 : STP (Partie 2)

Dans ce cours nous verrons plus en détail comment fonctionne le protocole Spanning Tree, nous expliquerons se qu'est l'état/Le temps du STP, puis nous verrons plus en détails de quoi est composé le STP BPDU, en dernier nous verrons comment configurer le STP.

Jusqu'à présent nous avons vu surtout deux état sur lesquelles une interface pouvait se mettre :

Le mode Forwarding qui va partager le trafique réseau, et le mode Blocking qui désactive l'interface pour bloquer le trafique.

Il existe néanmoins d'autres mode pour les interfaces dans Spanning Tree qui sont les modes :

Listening et Learning, ces modes sont dits de transition et sont passés lorsqu'une interface est activée ou lorsqu'un port bloqué doit être en transition vers un état de partage à cause d'un changement dans la topologie réseau.

Réexpliquons comment se forment les différents états des ports.

Un port qui est en Non Designated est dans un état « bloqué ». Les interfaces dans un état de blocage sont désactivés pour empêcher les boucles. Les interfaces dans l'état bloqué n'envoient ni ne reçoivent un trafique régulier. Les interfaces dans un état de blocage reçoivent des STP BPDU.

Les interfaces dans l'état bloqué ne partagent pas le trafique STP BPDU et n'apprennent pas non plus les adresses MAC.

Après l'état de blocage, les interfaces avec le Designated ou rôle Root entrent dans l'état de Listening.

Seulement les ports en Designated ou Root entrent dans l'état de Listening.

L'état de Listening est de 15 secondes par défaut. C'est déterminé par le chronomètre de délais de partage. Un interface dans l'état Listening partage/reçoit seulement les STP BPDU. Une interface dans l'état Listening n'envoie/ne reçoit pas de trafique régulier.

Une interface dans l'état Listening n'apprend pas les adresses MAC depuis un trafique régulier qui arrive sur une interface.

Après être dans l'état Listening, un port Designated ou Root entre dans l'état de Learning. L'état de Learning est long de 15 secondes par défaut. Cela est déterminé par le délai de partage. (Le même temps est utilisé pour l'état Listening et Learning). Une interface dans l'état Learning envoie/reçoie seulement des STP BPDU mais n'envoie ni ne reçois de trafique régulier. A la différence d'une interface en mode Listening une interface dans l'état Learning apprend les adresses MAC depuis un trafique régulier qui arrive sur l'interface.

Les port Root et Designated sont dans un état de Forwarding. Un port en Forwarding (partage) fonctionne normalement, il envoie/reçois des BPDU et envoie/reçoit le trafique normalement. Un port dans l'état de Forwarding apprend aussi les adresses MAC.

Voici en résumé dans ce tableau les différents états possibles d'un port STP :

État du port STP	Envoie/reçoit des BPDU	Partage de trames	Apprentissage des adresses MAC	Stable/Transition
Blocking	OUI/NON	NON	NON	Stable
Listening	OUI/OUI	NON	NON	Transitionnel
Learning	OUI/OUI	NON	OUI	Transitionnel
Forwarding	OUI/OUI	OUI	OUI	Stable
Disabled	NON/NON	NON	NON	Stable

Voici un tableau qui résume les temps dans Spanning Tree :

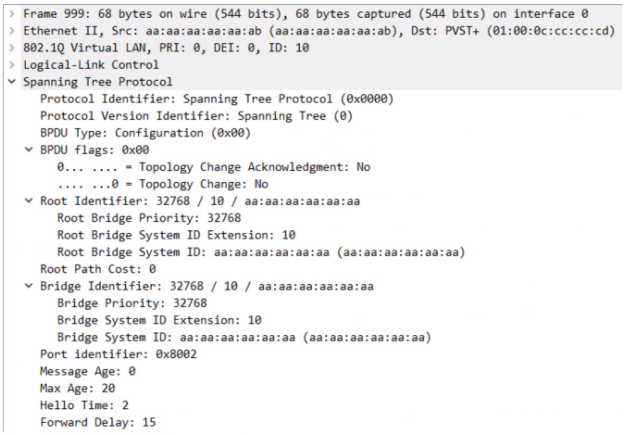
STP Timer	But	Durée
Hello	Le root bridge envoie des hello BPDU	2 secondes
Délais de partage	Temps passé dans l'état Listening/-Learning (chaque état : 15 sec)	15 secondes
Age Maximal	Temps d'attente après réception du Hello BPDU avant de changer la topologie STP	20 secondes (10 × Hello)

Pour la section Age Maximal, le temps est évalué pour savoir si un autre BPDU est reçu avant que le chronomètre du temps maximal descende à 0, le temps est remis à 20 lorsqu'un BPDU arrive et aucun changement ne se passe. Si en revanche aucun BPDU n'est reçu après les 20 secondes, le switch va réévaluer ses choix STP, en incluant le root bridge, et local bridge, designated ou non designated port. Si un port non designated est sélectionné pour devenir designated ou root port, il va transitionner de l'état de bloqué à l'état de Listening (15 secondes), puis en mode Learning (15 secondes) et puis finalement se mettre dans l'état de Forwarding. Donc cela peut prendre un total de 50 secondes pour une interface de passer du mode « bloqué » à « Forwarding ».

Ces temps et états de transitions sont fais pour être certains qu'aucune boucle ne se crée accidentellement par une interface en changeant l'interface en mode forwarding.

Par contre une interface en forwarding peut changer directement en mode bloqué car il n'y a pas de risque de boucles.

Voici plus en détail une capture d'écran d'un BPDU pris sur Wireshark :



Il existe une fonction qui permet de changer immédiatement du mode Forwarding sans passer par les mode Listening et Learning. S'il est utilisé il doit être activé seulement les ports connectés aux end hosts. S'il est activé sur un port connecté à un autre Switch il peut causer des boucles de couches 2.

Voici les commandes pour activer le mode Portfast :

```
SW1(config)#interface g0/2
SW1(config-if)#spanning-tree portfast
```

Cela active le portfast sur toutes les interfaces en mode access et non pas en mode trunk

Il existe tout de même des risques à utiliser Portfast car les interfaces ne passent pas par les modes Learning et Listening il y a donc des risques de boucle si un autre Switch se connecte à l'interface.

Il y a une autre fonction qui permet d'empêcher ces boucles avec la fonction : BPDU guard.

Voici les commandes nécessaire pour activer ce mode :

```
SW1(config)#interface g0/2
SW1(config-if)#spanning-tree bpduguard enable
```

ou pour l'activer par défaut on lance la commande :

```
SW1(config-if)#spanning-tree bpduguard default
```

Il existe d'autres modes utiles pour le spanning tree par exemple :

- Le root Guard : Si l'on active le root guard, même s'il reçoit un BPDU supérieur sur cette interface, le Switch ne va pas accepter le nouveau Switch comme root bridge. L'interface est désactivé.

- Le loop Guard : Si l'on active le loop guard sur une interface, même si l'interface arrête de recevoir des BPDU, il ne va pas arrêter de faire du forwarding (partage). L'interface est désactivé.

Voici à présent les commandes pour configurer le mode Spanning tree :

```
SW1(config)#spanning-tree mode pvst
```

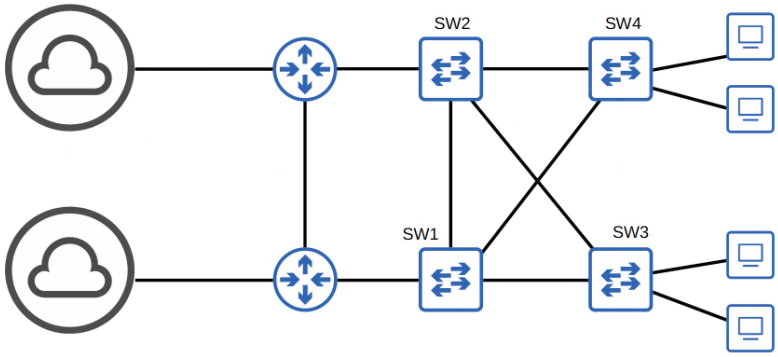
Voici les commandes pour changer un Switch afin qu'il devienne le root Bridge :

```
SW3(config)#spanning-tree vlan 1 root primary
```

Cette commande place la priorité STP à 24576. Si un autre Switch connecté à aussi une priorité de 24576 il va placer la priorité à 4096 de moins que l'autre priorité avec cette commande :

```
SW3(config)#spanning-tree vlan 1 root secondary
```

Voici un exemple pour mieux comprendre comment configurer la priorité sur cette topologie :



2 vlan sont actif sur ce réseau, 10 et 20. Par défaut le SW3 est le root bridge pour les deux vlan. Pour configurer le SW2 comme root primary pour le Vlan 20 et le second root pour le Vlan 10 quelles sont les deux commandes à utiliser ?

Voici les deux commandes à utiliser sur les deux Switch :

```
SW1(config)#spanning-tree vlan 10 root primary
SW1(config)#spanning-tree vlan 20 root secondary
SW2(config)#spanning-tree vlan 20 root primary
SW2(config)#spanning-tree vlan 10 root secondary
```

Cours 22 : Rapid STP

Dans ce cours nous verrons comment fonctionne le Rapid Spanning Tree. Nous verrons donc une comparaison des différentes versions de Spanning-Tree (Standard vs Cisco), Puis nous verrons comment fonctionne le Rapid PVST+

Commençons d'abord par comparer les deux versions de Spanning-Tree celle développée par l'IEEE et celle développée par Cisco :

Industry Standard (IEEE)

Spanning-Tree Protocol (802.1D) :

C'est le protocole STP originale, Toutes les Vlans partagent une seule instance STP. Il ne peut cependant pas être load balance.

Rapid Spanning-Tree Protocol (802.1w)

Bien plus rapide pour s'adapter aux changements du réseau que le 802.1D. Toutes les Vlans partagent une seule et même instance STP, par contre qui ne peuvent pas être load Balance.

Multiple Spanning Tree Protocol (802.1s)

Utilise des RSTP mécaniques, peut regrouper plusieurs Vlans dans différentes instances (par exemple Vlans 1-5 dans l'instance 1, Vlan 6-10 dans l'instance 2) pour faire fonctionner le load balancing.

Versions Cisco

Per-Vlan Spanning Tree Plus (PVST+) :

Mis à jour Cisco vers 802.1D, chaque Vlan à sa propre instance STP, peut load balance en bloquant différents ports dans chaque Vlan.

Rapid Per-Vlan Spanning Tree Plus (Rapid PVST+) :

Mis à jour Cisco vers 802.1w, chaque Vlan à sa propre instance STP, et peut load balance en bloquant différents ports dans chaque Vlan.

Dans ce cours nous verrons plus en détails les versions : Rapid Spanning-Tree Protocol (802.1w) et Rapid Per-Vlan Spanning Tree Plus (Rapid PVST+).

Résumé de RSTP donné par Cisco : « RSTP n'est pas un algorithme basé sur le temps comme 802.1D bien que RSTP offre une amélioration de 30 secondes ou plus par rapport à 802.1D pour changer au mode Forwarding. Le coeur de ce protocole est le nouveau mécanisme de pont/pont et handshake, qui permet aux port de changer directement en mode forwarding. »

Voici les ressemblances possibles entre STP et RSTP :

- RSTP sert le même but que STP, bloquer des ports spécifiques pour empêcher des boucles de couche 2.

- RSTP élit le root bridge avec les même règles que STP

- RSTP élit les ports designated avec les même règle que STP

Il y a par contre une mis à jour des coûts pour le RSTP par rapport aux coûts STP :

Rapidité	Coûts STP	Coût RSTP
10 Mbps	100	2 000 000
100 Mbps	19	200 000
1 Gbps	4	20000
10Gbps	2	2000
100Gbps	X	200
1Tbps	X	20

A la différence du protocole STP dans le protocole RSTP les modes : Blocking, Listening et Disabled qui sont combinés en un seul mode appelé le mode Discarding.

Voici donc un tableau avec l'état de statut des Ports pour le mode RSTP :

État des ports	Envoi/Réception BPDU	Partage de trame (trafic régulier)	Apprentissage des adresses MAC	Stable / Transitionnel
Discarding	NON/OUI	NON	NON	Stable
Learning	OUI/OUI	NON	OUI	Transitionnel
Forwarding	OUI/OUI	OUI	OUI	Stable

Si un port est désactivé il a le statut de « discarding » dans RSTP

Si un port est activé mais bloque le trafique pour empêcher les boucles de couche 2 il est aussi en mode « discarding ».

Les rôles de ports ne changent pas dans RSTP.

Le port le plus proche du root bridge devient le port root pour le switch. Le root bridge est le seule switch qui n'a pas de port root.

Le designated port est inchangé dans RSTP. Le port dans un segment (collision domain) qui envoie le meilleur BPDU est le segment designated port.

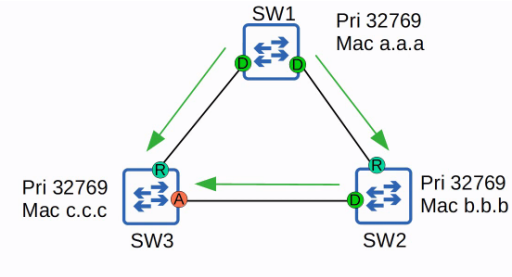
Le non-designated port est divisé en deux rôles séparés dans RSTP :

Le rôle de port alternatif et le rôle de port de backup.

Le rôle de port : « alternate » est un port qui reçoit un BPDU supérieur d'un autre switch.

C'est la même chose que ce que l'on a appris à propos des ports bloqués dans le STP classique.

Il fonctionne comme un Backup pour le port root. Si le root port ne marche plus, le switch peut immédiatement changer son meilleur port alternate vers le forwarding.



Sur cette topologie par exemple si le port root du SW3 ne fonctionne plus, le port en Alternate devient automatiquement le port root.

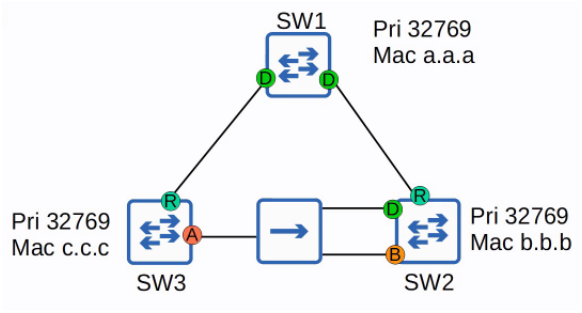
Un fonctionnalité optionnel ajouté à STP et intégré à RSTP est le BackboneFast.

Cette fonction permet qu'en cas de dysfonctionnement d'une interface le temps est rapidement transféré au BPDU supérieur sans délais supplémentaire. Cette fonctionnalité est intégré à RSTP il n'est donc pas nécessaire de le configurer.

UplinkFast et BackboneFast sont deux fonctionnalités optionnels dans le STP classique. Ils doivent être configurés pour fonctionner dans le Switch. Les deux fonctionnalités dont intégrés à RSTP, donc il n'est pas nécessaire de les configurer. Ils fonctionnent par défaut.

Le rôle de port de Backup dans RSTP est un port discarding qui reçoit un BPDU supérieur d'une autre interface sur le même switch. Cela se passe uniquement lorsque deux interface sont connectées au même domaine de collision (par un hub). Les Hubs ne sont plus trop utilisé dans un réseau moderne donc il est peu probable d'avoir un port de Backup dans RSTP.

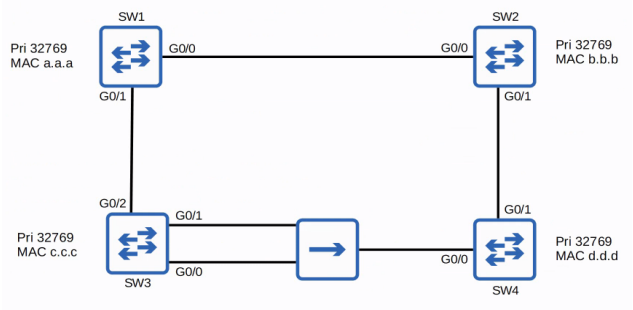
Il fonctionne comme un Backup pour le designated port comme sur ce schéma :



Si le port Designated du SW2 n'est plus fonctionnel c'est le port de Backup qui prendra le relais.

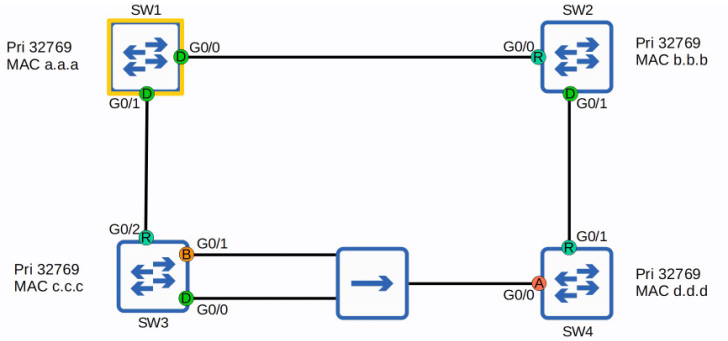
C'est l'interface avec le port ID le plus bas qui est sélectionné comme designated port l'autre est le backup.

Essayons de voir si tout a été compris, pour cela il faut trouver le root bridge et les rôles de chaque port sur cette topologie :



Le root Bridge est le SW1 puisqu'il a l'adresse MAC la plus basse de toutes, ses interfaces sont donc en mode Designated. Les ports root deviennent les interfaces g0/0 pour le SW2, G0/2 pour le SW3 et G0/1 pour le SW4 car son interface voisine connecté au SW2 a la plus basse adresse MAC comparé à l'adresse MAC du SW3 connecté. L'interface G0/0 est choisi pour être Designated par rapport à G0/1 qui est en Backup car c'est l'interface avec le numéro de port le plus bas.

L'interface G0/0 du SW4 est en Alternate. Voici en résumé le schéma avec les rôles des ports :



Voici les commandes pour configurer le mode PVST sur un Switch :

```
SW3(config)#spanning-tree mode rapid-pvst
```

On peut vérifier la configuration avec :

```
SW3(config)#show spanning-tree
```

Rapid STP est compatible avec le STP classique. L'interface connecté dans le Rapid STP active le Switch connecté au STP classique et marchera en mode STP classique.

Voici un BPDU utilisé par RSTP capturé par Wireshark :

```
> Frame 71: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> IEEE 802.3 Ethernet
> Logical-Link Control
v Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Rapid Spanning Tree (2)
    BPDU Type: Rapid/Multiple Spanning Tree (0x02)
  v BPDU flags: 0x3c, Forwarding, Learning, Port Role: Designated
      0... .... = Topology Change Acknowledgment: No
      .0.. .... = Agreement: No
      ..1. .... = Forwarding: Yes
      ...1 .... = Learning: Yes
      .... 11.. = Port Role: Designated (3)
      .... ..0. = Proposal: No
      .... ...0 = Topology Change: No
  v Root Identifier: 32768 / 1 / aa:aa:aa:aa:aa:aa
      Root Bridge Priority: 32768
      Root Bridge System ID Extension: 1
      Root Bridge System ID: aa:aa:aa:aa:aa:aa (aa:aa:aa:aa:aa:aa)
    Root Path Cost: 4
  v Bridge Identifier: 32768 / 1 / cc:cc:cc:cc:cc:cc
      Bridge Priority: 32768
      Bridge System ID Extension: 1
      Bridge System ID: SiliconL_cc:cc:cc (cc:cc:cc:cc:cc:cc)
    Port identifier: 0x8001
    Message Age: 1
    Max Age: 20
    Hello Time: 2
    Forward Delay: 15
    Version 1 Length: 0
```

Voyons quelques autres différences avec le RSTP :

Tous les switch qui lancent RSTP envoient leurs propre BPDU à chaque temps de hello (2 secondes)

Les Switch datent les informations de BPDU plus rapidement. Dans un STP classique un Switch attend 10 interval de hello (20 secondes). Dans le STP rapide, un switch considère que son voisin est perdu s'il perd 3 BPDU (6 secondes). Il va ensuite perdre toutes les adresses MAC apprises sur cette interface.

Dans RSTP il est distingué 3 différents type de lien :

- Edge : un port connecté à un End Host. Change directement en forwarding sans vérification. Les ports Edge sont connectés au end hosts. Puisqu'il n'y a pas de risque de créer de boucle, ils peuvent changer d'état sans vérifiacation. Ils fonctionnent comme un STP classique avec le PortFast activé.

Comme avec cette commande :

```
SW1(config-if)\#spanning-tree portfast
```

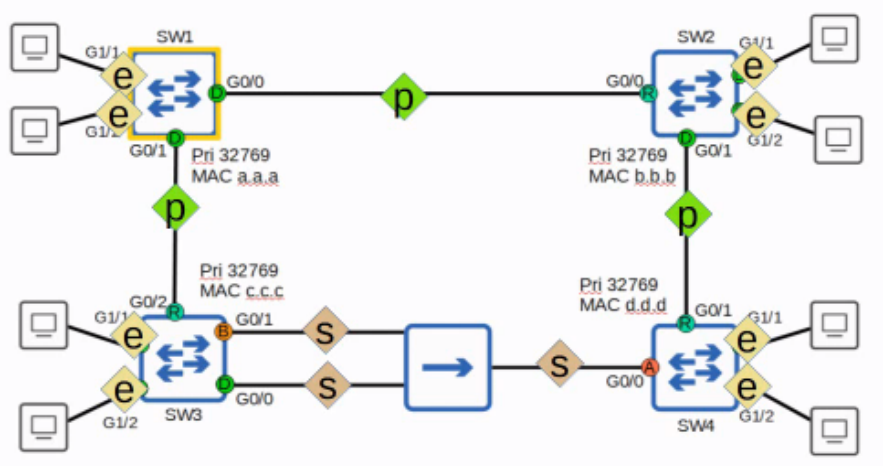
- Point to Point : une connexion direct entre deux Switch. Ils fonctionnent en Full Duplex. Il n'est pas nécessaire de configurer l'interface en point to point car détecté automatiquement. Cependant il est tout de même possible de le faire avec la commande suivante :

```
SW1(config-if)#spanning-tree link-type point-to-point
```

- Shared : Une connexion à un Hub. Ils fonctionnent en mode half duplex. Il n'est pas nécessaire de configurer l'interface en shared car détecté automatiquement. Cependant il est tout de même possible de le faire avec la commande suivante :

```
SW1(config-if)#spanning-tree link-type shared
```

Voici en résumé sur ce schéma les ports qui sont Edge, Shared et point to point :

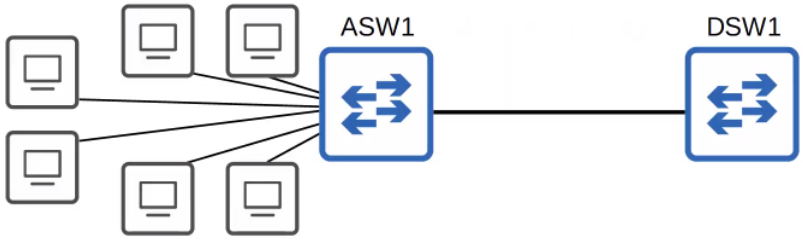


Cours 23 : Etherchannel

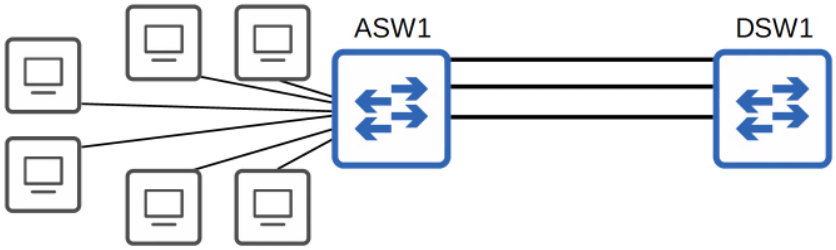
Dans ce cours nous allons apprendre comment fonctionne le protocole Etherchannel et quelle problème il permet de résoudre.

Nous Verrons comment configurer l'Etherchannel sur les couche 2/3 (Switch et routeur).

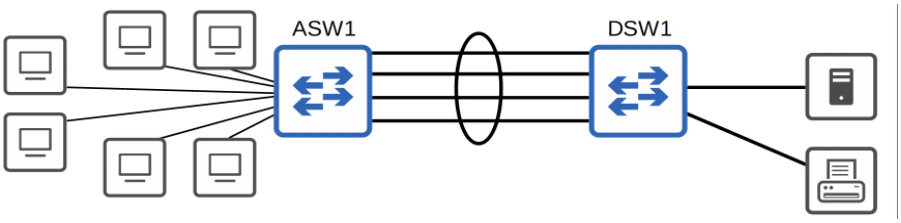
Pour comprendre voyons ce schéma :



On peut voir que sur le Switch ASW1 est connecté beaucoup de PC ce qui fais que la bande passante du réseau est limité entre le ASW1 et DSW1. Pour augmenter le trafique réseau le technicien décide d'ajouter un deuxième voir un troisième câble entre les deux Switch comme ceci :



Pourtant la connexion n'est toujours pas plus efficace qu'avant ceci est dû au fais que le Spanning tree ne va en faite utiliser qu'une seule interface pour éviter les tempêtes de Broadcast, pour que toutes les interfaces soient utilisé il faut utiliser et configurer le protocole Etherchannel entre les deux Switch :



Lorsque la bande passante de l'interface connecté aux PC est plus élevé que la bande passante qui relie les Switch c'est appelé en anglais : oversubscription

Certaine oversubscription sont acceptable, mais s'il y en a trop cela peut causer une congestion.

Le protocole permet donc à un groupe de plusieurs interfaces de fonctionner comme une seule interface. STP va traiter le groupe d'interface comme si qu'il s'agissait d'une seule interface.

Le protocole Etherchannel peut avoir d'autre noms comme :

« Port Channel » ou « LAG (Ling Aggregation Group) »

Etherchannel load balance le trafique basé sur le flux. Un flux est une communication entre deux nœud dans un réseau. Les trames du même flux sont envoyés sur la même interface. Si une trame du même flux est envoyé sur une interface différentes certaines trames pourraient arriver en dehors du temps se qui pourrait causer des problèmes.

Il est possible de changer les paramètres d'interface utilisé pour la sélection de l'interface.

Les paramètres qui peuvent être changés pour la sélection sont : l'adresse MAC source, l'adresse MAC de destination ou les deux, l'adresse IP source, l'adresse IP de destination ou les deux.

Voici les commandes pour vérifier l'etherchannel en load ballancing :

```
ASW1#show etherchannel load-balance
```

Pour changer la configuration de la méthode d'etherchannel à sélectionner (dans ce cas utiliser les adresse MAC source et destination pour méthode) il faut utiliser ces commandes :

```
ASW1#conf t
ASW1(config)#port-channel load-balance src-dst-mac
```

Voyons à présent comment configurer l'Etherchannel entre deux Switch, il existe 3 méthodes :

- PagP (Port Aggregation Protocol) : C'est un protocole propriétaire Cisco qui négocie dynamiquement la création/maintenance de l'Etherchannel (Comme DTP fais pour les trunks)

- LACP (Link Aggregation Control Protocol) : C'est un protocole Industry Standard (IEEE 802.3ad) qui négocie dynamiquement la création/maintenance de l'etherchannel (comme DTP fais pour les trunk)

- Static Etherchannel : C'est un protocole qui n'est pas utilisé pour déterminer si un Etherchannel doit être formé. Les interfaces sont configuré de manière statique pour former un Etherchannel.

Jusqu'à 8 interfaces peuvent être formés dans un seule Etherchannel (LACP permet jusqu'à 16, mais seulement 8 seront active, les autres 8 seront en mode standby, et attendent qu'une interface ne fonctionne plus)

Voici les commandes à utiliser pour configurer l'Etherchannel en mode PAgP :

```
ASW1(config)#interface range g0/0 -- 3
ASW1(config-if-range)#channel-group 1 mode desirable
```

Le « channel group » doit être le même numéro pour les membres de l'interface sur le même switch. Même s'il n'y a pas besoin que ce soit le même numéro pour un autre switch.

Il faut tout de même configurer l'etherchannel sur les interfaces des deux switch voulus pour l'etherchannel.

Voici les commandes pour configurer l'Etherchannel en mode LACP :

```
ASW1(config)#interface range g0/0 -- 3
ASW1(config-if-range)#channel-group 1 mode active
```

Pour configurer l'Etherchannel en mode Statique on lance les commandes :

```
ASW1(config)#interface range g0/0 -- 3
ASW1(config-if-range)#channel-group 1 mode on
```

Il est aussi possible de bloquer l'utilisation de l'etherchannel en utilisant qu'un seul protocole voulus (ici LACP) avec la commande :

```
ASW1(config-if-range)#channel-protocol lacp
```

Pour configurer ensuite l'interface Etherchannel et lancer le trunk on peut lancer directement la commandes :

```
ASW1(config)#interface port-channel 1
ASW1(config-if)#switchport trunk encapsulation dot1q
ASW1(config-if)#switchport mode trunk
```

Pour que l'Etherchannel fonctionne correctement il est nécessaire que la configuration soit la suivante :

Les membres des interfaces doivent avoir la même configuration, cela inclut : le même duplex (full/half), la même vitesse, Le même mode de switchport (access/trunk), la même Vlan native autorisé (pour l'interface trunk). Si la configuration d'une interface n'est pas la même avec les autres, l'interface sera exclut de l'etherchannel.

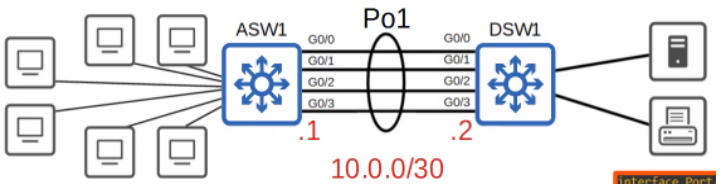
Pour vérifier le statut d'un etherchannel on peut lancer la commande :

```
ASW1#show etherchannel summary
```

une autre commande pour vérifier le statut Etherchannel :

```
ASW1#show etherchannel port-channel
```

Il est aussi possible d'utiliser l'Etherchannel avec des Switch de couche 3 qui font aussi office de routeur. Comme sur cette topologie :



Voyons à présent comment configurer une interface de couche 3 directement en ligne de commande :

```
ASW1(config)#int range g0/0 -- 3
ASW1(config-if-range)#no switchport
ASW1(config-if-range)#channel-group 1 mode active
```

La différence avec un switch de couche 2 est qu'il n'y a pas de risque de tempête de Broadcast et donc pas nécessaire de configurer le Spanning tree.

Par contre pour configurer ensuite l'interface et ajouter l'adresse IP il faut utiliser le port channel interface sur un switch de couche 3 :

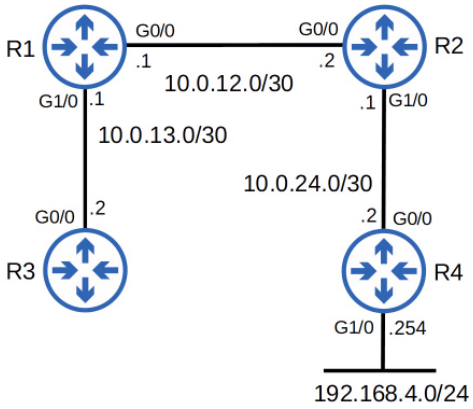
```
ASW1(config-if-range)#int po1
ASW1(config-if-range)#ip address 10.0.0.1 255.255.255.252
```

Cours 24 : Routage Dynamique

Dans ce cours cours nous verrons comment fonctionne le routage dynamique sur un routeur.

Nous commencerons par faire une introduction sur le protocole de routage dynamique, puis nous verrons quelles sont les différents types de routage de protocoles dynamique. Ensuite nous ferons le routage dynamique du protocole des métriques. Nous verrons en dernier : l'administrative distance (AD).

Nous allons commencer par utiliser la topologie suivante pour mieux comprendre :



La table de routage du routeur R1 est la suivante :

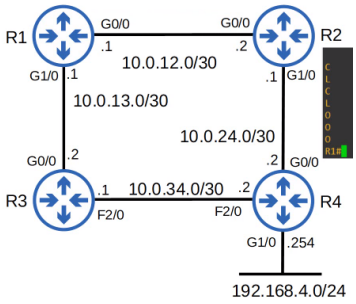
```
     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.0.12.0/30 is directly connected, GigabitEthernet0/0
L       10.0.12.1/32 is directly connected, GigabitEthernet0/0
C       10.0.13.0/30 is directly connected, GigabitEthernet1/0
L       10.0.13.1/32 is directly connected, GigabitEthernet1/0
R1#
```

On peut voir que les adresse 10.0.12.0/30 et 10.0.13.0/30 sont des adresses de routage du réseau car elles sont ajoutés avec des masques de /30.

Les adresses 10.0.12.1/32 et 10.0.13.1/32 sont des adresses de routage d'hôtes car elles sont ajoutés avec des masques en /32.

Au lieu de configurer les routeurs avec du routage statique il est possible de les configurer de manière dynamique, de manière à ce que les tables de routage soient communiqués entre les Routeurs directement. Un avantage dans la configuration dynamique est que si un routeur a une interface qui ne fonctionne plus en dynamique il est automatiquement retiré de la table de routage, tandis que ça n'est pas le cas en statique.

C'est pour cela qu'il faut ajouter une connexion supplémentaire de Backup comme ici :



Sur ce schéma dans le cas ou l'interface d'un des routeur ne fonctionne plus il y aura toujours une interface de Backup pour que le réseau fonctionne normalement si les routeur sont configurés en dynamique.

Les routeurs peuvent donc utiliser le routage dynamique pour avertir des informations à propos du routage qu'ils connaissent aux autres routeurs.

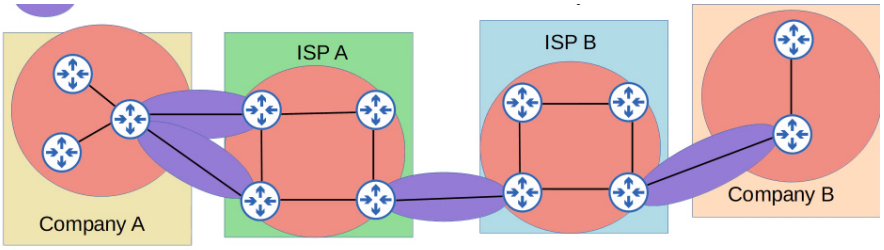
Il formes des « relation de voisin » « des voisins » avec les routeurs adjacents pour échanger les informations. Si plusieurs routes vers une destination est apprise, le routeur détermine quelle route est supérieur et l'ajoute à sa table de routage. Il utilise la « métrique » du routage pour décider quelle est la priorité (la plus basse métrique à la priorité comme dans STP)

Il existe différents types de protocoles de routage dynamique qui peuvent être divisés en deux catégories :

- IGP (Interior Gateway Protocol) : Il est utilise pour partager la route avec un seule système autonome « autonomous system » (AS) qui est une seule organisation

- EGP (Exterior Gateway Protocol) utilisé pour partager les routes entre différents systèmes autonomes.

Voici sur ce Schéma pour mieux comprendre la différence entre EGP et IGP :



IGP est constitué des LAN local en cercle beige, et EGP est constitué des interfaces qui font la connexion entre ces LAN en cercle violet.

Pour EGP le type d'algorithme qui est utilisé est le vecteur de chemin avec le protocole BGP (Border Gateway Protocol).

Par contre IGP utilise des types d'algorithme différents qui sont le vecteur distance et le l'état des lien.

Les protocole majeurs utilisés pour le vecteur de distance sont Routing Information Protocol (RIP) et Enhanced Interior Gateway Routing Protocol (EIGRP).

Les protocoles qui utilisent le type d'algorithme d'état des lien est Open Shortest Path First (OSPF) et Intermediate System to Intermediate System (IS-IS)

Voici quelques caractéristiques des protocoles de vecteur distant :

Les protocoles de vecteur distant ont été inventés avant les protocoles d'état des liens.

Un exemple est RIPv1 et le protocole Cisco propriétaire IGRP (renommé en EIGRP)

Les protocoles avec le vecteur distant fonctionnent en envoyant : les destinations réseau connus et leurs métrique pour joindre les destination à leurs voisins directement connectés.

Cette méthode de partage d'information est souvent appelé « routage par rumeur »

C'est par ce que le routage ne connais pas le réseau derrière leurs voisins. Il connais uniquement l'information que les voisins lui donnent.

C'est appelé aussi « vecteur distance » car les routeurs apprennent seulement la « distance » (métrique) et le « vecteur » (direction, le prochain bond du routeur) pour chaque routage.

Voici à présent quelques caractéristiques des protocoles d'état des liens :

Lorsqu'un protocole état des lien est utilisé, chaque routeur crée une carte de connectivité du réseau. Pour permettre cela chaque routeur averti l'information à propos de son interface à ses voisins. Ces avertissements sont passés à d'autres routeurs jusqu'à se que tous les routeurs développent la même carte du réseau. Chaque routeur utilise indépendamment cette carte pour calculer la meilleure route pour chaque destination.

L'état des lien utilise plus de ressources (CPU) sur le routeur car plus d'informations sont partagés. Cependant les protocoles état des liens on tendance à être plus rapide en réaction aux changements dans le réseau plutôt qu'aux protocole de vecteur distance.

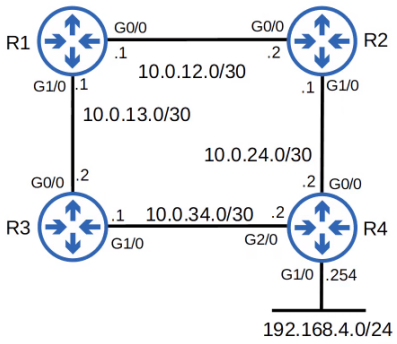
La table de routage d'un routeur contient la meilleur route qu'il connaisse pour chaque destination.

Si un routeur qui utilise un protocole de routage dynamique apprend deux routes différentes pour la même destination, comment déterminera t-il la meilleure route ?

Il utilisera la valeur de métrique pour déterminer la meilleur route, plus la valeur métrique est basse et plus cette route est meilleure. Chaque protocole de routage utilise une métrique différente pour déterminer quelle route est la meilleure.

Mais que ce passe t-il si la valeur de la métrique est la même ?

Pour répondre voyons plus en détail ce schéma :



On lance la commande show ip route sur le routeur 1 pour mieux identifier la table de routage :

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
O        10.0.24.0/30 [110/2] via 10.0.12.2, 00:00:09, GigabitEthernet0/0
O        10.0.34.0/30 [110/2] via 10.0.13.2, 00:00:09, GigabitEthernet1/0
O     192.168.4.0/24 [110/3] via 10.0.13.2, 00:00:09, GigabitEthernet1/0
                     [110/3] via 10.0.12.2, 00:00:09, GigabitEthernet0/0
R1#
```

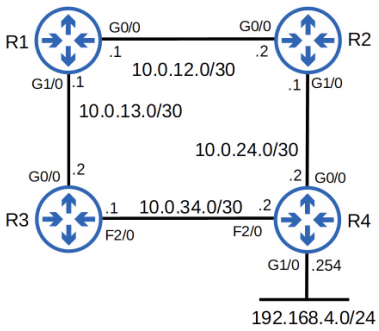
On peut voir que les deux interfaces avec les même métriques (ici /3) ont été ajoutés et le trafique va être load balance entre les deux routes ceci est appelé ECMP (Equal Cost Multi-Path). Ici le protocole utilisé est OSPF.

Si la configuration avait été faite en statique le résultat de la commandes aurait été différent :

```
R1(config)#ip route 192.168.4.0 255.255.255.0 10.0.12.2
R1(config)#ip route 192.168.4.0 255.255.255.0 10.0.13.2
R1(config)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
S     192.168.4.0/24 [1/0] via 10.0.13.2
                     [1/0] via 10.0.12.2
R1(config)#
```

IGP	Métrique	Explication
RIP	Nombre de sauts	Chaque saut vers un nouveau routeur équivaut à 1. La métrique totale est le nombre de sauts vers la destination. Tous les liens ont le même coût.
EIGRP	Bande passante et délai (par défaut)	Utilise une formule complexe prenant en compte plusieurs valeurs. Par défaut, la bande passante du lien le plus bas et le délai total de tous les liens sont utilisés.
OSPF	Coût	Calculé en fonction de la bande passante. La métrique totale est la somme des coûts des liens dans la route.
IS-IS	Coût	La métrique totale est la somme des coûts des liens. Le coût de chaque lien n'est pas automatiquement calculé ; par défaut, tous les liens ont un coût de 10.

Pour mieux démontrer les différences entre les protocoles RIP et EIGRP utilisons le schéma suivant :



Pour le protocoçle RIP ce sont tous les réseaux qui vont être ajoutés à la table de routage pour identifier la meilleur route, tandis que pour le protocole OSPF c'est uniquement la route du R1 puis R2 et R4 qui est retenu comme la meilleure route.

Dans la plupart des cas une entreprise utilise un seule IGP, soit OSPF ou EIGRP.

Cependant il se peut que l'entreprise utilise les deux. Par exemple si deux entreprises connectent leurs réseaux pour partager des informations, deux protocoles de routage différents devraient être utilisés. Les métriques sont utilisés pour comparer les routes connu par le même protocole de routage. Différents protocoles de routage utilisent des métriques totalement différents donc ils ne peuvent pas être comparés. Par exemple un routage OSPF vers 192.168.4.0/24 devrait avoir une métrique de 30, quant à une route vers la même destination avec EIGRP devrait avoir une métrique de 33280. Quelle est la route que le routeur va utiliser dans sa table de routage ?

Le administrative distance (AD) est utilisé pour déterminer quelle protocole est préféré.

Le plus bas AD est préféré et indique que le protocole de routage est considéré comme plus sûre.

Voici un tableau qui résume les administrative distance pour chaque type de protocole :

Type de protocole	AD
Directement connecté	0
Statique	1
BGP externe (eBGP)	20
EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EIGRP (externe)	170
BGP interne (iBGP)	200
Route inutilisable	255

Voici une question pour voir si c'est bien clair :

Les routes suivantes vers la destination réseau 10.1.1.0/24 sont connu :

→ bond 192.168.1.1 appris via RIP, métrique 5

→ bond 192.168.2.1 appris via RIP, métrique 3

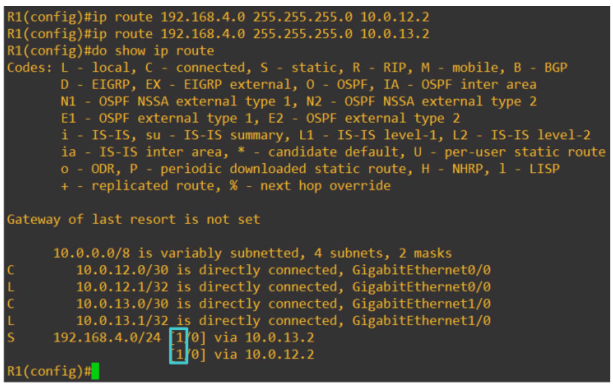
→ bond 192.168.3.1 appris via OSPF, métrique 10

Quelle sera la route que 10.1.1.0/24 va ajouter à sa table de routage ?

La métrique est utilisé pour comparer les routes apprises depuis le même protocole de routage.

Cependant avant de comparer les métriques AD est utiliser pour sélectionner la meilleure route.

Le route OSPF va toujours prendre une avance sur le routage RIP, car il a le plus bas AD.



Sur cette table de routage ont peut voir une valeur AD de « 1 », l'AD est placé à gauche de la valeur de métrique.

Il est possible de changer l'AD d'un protocole de routage.

Il est aussi possible de changer l'AD sur un routage statique avec la commande suivante :

```
R1(config)#ip route 10.0.0.0 255.0.0.0 10.0.13.2 100
```

Ici l'AD utilisé est 100.

En changeant l'AD d'un routage statique on peut le rendre moins préféré qu'une route appris en dynamic routing protocol avec la même destination. Ceci est appelé le « floating static route »

La route devient inactive jusqu'à ce que la route apprise par la manière dynamique est supprimé.

Cours 25 : RIP & EIGRP

Dans ce cours nous verrons comment fonctionne les protocoles : Routing Information Protocol (RIP) et Enhanced Interior Gateway Routing Protocol (EIGRP).

Routing Information Protocol (industry Standard) est un vecteur de distance IGP (utilise la logique du routage par rumeur pour apprendre/partager les routes), il utilise un compte des bonds comme métrique. Un routeur est égal à un saut. Le compte de sauts maximal est de 15 (s'il y a plus de sauts que cela la destination n'est plus joignable). Ce protocole possède 3 versions :

- RIPv1 et RIPv2 utilisé pour IPv4

- RIPng (Nouvelle génération de RIP), utilisé pour IPv6

Il utilise 2 types de messages :

Des requêtes : Pour demander d'activer le RIP au routeurs voisins pour envoyer leurs tables de routage.

Des réponses : Pour envoyer la table de routage aux routeurs voisins.

Par défaut lorsque le RIP est activé, les routeurs vont partager leurs tables de routage toutes les 30 secondes.

Comparons à présent RIPv1 et RIPv2 :

- RIPv1 : N'avertit que les adresses classés (Classe A, Classe B, Classe C), il ne supporte pas VLSM et CIDR, il n'inclut pas les informations de masque de sous réseau dans ses avertissements (messages de réponses) par exemple :

10.1.1.0/24 deviendra 10.0.0.0 (Une adresse de classe A, et être en /8)

172.16.192.0/18 deviendra 172.16.0.0 (une adresse de Classe B, et être en /16)

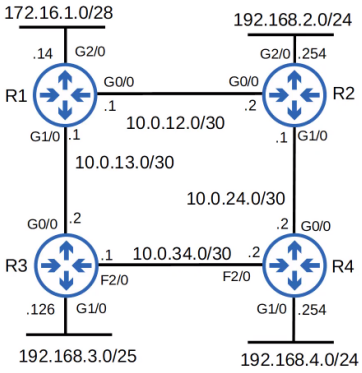
192.168.1.4/30 deviendra 192.168.1.0 (une adresse de Classe C, et être en /24)

Les messages sont envoyés en Broadcast à l'adresse 255.255.255.255

- RIPv2 : Supporte le VLSM et le CIDR, il inclut les informations dans son sous masque réseau dans les avertissements. Les messages sont envoyés en Multicast à l'adresse 224.0.0.9

Pour rappel les messages en Broadcast sont envoyés à tous les appareils du réseau local. En multicast les messages sont envoyés uniquement aux appareils qui ont joins le groupe multicast spécifique.

Voyons à présent comment faire de la configuration basique avec RIP prenons pour exemple cette topologie :



Voici donc les commandes nécessaires pour la configuration du protocole RIP :

```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto-summary
R1(config-router)#network 10.0.0.0
R1(config-router)#network 172.16.0.0
```

Les réseaux das RIP sont uniquement classés c'est à dire que c'est automatiquement convertis en adresse d'un réseau classé.

Par exemple même si l'on entre les commandes :

network 10.0.12.0 cela convertira la commande en network 10.0.0.0 (un réseau de classe A)

Il n'y a donc pas besoin d'entrer de masque de sous réseau.

La commande «*network*» dis au routeur de :

- voir l'interface avec l'adresse IP spécifié

- activer le RIP sur les interfaces spécifiés

- Former des connexions adjacentes avec les voisins RIP

- avertis le préfixe réseau de l'interface (Pas le préfixe dans la commande network)

La commande « network » fonctionne de la même manière pour OSPF et EIGRP.

Par exemple avec la commande :

```
R1(config-router)#network 10.0.0.0
```

Puisque la commande « network » est classé 10.0.0.0 sera en 10.0.0.0/8

R1 va regarder si les interfaces avec une adresse IP correspondent à 10.0.0.0/8 (puisque en /8 le réseau doit correspondre seulement au 8 premiers Bits)

10.0.12.1 et 10.0.13.1 correspondent, donc RIP sera activé sur G0/0 et G1/0 sur le schéma.

R1 forme des connexions adjacentes avec ses voisins R2 et R3.

R1 avertis 10.0.12.0/30 et 10.0.13.0/30 (et non pas 10.0.0.0/8) à ses voisins RIP.

La commande « network » ne dis pas au routeur quelle réseau avertir. Il dis au routeur quelle interfaces doit activer RIP, ensuite le routeur va avertir le préfixe réseau à ces interfaces.

Avec la commande :

```
R1(config-router)#network 172.16.0.0
```

Puisque la commande « network » est classé 172.16.0.0 sera en 172.16.0.0/16

R1 regardera n'importe quelle interface avec une adresse IP qui correspond à 172.16.0.0/16

172.16.1.14 correspond donc R1 va activer RIP sur G2/0

Il n'y a pas de voisins RIP connectés à G2/0 donc de nouvelles connexions adjacentes sont formés.

R1 avertit 172.16.1.0/28 (non pas 172.16.0.0/16) à ses voisins RIP.

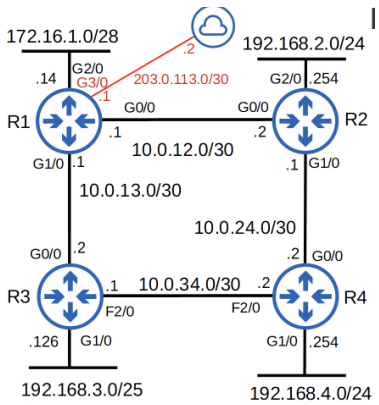
Puisqu'il n'y a pas de voisins RIP connectés à G2/0, R1 va continuellement envoyer des avertissements vers G2/0. Comme c'est du trafique non nécessaire G2/0 devrait être configuré comme interface passive.

Pour configurer l'interface en passif on utilise la commande :

```
R1(config-router)#passive-interface g2/0
```

La commande passive-interface dis au routeur d'arrêter d'envoyer des avertissements RIP vers l'interface spécifié (G2/0). Cependant le routeur continuera d'avertir le préfixe réseau de son interface (172.16.1.0/28) à ses voisins RIP (R2, R3). On devrait toujours utiliser cette commande dans l'interface qui n'a pas de voisins RIP. EIGRP et OSPF ont les deux la même fonctionnalité d'interface passive, en utilisant la même commande.

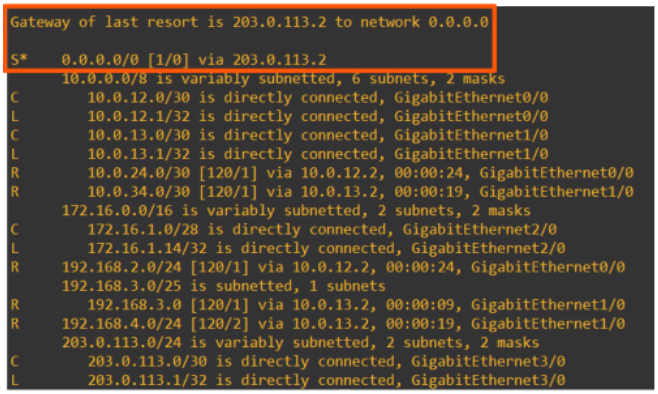
Pour la démonstration il a été ajouté une connexion internet sur une interface R1 :



La commande suivante a ensuite été configuré :

```
R1(config)#ip route 0.0.0.0 0.0.0.0 203.0.113.2
```

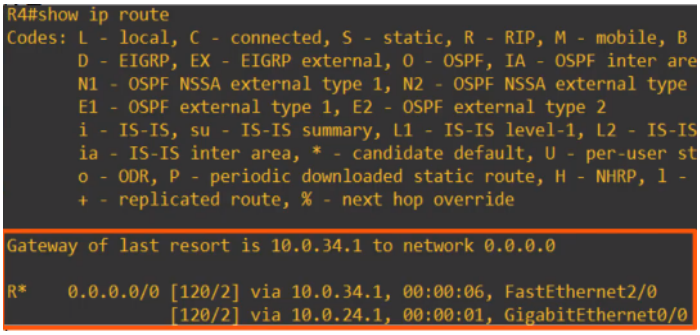
On peut voir que l'adresse a été ajouté à la table de routage :



A présent pour avertir les routeurs R2, R3 et R4 de l'ajout de cette route par défaut en RIP, on peut utiliser la commande :

```
R1(config-router)#default-information originate
```

On peut voir qu'avec la commande la table de routage à été mis à jour sur R4 :



Voici une commande pour afficher les protocoles utilisés :

```
R1#show ip protocols
```

Voyons à présent plus en détail le protocole EIGRP.

Enhanced Interior Gateway Routing Protocol (EIGRP) était propritaire Cisco, mais Cisco l'a maintenant publié gratuitement donc les autres vendeurs peuvent l'implémenter à leurs systèmes. Il y a tout de même une partie du protocole qui reste propriétaire.

Il est considéré comme un protocole « avancé »/ »hybride » de routage de vecteur distant.

Il est plus rapide que RIP en réaction aux changements dans le réseau. Il n'a pas la limite de bonds de 15 comme pour le protocole RIP. Les messages sont envoyés en utilisant une adresse de multicast : 224.0.0.10

EIGRP est le seule IGP qui peut faire fonctionner un load balance qui est inégal (par défaut il fonctionne le ECMP load-balancing par 4 chemins comme RIP)

Voici les commandes pour une configuration basique de EIGRP :

```
R1(config)#router eigrp 1
R1(config-router)#no auto-summary
R1(config-router)#passive-interface g2/0
R1(config-router)#network 10.0.0.0
R1(config-router)#network 172.16.1.0 0.0.0.15
```

Le numéro AS (autonomous system) doit correspondre entre les routeurs (ici c'est « 1 ») sinon ils ne formeront pas d'adjacence entre routeurs et ne partageront pas leurs informations.

La commande auto-summary doit être activé ou désactivé par défaut ce qui dépend de la version du routeur/IOS. S'il est activé il faut le désactiver.

La commande « network » va classifier l'adresse si l'on ne spécifie pas le masque, par exemple pour l'adresse 10.0.0.0 le masque sera automatiquement : /8

Pour spécifier le masque il faut lancer la commande ainsi : *network 172.16.1.0 0.0.0.15*

EIGRP utilise en fait un masque inversé voici comment cela fonctionne :

Tous les 1 du masque de sous réseau sont inversés en 0 dans le masque inversé. Tous les 0 du masque de sous réseau sont inversés par des 1 dans le masque inversé par exemple pour :

11111111.11111111.11111111.00000000 = 255.255.255.0

sera égal à cela en masque inversé :

00000000.00000000.00000000.11111111 = 0.0.0.255

c'est l'équivalent d'un masque en /24

un autre exemple :

11111111.11111111.00000000.00000000 = 255.255.0.0

sera égal à cela en masque inversé :

00000000.00000000.11111111.11111111 = 0.0.255.255

équivalent à un masque en /16

dernier exemple :

11111111.00000000.00000000.00000000 = 255.0.0.0

sera égal à cela en masque inversé :

00000000.11111111.11111111.11111111 = 0.255.255.255

équivalent à un masque en /8

Dans notre exemple pour convertir 255.255.255.240 on fais le calcul suivant :

11111111.11111111.11111111.11110000 = 255.255.255.240

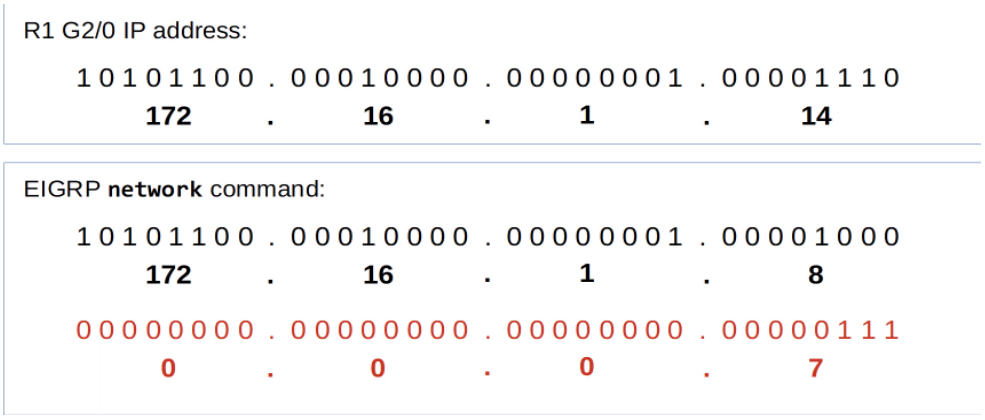
qui donnera en masque inversé :

00000000.00000000.00000000.00001111 = 0.0.0.15

équivalent à un /28

A présent il faut ajouter l'adresse du réseau pour qu'elle corresponde entre l'adresse EIGRP et l'adresse de l'interface.

Par exemple avec une commande qui inclut les deux adresses suivante :



Les deux adresses correspondent car l'adresse du routeur et l'adresse EIGRP sont les même pour tous les bit incluant le masque de sous réseau de l'adresse EIGRP en rouge donc EIGRP sera activé.

Voici un résultat de la commande show ip protocols lorsque EIGRP est activé :

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 172.16.1.14
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.1.0/28
  Passive Interface(s):
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.0.12.2            90      00:00:23
    10.0.13.2            90      00:00:23
  Distance: internal 90 external 170
```

La priorité du routeur ID est le suivant :

1) configuration manuelle

2) Adresse IP la plus haute sur une interface loopback

3) Adresse IP la plus haute sur une interface physique

Pour configurer le router id manuellement on lance la commande :

```
R1(config-router)#eigrp router-id 1.1.1.1
```

Voici à quoi ressemble la table de routage configuré avec EIGRP :

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
D        10.0.24.0/30 [90/3072] via 10.0.12.2, 00:11:09, GigabitEthernet0/0
D        10.0.34.0/30 [90/28416] via 10.0.13.2, 00:11:09, GigabitEthernet1/0
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/28 is directly connected, GigabitEthernet2/0
L        172.16.1.14/32 is directly connected, GigabitEthernet2/0
D     192.168.2.0/24 [90/3072] via 10.0.12.2, 00:11:09, GigabitEthernet0/0
      192.168.3.0/25 is subnetted, 1 subnets
D        192.168.3.0 [90/3072] via 10.0.13.2, 00:11:10, GigabitEthernet1/0
D     192.168.4.0/24 [90/3328] via 10.0.12.2, 00:11:09, GigabitEthernet0/0
```

Cours 26 : OSPF (Partie 1)

Dans ce cours nous verrons comment fonctionne le protocole OSPF avec d'abord une compréhension basique des opérations. Puis comment fonctionne OSPF Areas et nous ferons une configuration basique de OSPF.

Pour rappel OSPF est un protocole état des liens. Lorsque l'on utilise un protocole de routage état de lien, tous les routeur crée une carte de connectivité du réseau. Pour permettre cela chaque routeur avertit des informations à propos de son interface (connecté au réseau) à ses voisins. Ces avertissements passent vers d'autres routeurs jusqu'à se que tous les routeurs du réseau développent la même carte du réseau. Chaque routeur utilise indépendamment cette carte pour calculer la meilleure route vers chaque destination. Le protocole état des liens utilise plus de ressources (CPU) sur le routeur car plus d'informations sont partagés. Cependant le protocole état des liens à tendance à être plus rapide lorsqu'il y a des changements dans le réseau par rapport aux protocole de vecteur de distance.

OSPF est l'acronyme de Open Shortest Path First il utilise l'algorithme Shortest Path First un algorithme développé par le scientifique Edsger Dijkstra (c'est aussi l'algorithme Dijkstra).

Il existe 3 versions du protocole :

OSPFv1 (1989) : ancienne version plus vraiment en cours d'utilisation

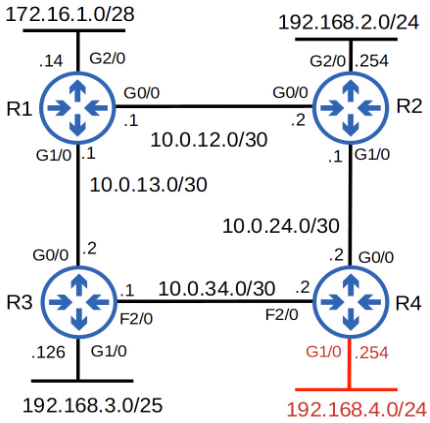
OSPFv2 (1998) : utilisé pour IPv4

OSPFv3 (2008) : utilisé pour IPv6 (peut aussi être utilisé pour IPv4 mais actuellement la v2 est utilisé pour IPv4)

Les routeurs stockent les informations à propos du réseau dans des LSA (Link State Advertisements) qui organise une structure appelé LSDB (Link State Database)

Les routeurs vont inonder les LSA jusqu'à se que tous les routeurs dans la zone OSPF ou OSPF area en anglais développe la même carte du réseau (LSDB)

Pour mieux comprendre utilisons ce schéma :



OSPF est ici activé sur le R4 avec l'interface G1/0. R4 crée un LSA pour dire à ses voisins à propos du réseau sur G1/0 Le LSA est ensuite inondé sur tout le réseau jusqu'à se que toutes les interfaces l'ait reçu. Ce qui résulte que tous les routeurs partagent le même LSDB.

Chaque routeur utilise ensuite l'algorithme SPF pour calculer la meilleur route vers 192.168.4.0/24

Chaque LSA a un temps définis de 30 minutes par défaut. Ce qui fais que un LSA est à nouveau partagé une fois le temps expiré.

En résumé pour OSPF il y a trois étapes dans le processus de partage de LSA pour déterminer la meilleure route pour chaque destination du réseau :

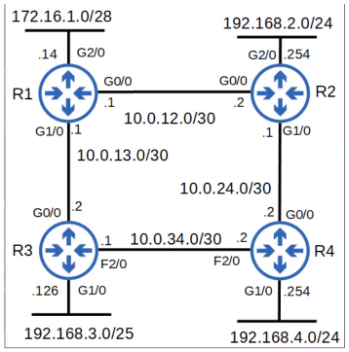
1) Devenir voisins avec d'autres routeurs connectés au même segment.

2) Echanger les LSA avec les routeurs voisins

3) Calculer la meilleure route possible et l'insérer dans la table de routage

OSPF utilise les zones ou « area » en anglais pour diviser le réseau. Les petits réseaux peuvent être en une seule zone sans qu'il y ait d'effet négatif sur les performances.

Par exemple sur ce réseau, il est possible de n'utiliser qu'une seule zone :



Dans de grand réseau, une seule zone peut avoir des effets négatifs comme :

- l'algorithme SPF peut prendre plus de temps pour calculer les routes

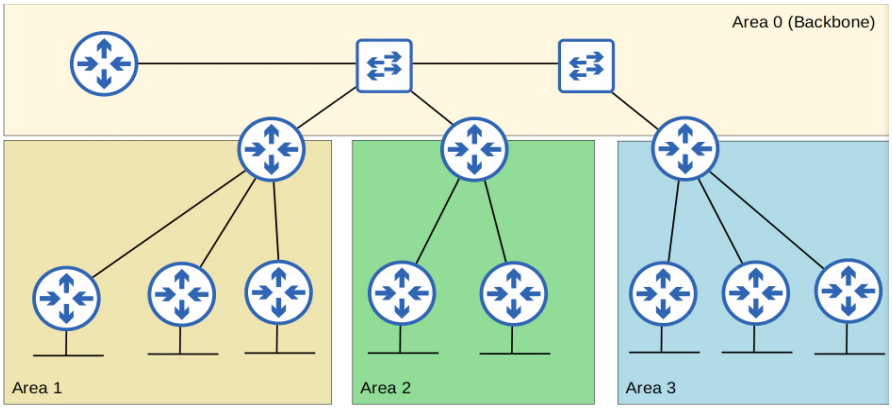
- l'algorithme SPF requière plus de puissance du processeur sur le routeurs

- Un grand LSDB peut prendre plus de mémoire dans le routeur

- N'importe quelle changement dans le réseau peut causer que chaque routeur inonde de LSA et relance l'algorithme SPF plusieurs fois.

En divisant un grand réseau OSPF en plusieurs on peut réduire ces effets négatifs.

Voici l'exemple d'un réseau qui est divisé en plusieurs zones :



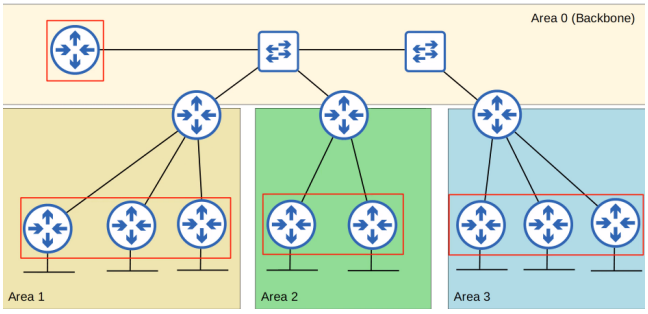
Pour mieux comprendre ce schéma il faut déjà définir se qu'est une « area ». Une zone ou « area » est un nombre de routeurs et de liens qui partagent le même LSDB.

Sur le schéma par exemple il y a en tout 4 zones ou « area » qui partagent le même LSDB.

La backbone area (Area 0) est une zone sur laquelle toutes les autres zones doivent être connectés.

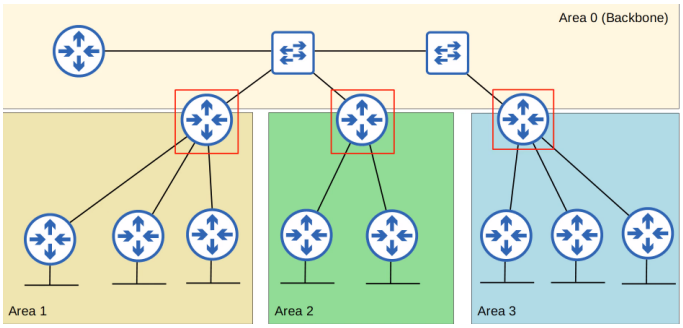
Les routeurs avec les interfaces dans une zone qui ne partagent pas d'autres zones sont appelé des routeurs internes.

Par exemple sur le schéma les routeurs internes sont ceux encadrés en rouge :



Ensuite les routeurs avec des interfaces sur plusieurs zones sont appelés les area border routers (ABR)

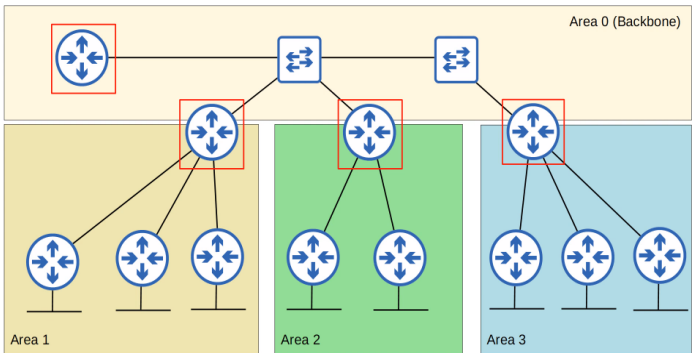
Sur le schéma se sont les routeurs encadrés en rouge :



Les ABR maintiennent un LSDB séparé pour chaque zone à laquelle ils sont connectés. Il est recommandé de connecter un ABR à un maximum de 2 zones. Connecter un ABR à plus de 3 zones peut surcharger le routeur.

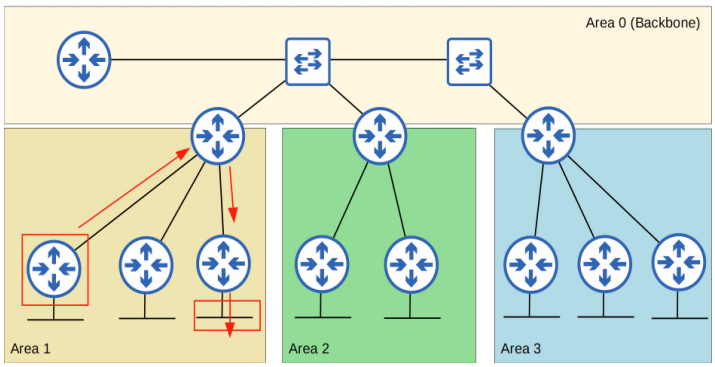
Les routeurs connectés à la backbone area (area 0) sont appelé les backbone routers.

Sur le schéma les backbone router sont encadrés en rouge :



Une intra-area route est une route vers une destination à l'intérieur de la même zone OSPF.

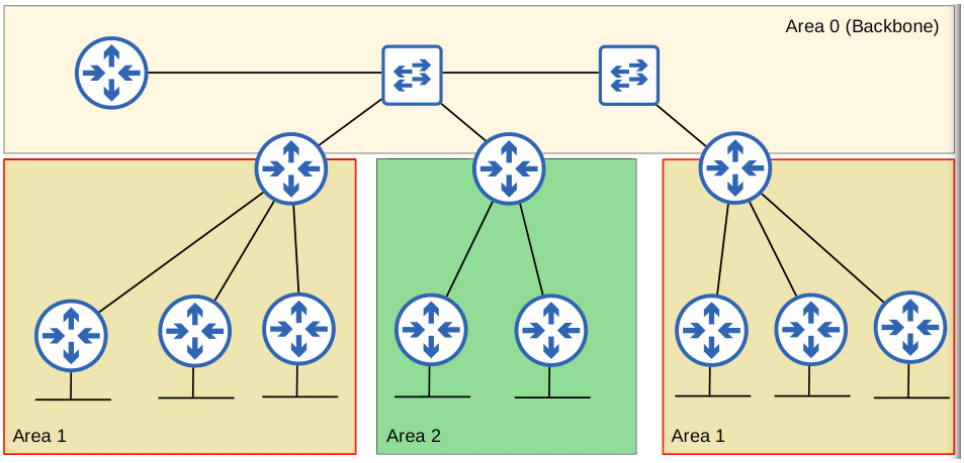
Par exemple sur cette exemple :



Une interarea route est une route vers une destination avec une zone OSPF différente ou extérieur.

Les zones OSPF doivent toujours être connectés entre elles et non pas divisés.

Dans cette exemple on peut voir deux zone séparés en deux une partie se trouve à gauche l'autre à droite. Ce type de schéma n'est pas fonctionnel avec OSPF.

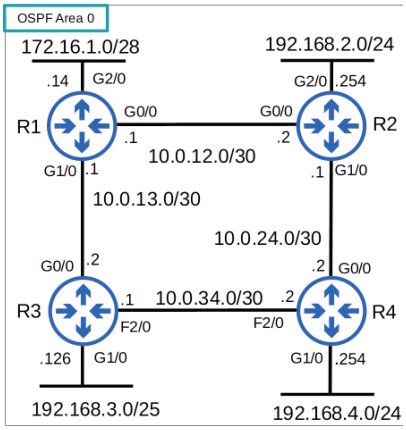


Toutes les zones OSPF doivent avoir au moins un ABR connecté à la backbone area.

Les interfaces OSPF du même sous réseau doivent être dans la même zone.

Voyons à présent comment faire une configuration OSPF basique.

Nous utiliserons cette topologie pour la configuration :



Commençons tout d'abord par configurer R1 :

```
R1(config)#router ospf 1
R1(config-router)#network 10.0.12.0 0.0.0.3 area 0
R1(config-router)#network 10.0.13.0 0.0.0.3 area 0
R1(config-router)#network 172.16.1.0 0.0.0.15 area 0
```

Le processus OSPF process ID est localement signifiant. Les routeurs avec différents process ID peuvent devenir des voisins OSPF.

Il faut spécifier l'area lorsque l'on utilise la commande « network » avec OSPF.

La commande network dis à OSPF de regarder pour n'importe quelle interface avec une adresse IP qui contient le classement spécifié dans la commande. D'activer OSPF sur l'interface avec l'interface spécifié.

Le routeur va ensuite essayer de devenir le voisin OSPF avec d'autre routeurs OSPF voisins d'activés.

Voici d'autres commandes utiles pour la configuration OSPF :

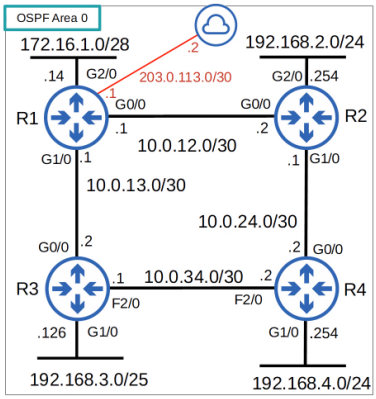
```
R1(config-router)#passive-interface g2/0
```

Cette commande est déjà utile avec les protocoles RIP et EIGRP, la commande « passive interface » dis au routeur d'arrêter d'envoyer des messages « hello » en dehors de l'interface.

Cependant le routeur va continuer d'envoyer des LSA pour informer ses voisins à propos du sous réseau configuré sur l'interface.

On devrait toujours pouvoir utiliser cette commande sur des interfaces qui n'ont aucuns voisins OSPF.

Voyons à présent comment avertir une route par défaut avec OSPF sur ce schéma :




```
R1(config)#ip route 0.0.0.0 0.0.0.0 203.0.113.2
```

Voici la commande pour avertir de la route par défaut :

```
R1(config-router)#default-information originate
```

L'ordre de priorité pour le router ID est :

1) Configuration manuelle

2) Adresse IP sur une interface loopback

3) Plus haute adresse IP d'une interface physique

La commande pour configurer manuellement le router ID est :

```
R1(config-router)#router-id 1.1.1.1
```

Il faut ensuite lancer cette commande pour mettre à jour la configuration :

```
R1#clear ip ospf process
```

La commande est légèrement différente que pour le protocole EIGRP

Un autonomous system boundary router (ASBR) est un routeur OSPF qui connecte le réseau OSPF à un réseau externe. R1 est connecté à internet. en utilisant la commande : « *default-information originate* » R1 devient le ASBR.

Pour changer l'administratice distance (AD) la commande est la même que pour EIGP :

```
R1(config-router)#distance 85
```

Cours 27 : OSPF (Partie 2)

Dans ce cours nous allons apprendre plus en détail comment fonctionne le protocole OSPF avec les Métrique OSPF (coût), et comment les routeurs deviennent voisins dans OSPF. Nous verrons en dernier plus de configuration avec OSPF.

Les métrique OSPF sont appelé : coûts

C'est automatiquement calculé basé sur la bande passante (vitesse) de l'interface.

C'est calculé en divisant la valeur de la bande passante de référence par la bande passante de l'interface.

La référence par défaut de la bande passante est de 100mbps. Les références sont donc :

Reference : 100 mbps / Interface : 10 mbps = coût de 10

Reference : 100 mbps / Interface : 100 mbps = coût de 1

Reference : 100 mbps / Interface : 1000 mbps = coût de 1

Reference : 100 mbps / Interface : 1000 mbps = coût de 1

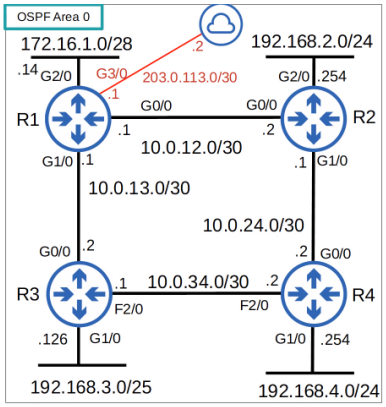
Toutes les valeurs en dessous de 1 sont convertis à 1

Toutefois FastEthernet, Gigabit Ethernet, 10Gig Ethernet, etc.. sont égal et ont toutes un coût de 1 par défaut.

On peut (et devrait) changer la bande passante de référence avec cette commande :

```
R1(config-router)#auto-cost reference-bandwidth megabits-par-secondes
```

Utilisons ce schéma pour mieux comprendre :



Par exemple avec la commande suivante entrée :

```
R3(config-router)#auto-cost reference-bandwidth 100000
```

La commande est entré en megabits par seconde (par défaut c'est de 100)

100000 / 100 = coût de 1000 pour FastEthernet

100000 / 1000 = coût de 100 pour GigEthernet

On devrait configurer une bande passante de référence plus haute que le lien rapide dans notre réseau (pour permettre des mis à jour future).

On devrait aussi configurer avec la même bande passante de référence sur tous les routeurs OSPF dans le réseau.

Le coût OSPF vers une destination est le coût total d'un 'outgoing/exit interface'

Par exemple les coûts de R1 pour joindre 192.168.4.0/24 est : 100 (R1 G0/0) + 100 (R2 G1/0) + 100 (R4 G1/0) = 300

Les interfaces loopback ont un coût de 1

Donc quelle serait le coût pour joindre 2.2.2.2 (interface loopback R2) ?

100 (R1 G1/0) + 1 (R2 L0) = 101

Voici la table de routage de R1 avant le changement de la bande passante de reference 100 :

```
Gateway of last resort is 203.0.113.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 203.0.113.2
      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0
      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/2] via 10.0.12.2, 00:00:26, GigabitEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/2] via 10.0.13.2, 00:00:26, GigabitEthernet1/0
      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/3] via 10.0.13.2, 00:00:16, GigabitEthernet1/0
                 [110/3] via 10.0.12.2, 00:00:16, GigabitEthernet0/0
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
O        10.0.24.0/30 [110/2] via 10.0.12.2, 00:00:16, GigabitEthernet0/0
O        10.0.34.0/30 [110/2] via 10.0.13.2, 00:00:16, GigabitEthernet1/0
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/28 is directly connected, GigabitEthernet2/0
L        172.16.1.14/32 is directly connected, GigabitEthernet2/0
O     192.168.2.0/24 [110/2] via 10.0.12.2, 00:00:16, GigabitEthernet0/0
      192.168.3.0/25 is subnetted, 1 subnets
O        192.168.3.0 [110/2] via 10.0.13.2, 00:00:16, GigabitEthernet1/0
O     192.168.4.0/24 [110/3] via 10.0.13.2, 00:00:04, GigabitEthernet1/0
                     [110/3] via 10.0.12.2, 00:00:04, GigabitEthernet0/0
      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C        203.0.113.0/30 is directly connected, GigabitEthernet3/0
L        203.0.113.1/32 is directly connected, GigabitEthernet3/0
```

Et voici la table de routage R1 après le changement de la bande passante de référence :

```
Gateway of last resort is 203.0.113.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 203.0.113.2
      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0
      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/101] via 10.0.12.2, 00:34:04, GigabitEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/101] via 10.0.13.2, 00:33:54, GigabitEthernet1/0
      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/201] via 10.0.12.2, 00:33:54, GigabitEthernet0/0
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
O        10.0.24.0/30 [110/200] via 10.0.12.2, 00:33:54, GigabitEthernet0/0
O        10.0.34.0/30 [110/1100] via 10.0.13.2, 00:33:44, GigabitEthernet1/0
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/28 is directly connected, GigabitEthernet2/0
L        172.16.1.14/32 is directly connected, GigabitEthernet2/0
O     192.168.2.0/24 [110/200] via 10.0.12.2, 00:34:04, GigabitEthernet0/0
      192.168.3.0/25 is subnetted, 1 subnets
O        192.168.3.0 [110/300] via 10.0.13.2, 00:33:54, GigabitEthernet1/0
O     192.168.4.0/24 [110/300] via 10.0.12.2, 00:26:46, GigabitEthernet0/0
      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C        203.0.113.0/30 is directly connected, GigabitEthernet3/0
L        203.0.113.1/32 is directly connected, GigabitEthernet3/0
```

Il n'y a plus qu'une route pour 192.168.4.0/24 après le changement.

La commande pour modifier manuellement le coût OSPF est :

```
R1(config)#interface g0/0
R1(config-if)#ip ospf cost 10000
```

Une autre option pour changer le coût OSPF d'une interface est de changer la bande passante de l'interface avec la commande : bandwidth

La formule pour calculer le coût OSPF est la bande passante de référence / bande passante de l'interface.

Même si la bande passante correspond à la vitesse de la bande passante par défaut, changer la bande passante de l'interface ne change pas la vitesse à laquelle l'interface fonctionne.

La bande passante est juste une valeur qui est utilisé pour calculer le coût OSPF, la métrique EIGRP, etc..

Pour changer la vitesse à laquelle l'interface fonctionne, on utilise la commande : « speed »

Puisque la valeur de la bande passante est utilisée dans d'autres calculs, il n'est pas recommandé de changer cette valeur pour modifier le coût de l'interface OSPF.

Il est recommandé de changer la bande passante de référence, et puis d'utiliser la commande « ip ospf cost » pour changer le coût d'une interface individuellement.

Pour changer la valeur de la bande passante on utilise la commande :

```
R1(config-if)#bandwidth 100000
```

Cette commande est utilisé avec les kilobits par secondes contrairement à la bande passante de référence qui est calculé en mbps.

En résumé il y a trois manière de changer le coût OSPF :

1) changer la bande passante de référence :

```
R1(config-router)#auto-cost reference-bandwidth megabits-par-secondes
```

2) configuration manuelle :

```
R1(config-if)#ip ospf cost coût
```

3) changer la bande passante de l'interface :

```
R1(config-if)#bandwidth {kilobis-par-secondes}
```

Voici une commande pour vérifier rapidement le coût des interfaces OSPF :

```
R3#show ip ospf interface brief
```

Il faut être certain que les routeurs deviennent avec succès des voisins OSPF c'est la tâche principale dans la configuration et la résolution des problèmes OSPF.

Une fois les routeurs devenus voisins, ils font automatiquement le travail de partager les informations réseau, calculer les routes, etc..

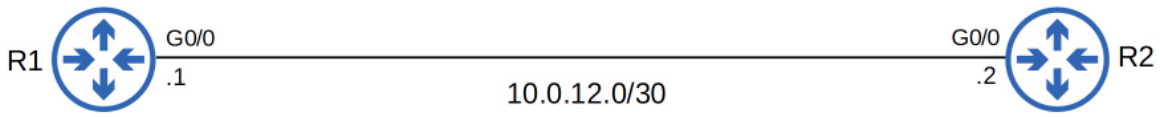
Lorsque OSPF est activé sur une interface, le routeur commence par envoyer des messages de hello en dehors de l'interface à des intervalles régulier (déterminés par des temps définis de hello). Ils sont utilisés pour introduire le routeur à des voisins OSPF.

Le temps définis des hello par défaut est de 10 secondes sur une connexion Ethernet.

Les messages Hello sont envoyés en multicast vers 224.0.0.5 (adresse de multicast pour tous les routeurs OSPF)

Les messages OSPF sont encapsulés dans une entête IP avec pour valeur 89 dans la partie protocole.

Prenons pour exemple avec cette topologie réseau pour mieux comprendre :



OSPF est activé sur l'interface G0/0 R1. Il envoie des messages de hello vers 224.0.0.5

Il ne connais rien à propos des voisins OSPF, donc le statut des voisins est Down (éteint)

Lorsque R2 reçois des paquet hello, il va ajouter une entrée pour R1 à sa propre table de voisins OSPF.

Dans la table des voisins R2, la relation avec R1 est à présent dans l'état « Init » (initialisation)

L'état « Init » signifie que les paquet Hello sont reçus mais son propre router ID n'est pas le paquet Hello.

Après cela R2 envoie des paquets Hello contenant le Router ID des deux routeurs.

R1 va insérer R2 dans sa table de voisins OSPF dans l'état 2-way (2 chemins)

R1 va envoyer un autre message Hello, cette fois contenant le Router ID de R2. A présent les deux routeurs sont dans l'état « 2-way »

L'état 2-way signifie que le routeur a reçus un paquet Hello avec son propre Router ID à l'intérieur.

Si les deux routeurs ont le statut 2-way state, cela signifie que toutes les conditions sont réunis pour qu'ils deviennent des voisins OSPF.

Ils peuvent à présent partager des LSA pour construire un LSDB commun.

Dans certain type de réseau, un DR (Designated router) et un BDR (Backup Designated Router) vont être élus à ce niveau.

Les deux routeurs vont maintenant se préparer à échanger des informations à propos de leurs LSDB.

Avant cela ils doivent choisir lequel va démarrer l'échange. Il font cela avec le statut « Exstart »

Le router avec le Router ID le plus haut devient le Master (maître) et initie la connexion, et le routeur avec le Router ID le plus bas devient le Slave (esclave).

Pour décider lequel est le master/slave, ils échangent des paquets DBD (Database Description)

Dans l'état « d'échange », les routeurs échangent des DBD qui contiennent une liste de LSA dans leurs LSDB.

Ces DBD n'incluent pas des informations détaillé à propos des LSA, juste des informations basique.

Les routeurs comparent l'information dans le DBD qu'ils reçoivent à l'information de leur propre LSDB pour déterminer quelle LSA ils doivent recevoir de leurs voisins.

Après avoir échanger les DBD ils changent de statut pour :

L'état de Loading (chargement), dans ce statut les routeurs envoient des messages Link State Request (LSR) pour faire la requête à leurs voisins d'un LSA qu'ils n'auraient pas.

Les LSA sont envoyés dans des messages Link State Update (LSU)

Les routeurs envoient au final des messages LSAck pour confirmer qu'ils ont reçus le LSA.

Dans l'état du statut Full, les routeurs ont une adjacence OSPF entière et des LSDB identiques.

Ils continuent d'envoyer et écouter des paquet Hello (toutes les 10 secondes par défaut) pour maintenir les voisins adjacent.

A chaque fois qu'un paquet Hello est reçu, le 'Dead' Timer (de 40 secondes par défaut) est remis.

Si le Dead timer compte jusqu'à 0 et qu'aucun message Hello n'est reçu, le voisin est supprimé.

Les routeurs vont continuer de partager des LSA lorsque le réseau change pour être sûre que chaque routeur à une carte complète et précise du réseau (LSDB).

En résumé il a 7 statut différents pour l'ajout d'un voisin OSPF :

1) Down 2) Init 3) 2-way 4) Exstart 5) Exchange 6) Loading 7) Full

Les trois premiers Statuts permettent que les routeurs deviennent voisins entre eux, les 3 statut suivant permettent l'échange des LSA.

Voici un tableau qui résume les statut des messages OSPF :

Type	Nom	But
1	Hello	Découverte des voisins.
2	Database Description (DBD)	Résumé de la base de données LSDB envoyée aux routeurs. Utilisé pour vérifier si les LSDB de chaque routeur sont identiques.
3	Link-State Request (LSR)	Demande spécifique de LSA à un voisin.
4	Link-State Update (LSU)	Envoi de LSA spécifiques aux voisins.
5	Link-State Acknowledgement (LSAck)	Confirmation de réception d'un message par un routeur.

Voici quelques commandes utiles pour OSPF :

Pour vérifier les voisins OSPF on lance la commande :

```
R1#show ip ospf neighbor
```

Il est possible d'activer OSPF directement sur une interface avec cette commande :

```
R1(config-if)#ip ospf {process-id} area {zone}
```

Il est aussi possible de configurer toutes les interface OSPF en mode passive interface avec cette commande :

```
R1(config-router)#passive-interface default
```

Il est ensuite possible de désactiver des interface spécifique :

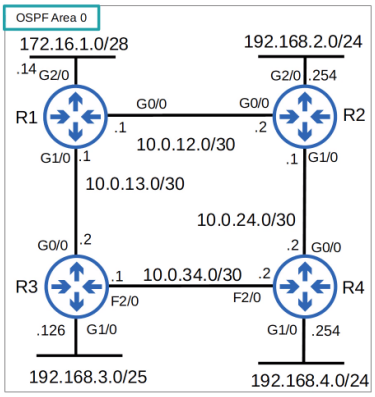
```
R1(config-router)#no passive-interface {int-id}
```

Cours 28 : OSPF (Partie 3)

Dans ce cours nous allons apprendre plus en détail comment fonctionne OSPF, Nous commencerons par voir les différents type de réseau OSPF, puis nous verrons les prérequis pour qu'il y ait des voisins/adjacence OSPF. Nous verrons ensuite les différents types de LSA.

Rappelons d'abord se qu'est une interface loopback. Une interface loopback est une interface virtuel sur un routeur. Il est toujours en up/up (jusqu'à se qu'on ne l'éteigne manuellement), ce type d'interface n'est pas dépendant d'une interface physique. Donc il fournit une adresse IP qui peut être utilisé pour joindre/identifier le routeur.

Utilisons cette topologie réseau :



Disons que R1 ne possède pas d'interface loopback pour le moment et que R4 reçoit un paquet avec pour destination 10.0.13.1/30, le paquet sera partagé à R1 en passant par le routeur R3.

Si l'interface G1/0 de R1 ne fonctionne plus le routeur ne sera plus capable de redistribuer le paquet à R1.

A présent si R1 a une interface loopback avec pour adresse : L0 : 1.1.1.1

Et que R4 reçoit un paquet à partager à R1, R4 pourra toujours continuer à partager le paquet par R2 même si l'interface G1/0 de R1 ne fonctionne plus.

C'est pourquoi c'est une bonne idée de configurer une interface loopback sur un routeur.

Le type de réseau dans OSPF se réfère au type de connexions entre les voisins OSPF (Ethernet, etc..)

Il existe 3 principaux types de réseaux OSPF :

- Broadcast : Activé par défaut sur Ethernet et les interfaces FDDI (Fiber Distributed Data Interfaces)

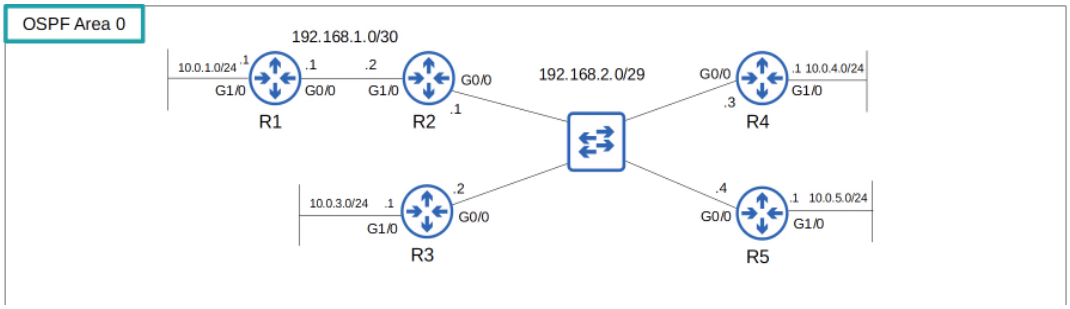
- Point to Point : Activé par défaut sur PPP (Point to Point Protocol) et les interfaces HDLC (High Level Data Link Control)

- Non Broadcast : Activé par défaut sur les relais de trames et les interfaces X.25

Nous allons voir principalement comment fonctionne les réseaux en Broadcast et en Point to point

Sur cette topologie réseau le type de réseau en Broadcast est activé sur Ethernet et les interfaces FDDI par défaut.

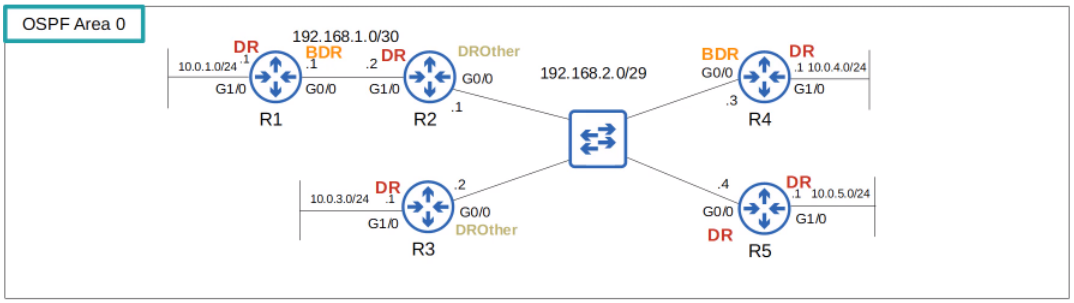
Les routeurs découvrent dynamiquement les voisins en envoyant/écoutant des messages Hello OSPF en utilisant l'adresse de multicast 224.0.0.5



Un DR (Designated Router) et BDR (Backup Designated Router) doivent être élu sur chaque sous réseau (il n'y a que un DR s'il n'y a pas de voisins OSPF, par exemple les interfaces G1/0 de R1)

Les routeurs qui ne sont pas les DR ou BDR deviennent les DR Other

Sur la topologie précédent par exemple les DR et BDR seront répartis de la manière suivante :



L'élection des DR/BDR ont l'ordre de priorité suivant :

1 : La plus haute interface de priorité

2 : Le plus haut Router ID OSPF

- La « première place » devient le DR du sous réseau, la « seconde place » devient le BDR

- L'interface par défaut de priorité des interface OSPF est 1 sur toutes les interfaces.

Ces informations peuvent être affiché avec la commande « show ip ospf interface g0/0 » :

```
R5#show ip ospf interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 192.168.2.4/29, Area 0, Attached via Network Statement
  Process ID 1, Router ID 5.5.5.5, Network Type BROADCAST, Cost: 1
  Topology-MTID    Cost    Disabled    Shutdown      Topology Name
        0           1          no          no            Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 5.5.5.5, Interface address 192.168.2.4
  Backup Designated router (ID) 4.4.4.4, Interface address 192.168.2.3
```

La commande pour changer la priorité des interfaces est :

```
R2(config)#int g0/0
R2(config-if)#ip ospf priority 255
```

Le maximum est de 255

Si l'on place la priorité de l'interface OSPF à 0, le routeur de pourra pas être le DR/BDR pour le sous réseau.

```
R5#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
R5#
*Aug 22 04:25:05.307: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
*Aug 22 04:25:05.311: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
*Aug 22 04:25:05.311: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
R5#
*Aug 22 04:25:13.903: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
*Aug 22 04:25:13.907: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R5#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2         255   FULL/BDR        00:00:37    192.168.2.1     GigabitEthernet0/0
3.3.3.3           1   2WAY/DROTHER    00:00:37    192.168.2.2     GigabitEthernet0/0
4.4.4.4           1   FULL/DR         00:00:39    192.168.2.3     GigabitEthernet0/0
```

Ici le processus OSPF a été réinitialisé, on peut voir que dans la section encadré en jaune, R4 est devenu le DR et non pas R2. R2 est devenu le BDR.

R4 devient le DR non plus R2. R2 devient le BDR.

Lorsque le DR ne fonctionne plus, le BDR devient le nouveau DR. Ensuite le prochain BDR est élu.

R3 est le DR Other et est stable dans le 2-way state.

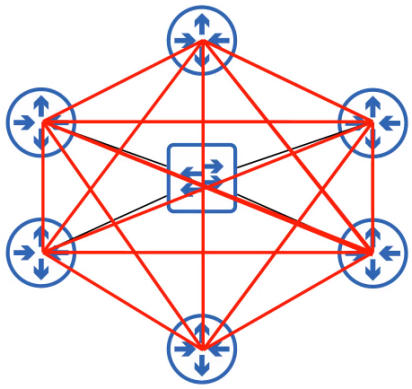
Le DR Other (R3 et R5 dans ce réseau) vont uniquement changer à l'état FULL avec le DR et BDR. Les états voisins avec les autres DR Others seront des 2-way

Dans le type de réseau en Broadcast, les routeurs vont uniquement former des adjacence full OSPF avec le DR et BDR du segment.

Les routeurs échangent uniquement des LSA avec le DR et BDR. Les DR Other n'échangent pas de LSA entre eux.

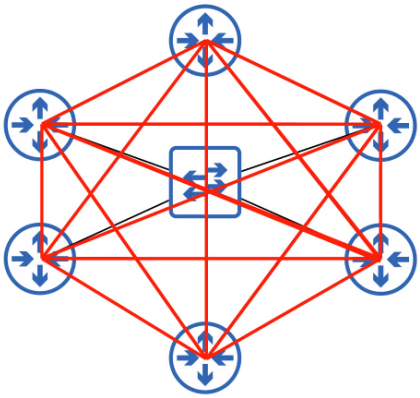
Tous les routeurs auront toujours le même LSDB, mais cela va réduire le montant pour que le réseau soit « inondé » de LSA.

Voici un exemple pour un réseau :



Dans cette exemple tous les routeurs sont connectés entre eux et vont inonder de LSA le réseau

Par contre si les routeurs échangent uniquement les informations entre DR et BDR le trafique sera moins important comme suit :



Les messages vers le DR/BDR sont en multicast en utilisant les adresses 224.0.0.6

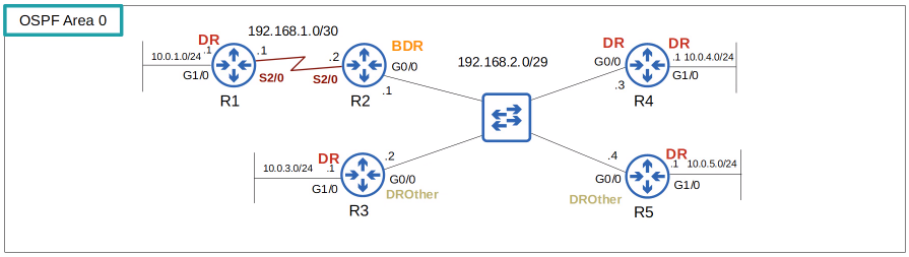
Le DR et BDR forment une seule FULL adjacence avec tous les routeurs du sous réseau.

Les DR Others vont former une FULL adjacence seulement avec les DR/BDR

On peut voir toutes ces informations directement avec la commande show ip ospf interface g0/0 :

```
R3#show ip ospf interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 192.168.2.2/29, Area 0, Attached via Network Statement
  Process ID 1, Router ID 3.3.3.3, Network Type BROADCAST, Cost: 1
  Topology-MTID    Cost    Disabled    Shutdown      Topology Name
        0           1         no          no            Base
  Transmit Delay is 1 sec, State DROTHER, Priority 1
  Designated Router (ID) 4.4.4.4, Interface address 192.168.2.3
  Backup Designated router (ID) 2.2.2.2, Interface address 192.168.2.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:09
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 3, Adjacent neighbor count is 2
    Adjacent with neighbor 2.2.2.2  (Backup Designated Router)
    Adjacent with neighbor 4.4.4.4  (Designated Router)
  Suppress hello for 0 neighbor(s)
```

Voyons à présent comment sont formés les type de réseau OSPF Point to Point avec ce schéma :



Il faut utiliser une interface serial en utilisant PPP ou l'encapsulation HDLC par défaut.

Les routeurs découvrent dynamiquement les voisins en envoyant/écoutant des messages Hello OSPF en utilisant l'adresse de multicast 224.0.0.5

Un DR et BDR ne sont pas élus

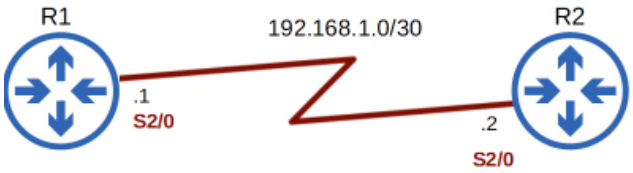
Cette encapsulation est utilisé pour les connexion « point to point »

Cependant il n'y a pas de point à choisir un DR et BDR

Les deux routeurs vont former un adjacence Full entre eux.

Les interfaces serial se présentent de cette manière :

Pour expliquer les connexion serial voici comment configurer l'interface s2/0 :



Les commandes à utiliser pour R1 :

```
R1(config)#interface s2/0
R1(config-if)#clock rate 64000
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown
```

Avec les connexion Serial, un côté de la connexion serial fonctionne comme DCE (Data Communications Equipment), et l'autre côté fonctionne comme DTE (Data Terminal Equipment)

Le côté DCE doit spécifier le « clock rate » (ou vitesse) de la connexion

Dans la topologie précédente R1 est le DCE et doit dire à R2 à quelle vitesse la connexion doit se faire.

Les interfaces Ethernet utilisent la commande « speed » pour configurer la vitesse de l'interface. L'interface Serial utilise la commande « clock rate ».

L'encapsulation par défaut d'une interface Serial est HDLC

On peut le voir avec la commande show interface s2/0 :

```
R1#show interface s2/0
Serial2/0 is up, line protocol is up
  Hardware is M4T
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC  crc 16, loopback not set
```

Si l'on change l'encapsulation, cela doit fonctionner sur les deux fins ou l'interface ne fonctionnera plus.

Pour vérifier si une interface fonctionne comme DCE il faut lancer la commande :

```
R1#show controllers s2/0
```

Il est aussi possible de configurer directement le type d'interface du réseau en utilisant la commande suivante :

```
R1(config-if)#ip ospf network {type}
```

Il faut remplacer *type* par le type de réseau que l'on veux : broadcast, non-broadcast, point-to-multipoint, point-to-point

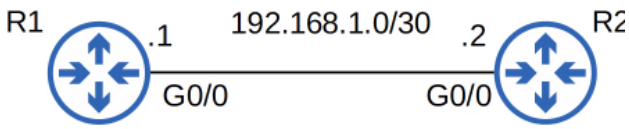
Par exemple si deux routeurs sont directement connectés avec un lien Ethernet, il n'y a pas besoin de DR/BDR. On peut configurer le type de connexion point-to-point dans ce cas.

Il est à noter que pas tous les types de réseaux fonctionnent sur tous les types de liens (par exemple, une connexion serial ne peut pas utiliser un type de réseau broadcast)

Voici un tableau pour résumer :

Broadcast	Point to point
Ethernet par défaut, interfaces FDDI	HDLC par défaut, interfaces PPP (serial)
DR/BDR élu	Pas de DR/BDR
Les voisins sont découverts dynamiquement	Les voisins sont découverts dynamiquement
Le timers par défaut est : Hello 10, Dead 40	Le timers par défaut est : Hello 10, Dead 40

Voyons à présent les prérequis pour que des interfaces OSPF deviennent voisins, utilisons cette topologie :



Lorsque l'on lance la commande pour vérifier les zones de chaques routeur on voit que R1 est dans l'area 0 et que R2 est dans l'area 1, ils ne sont donc pas voisins

```
R1#show running-config | section ospf
router ospf 1
 network 192.168.1.0 0.0.0.3 area 0
```

```
R2#show running-config | section ospf
router ospf 1
 network 192.168.1.0 0.0.0.3 area 1
```

```
R1#show ip ospf neighbor
R1#
R1#
```

```
R2#show ip ospf neighbor
R2#
R2#
```

Pour qu'ils deviennent voisins il faut changer la zone du routeur 2 pour qu'elle soit la même que le routeur 1 on peut voir que les deux interfaces sont voisins à présent :

```
R1#show running-config | section ospf
router ospf 1
 network 192.168.1.0 0.0.0.3 area 0
```

```
R2#show running-config | section ospf
router ospf 1
 network 192.168.1.0 0.0.0.3 area 0
```

```
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.2       1   FULL/BDR        00:00:34    192.168.1.2     GigabitEthernet0/0
R1#
```

```
R2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1       1   FULL/DR         00:00:39    192.168.1.1     GigabitEthernet0/0
R2#
```

Un deuxième prérequis pour que les routeurs deviennent voisins est qu'ils doivent avoir le même masque de sous réseau :

On peut voir ici que les routeurs R1 et R2 ne sont pas voisins :

```
R1#show running-config | section ospf
router ospf 1
 network 192.168.1.0 0.0.0.3 area 0
R1#
```

```
R2#show running-config | section ospf
router ospf 1
 network 192.168.2.0 0.0.0.3 area 0
R2#
```

```
R1#show ip ospf neighbor
R1#
R1#
```

```
R2#show ip ospf neighbor
R2#
R2#
```

Pour qu'ils le deviennent il est nécessaire de les configurer dans les même masques de sous réseau :

```
R1#show running-config | section ospf
router ospf 1
 network 192.168.1.0 0.0.0.3 area 0
R1#
```

```
R2#show running-config | section ospf
router ospf 1
 network 192.168.1.0 0.0.0.3 area 0
R2#
```

```
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.2       1   FULL/BDR        00:00:34    192.168.1.2     GigabitEthernet0/0
R1#
```

```
R2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1       1   FULL/DR         00:00:39    192.168.1.1     GigabitEthernet0/0
R2#
```

Un troisième prérequis est que le processus OSPF ne doit pas être éteint, par exemple on peut voir que l'interface ospf est éteinte il suffit de la rallumer avec la commande : « no shutdown »

```
R2(config)#router ospf 1
R2(config-router)#shutdown
R2(config-router)#
*Aug 23 03:43:31.719: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Int
R2(config-router)#do show ip ospf neighbor
R2(config-router)#
```

```
R2(config-router)#no shutdown
R2(config-router)#
*Aug 23 03:49:52.931: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R2(config-router)#do show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1       1   FULL/DR         00:00:38    192.168.1.1     GigabitEthernet0/0
R2(config-router)#
```

Le quatrième prérequis est que les Router ID doivent être unique.

Par exemple ici les routeurs voisins ont le même router ID.

Le router ID de R2 a été changé pour être le même que celui de R1 et ont peut voir que les deux routeurs ne sont plus voisins :

```
R2(config-router)#router-id 192.168.1.1
% OSPF: Reload or use "clear ip ospf process" command, for this to take effect
R2(config-router)#end
R2#clear ip
*Aug 23 03:57:58.835: %SYS-5-CONFIG_I: Configured from console by console
R2#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
R2#
*Aug 23 03:58:04.055: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or
R2#
*Aug 23 03:58:06.495: %OSPF-4-DUP_RTRID_NBR: OSPF detected duplicate router-id 192.168.1.1 from 192.168.1.1 on interface GigabitEthernet0/0
R2#show ip ospf neighbor
R2#
```

Pour mettre un Router ID différents on utilise la commande :

```
R2(config-router)#no router-id
R2(config-router)#
*Aug 23 04:10:10.207: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R2(config-router)#do show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1       1   FULL/DR         00:00:35    192.168.1.1     GigabitEthernet0/0
R2(config-router)#
```

Le cinquième prérequis est que les Hello et Dead timers doivent correspondre entre routeurs.

Par exemple dans cette exemple les interval Hello et dead ne correspondent pas et les deux routeurs ne sont plus voisins :

```
R2(config-if)#ip ospf hello-interval ?
  <1-65535>  Seconds

R2(config-if)#ip ospf hello-interval 5
R2(config-if)#ip ospf dead-interval ?
  <1-65535>  Seconds
  minimal    Set to 1 second

R2(config-if)#ip ospf dead-interval 20
R2(config-if)#
*Aug 23 04:29:30.623: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Dead timer expired
R2(config-if)#do show ip ospf neighbor
R2(config-if)#
```

Pour remettre les hello/dead timer par défaut on utilise les commandes :

```
R2(config-if)#no ip ospf hello-interval
R2(config-if)#no ip ospf dead-interval
R2(config-if)#
*Aug 23 04:31:32.727: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R2(config-if)#do show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1       1   FULL/BDR        00:00:35    192.168.1.1     GigabitEthernet0/0
R2(config-if)#
```

Le sixième prérequis est que les paramètres d'authentification doivent correspondre, par exemple le mot de passe ospf est ici configuré comme suit :

```
R2(config-if)#ip ospf authentication-key jeremy
R2(config-if)#ip ospf authentication
R2(config-if)#
*Aug 23 04:56:28.435: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: D
R2(config-if)#do show ip ospf neighbor
R2(config-if)#
```

Mais les mots de passe entre les routeurs R1 et R2 ne correspondent pas. Donc l'interface ne fonctionne plus. Il faut soit désactiver l'authentification sur R2 ou bien l'ajouter sur R1. On choisi de supprimer l'authentification et les deux routeurs sont à nouveau voisins :

```
R2(config-if)#no ip ospf authentication
R2(config-if)#no ip ospf authentication-key jeremy
*Aug 23 04:59:37.315: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R2(config-if)#do show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1       1   FULL/BDR        00:00:34    192.168.1.1     GigabitEthernet0/0
R2(config-if)#
```

Un septième prérequis est que les paramètres IP MTU doivent correspondre.

```
R2(config-if)#ip mtu ?
  <68-1500>  MTU (bytes)

R2(config-if)#ip mtu 1400
R2(config-if)#do show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1       1   FULL/BDR        00:00:34    192.168.1.1     GigabitEthernet0/0
R2(config-if)#do clear ip ospf process
Reset ALL OSPF processes? [no]: yes
R2(config-if)#
*Aug 23 05:16:07.474: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
R2(config-if)#do show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1       1   EXSTART/DR      00:00:38    192.168.1.1     GigabitEthernet0/0
```

Ici on peut voir que le paramètre ip mtu à été activé et que le routage OSPF reste sur le statut : "EXSTART"

```
*Aug 23 05:21:12.946: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from EXSTART to DOWN, Neighbor Down: Too many retransmissions
R2(config-if)#
*Aug 23 05:22:12.946: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from DOWN to DOWN, Neighbor Down: Ignore timer expired
```

Le mode MTU est donc désactivé avec la commande suivante et les routeurs sont à nouveau voisins :

```
R2(config-if)#no ip mtu
R2(config-if)#
*Aug 23 05:25:49.362: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
```

Le huitième et dernier prérequis est que le type de réseau OSPF doit correspondre.

Ici une interface loopback a été configuré avec un type de connexion point to point :

```
R2(config)#interface l0
R2(config-if)#
*Aug 23 05:52:53.898: %LINK-3-UPDOWN: Interface Loopback0, changed state to up
*Aug 23 05:52:54.898: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
R2(config-if)#ip address 2.2.2.2 255.255.255.255
R2(config-if)#router ospf 1
R2(config-router)#network 2.2.2.2 0.0.0.0 area 0
R2(config-router)#interface g0/0
R2(config-if)#ip ospf network point-to-point
R2(config-if)#
*Aug 23 05:53:34.818: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interfac
*Aug 23 05:53:34.914: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R2(config-if)#do show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.1       0   FULL/  -        00:00:36    192.168.1.1     GigabitEthernet0/0
R2(config-if)#
```

On peut voir que le statut de l'interface est FULL. Et que les deux routeurs ne sont pas voisins.

On lance la commande suivante pour vérifier sur R1 et on peut voir que l'adresse de R2 n'est pas présente dans la table de routage de R1 :

```
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.1.2       1   FULL/BDR        00:00:31    192.168.1.2     GigabitEthernet0/0
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
R1#
```

Ceci est dû au fais que les deux routeurs utilisent des types de connexions différentes.

Voici en résumé les 8 prérequis que nous avons pu voir :

1) Le numéro de zone doit correspondre

2) Les interfaces doivent être dans le même sous réseau

3) Le processus OSPF ne doit pas être éteint

4) Les Router ID OSPF doivent être les même

5) Les timer Hello et Dead doivent correspondre entre routeurs

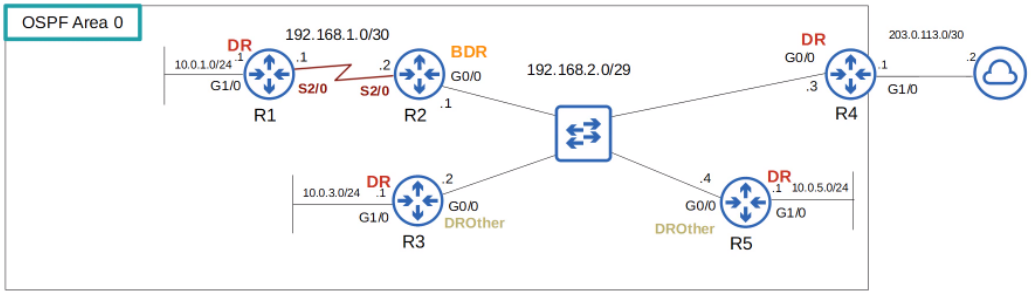
6) L'authentification doit être la même entre routeurs

7) Les paramètres IP MTU doivent correspondre

8) Le type de réseau OSPF doit correspondre

Voyons à présent le dernier sujet qui concerne les type de LSA OSPF

Nous utiliserons le schéma suivant :



Le OSPF LSDB est fais de LSA, il existe 11 types de LSA mais il n'y en a que 3 que l'on doit connaître principalement :

Type 1 (Router LSA) : Tous les routeurs OSPF génèrent ce type de LSA, il identifie le routeur en utilisant son Router ID, il liste aussi le réseau attaché aux interfaces routeur OSPF activés

Type 2 (Network LSA) : Ce LSA est généré par le DR de chaque réseau « multi-access », il liste les routeurs qui sont ratachés au réseau multi accès.

Type 5 (AS External LSA) : Ce type de LSA est généré par les ASBR pour décrire les routes vers une destination en dehors du AS (domaine OSPF)

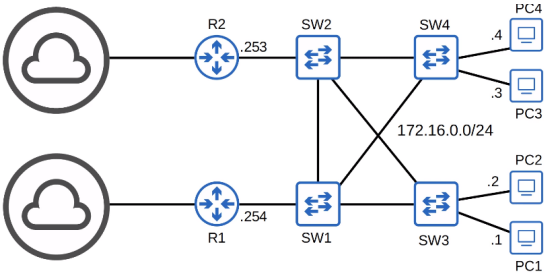
Avec cette commande il est possible de vérifier quelle type de LSA est utilisé :

```
R1#show ip ospf database

            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         1396        0x80000002 0x00FE8D 4
2.2.2.2         2.2.2.2         932         0x80000005 0x00753F 4
3.3.3.3         3.3.3.3         974         0x80000004 0x00AD70 2
4.4.4.4         4.4.4.4         975         0x80000005 0x004CC2 2
5.5.5.5         5.5.5.5         976         0x80000004 0x00D212 3

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.2.3     4.4.4.4         932         0x80000002 0x00740D

                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
0.0.0.0         4.4.4.4         273         0x80000002 0x00C0E0 1
R1#
```

Cours 29 : FHRP

Dans ce cours nous allons apprendre ce qu'est le protocole First Hop Redundancy Protocols (FHRP) Nous verrons tout d'abord le but de FHRP puis nous expliquerons comment fonctionne 2 FHRP : Le HSRP (Hot Standby Router Protocol), VRRP (Virtual Router Redundancy Protocol), GLBP (Gateway Load Balancing Protocol) puis nous verrons en dernier temps comment faire une configuration basique de HSRP.

Nous utiliserons la configuration suivante dans nos exemples :



On peut voir qu'il y a deux réseaux mis en place avec deux routeurs différents disons que le Routeur 1 est configuré comme étant la passerelle par défaut et que tous les PC ont configurés le Routeur 1 comme passerelle par défaut, chaque fois qu'un PC veut communiquer avec Internet il passera par le Routeur 1 qui utilise la passerelle par défaut. Mais que se passe t-il si R1 ne fonctionne plus ?

Il y aura tout de même un routeur de rechange qui est le Routeur 2, le seul problème de configuration sera que la passerelle est pour le moment resté configuré sur tous les PC sur le Routeur 1, Donc comment faire pour faire passer le routeur 2 comme passerelle par défaut ?

C'est le rôle du protocole FHRP.

Sur Wikipédia il est écrit que le protocole FHRP est un protocole conçu pour protéger la passerelle par défaut utilisé dans un sous réseau en permettant à deux ou plus de routeurs de fournir une récupération pour cette adresse dans le cas de dysfonctionnement du routeur principale. Le routeur de récupération utilisera cette adresse en quelques secondes.

Pour que cela marche les deux routeurs utilisent en vérité un VIP (Virtual IP)

puis on configure les autres PC pour qu'ils utilisent cette même adresse VIP.

Les routeurs doivent ensuite négocier le rôle entre eux, pour ce faire, ils envoient des messages multicast Hello, puis l'un prend le rôle en « Active » l'autre en « Standby »

Ce sera le routeur actif qui répondra au requêtes ARP, en donnant la VIP.

Une adresse MAC virtuel est généré pour les adresse IP virtuel.

Si le routeur actif redevient fonctionnel il ne redeviendra pas automatiquement le routeur actif. Il deviendra le routeur en Standby.

Il est possible de configurer le preemption pour que l'ancien routeur ait à nouveau le rôle d'actif par défaut.

Voyons à présent comment fonctionne le HSRP (Hot Standby Router Protocol)

Ce protocole est propriétaire de Cisco, un routeur actif et en Standby sont utilisés. Il existe deux versions : 1 et 2, la version 2 ajoute le support à l'IPv6 et augmente le nombre de groupe qui peuvent être configurés.

Ce protocole utilise les adresses en multicast IPv4 en V1 : 224.0.0.2 et en V2 : 224.0.0.102

Et les adresses MAC virtuelle en V1 : 0000.0c07.acXX (XX = HSRP numéro de groupe)

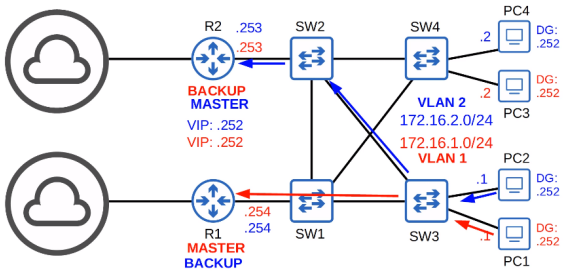
et en V2 : 0000.0c9f.fXXX (XXX = le numéro de groupe HSRP)

Dans le cas ou il y a plusieurs sous réseau, Vlan on peut configurer un routeur actif différent dans chaque sous réseau pour faire basculer le réseau

Le VRRP (Virtual Router Rdundancy Protocol) est un protocole open standard, sur ce protocole un maître et une restauration sont sélectionnés l'adresse de multicast IPv4 est 224.0.0.18 l'adresse MAC virtuel est : 0000.5e00.01XX (XX est le groupe de numéro VRRP)

Dans une situation de plusieurs sous réseau/Vlans on peut configurer un routeur maître différent de chaque sous réseau/Vlan pour load balance.

Voici le même schéma qu'auparavant mais avec VRRP :

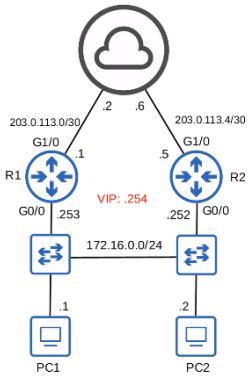


Pour finir le protocole GLBP (Gateway Load Balancing Protocol) est propriétaire Cisco il peut balancer le trafique sur plusieurs routeurs sur un seul sous réseau. Un AVG (Active Virtual Gateway) est sélectionné. Il peut y avoir jusqu'à 4 AVF (Active Virtual Forwarders) qui sont assignés par le AVG (Le AVG peut être un AVF aussi) Chaque AVF fonctionne comme la passerelle par défaut pour une portion d'hôtes du sous réseau. L'adresse Ipv4 de multicast est : 224.0.0.102

FHRP	Terminologie	Multicast IP	Virtual MAC	Cisco Priopriétaire
HSRP	Actif/Standard	V1 : 224.0.0.2 V2 : 224.0.0.102	V1 : 0000.0c07.acXX V2 : 0000.0c9f.fXXX	Oui
VRRP	Master/Backup	224.0.0.18	0000.5e00.01XX	Non
GLBP	AVG/AVF	224.0.0.102	0007.b400.XXYY	Oui

Voyons à présent comment configurer ces protocoles.

Commençons par HSRP :



Sur ce diagramme nous allons commencer par configurer R1

```
R1(config)#interface g0/0
R1(config-if)#standby
```

Il est possible de passer au protocole version avec cette commande :

```
R1(config-if)#standby version 2
```

On configure l'IP virtuel avec cette commande :

```
R1(config-if)#standby 1 ip 172.16.0.254
R1(config-if)#standby 1 priority 200
R1(config-if)#standby 1 preempt
```

Le routeur actif est déterminé dans cette ordre :

1 – La plus haute priorité (Par défaut 100)

2 – La plus haute adresse IP

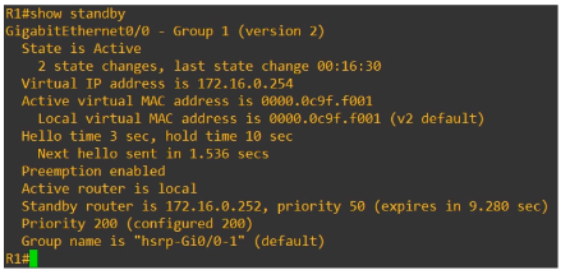
La commande « preempt » permet de configurer un routeur pour qu'il reste actif même après qu'il ait été dysfonctionnel il redeviendra actif automatiquement.

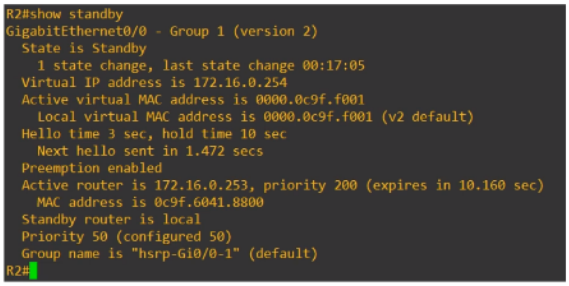
Configurons à présent le routeur 2 :

```
R2(config-if)#standby version 2
R2(config-if)#standby 1 ip 172.16.0.254
R2(config-if)#standby 1 priority 50
R2(config-if)#standby 1 preempt
```

Les versions HSRP version 1 et version 2 ne sont pas compatibles. Si R1 utilise la version 2, R2 devra utiliser la version 2 aussi.

Voici une capture d'écran de la commande « show standby » pour chaque routeur :





Cours 30 : TCP & UDP

Dans ce cours nous allons parler du fonctionnement de TCP&UDP ainsi que de leurs différences.

Nous commencerons par voir comment fonctionne la couche 4 du modèle OSI, puis nous verrons rapidement TCP (Transmission Control Protocol), puis UDP (User Datagram Protocol) en dernier temps nous comparerons les deux.

Commençons donc par rappeler quelques fonctions de la couche 4 qui correspond à la couche Transport du modèle OSI.

La couche 4 permet le transfert de données entre les hôtes. Il encapsule les données avec une entête de couche 4 puis utilise les services des couches les plus basses (les couches 3, 2 et 1) pour transférer le message intact jusqu'à l'hôte voulue. Les hôtes eux mêmes ne font pas attention aux détails des sous couche du réseau, le transfert des données est transparent pour eux.

Une autre fonction de la couche 4 est qu'il fournit des services variés aux applications, ces services peuvent être la sûreté de transfert des données. Un autre service est la réparation des erreurs, dans le cas ou une erreur survient le message est redistribué. Un autre service est le séquençage de donnée, dans le cas ou les données transmises arrive en dehors des délais, il y a une sûreté pour que les hôtes puissent séquencer les données dans le bonne ordre.

Un service supplémentaire est le contrôle de flux, qui permet le contrôle de la vitesse de transfert des données en s'assurant que l'hôte ne reçoit pas de données en excès.

Les services énuméré ci dessus sont fournit par TCP et non pas UDP.

La couche 4 permet également une chose importante qui est de pouvoir fournir les numéros de ports (On ne parle pas ici des ports physique d'une interface) Il identifie ainsi la couche Application du protocole utilisé. Ceci permet également le multiplexage de session, qui sert à pouvoir délivrer plusieurs sessions en même temps sur un même ordinateur pour plusieurs services ou serveurs différents.

Les classement des ports sont désignés par l'IANA (Internet Assigned Numbers Authority)

- les ports connus ont des numéros de ports qui vont de 0 à 1023

- Les ports enregistrés ont des numéros de ports qui vont de 1024 à 49151

- Les ports éphémère/privée/dynamique ont des numéros de ports allant de 49152 à 65535

Maintenant qu'à été expliqué le fonctionnement de la couche 4, expliquons comment fonctionne TCP (Transmission Control Protocol)

TCP est orienté pour pouvoir établir une connexion, avant même d'envoyer les données à l'hôte en question, les deux hôtes vont établir une connexion. Une fois la connexion établie, les données commencent l'échange de données.

TCP fournit une communication fiable, les destination des hôtes sont assurés d'avoir reçu chaque segment TCP. Si le segment ne reçoit pas de confirmation (ackowlegment en anglais) il renvois le segment.

TCP fournit également le séquençage qui assure que l'hôte de destination place bien les segments dans le bonne ordre même s'ils arrivent en dehors du temps.

TCP fournit le contrôle de flux qui permet à l'hôte source d'augmenter/ralentir le taux de réception de donnée.

Voici une capture écran d'une entête TCP :

TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0			NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.)																															
...	...	...																															

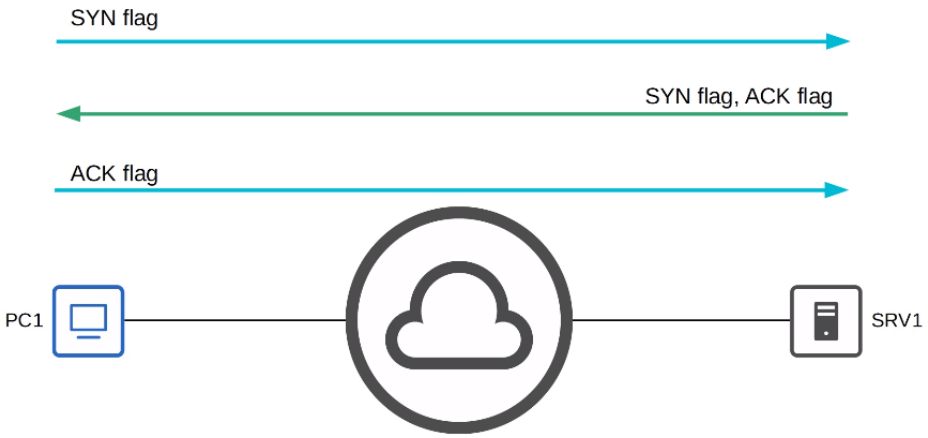
On peut voir tout d'abord les port source/destination utilisé chaque partie est d'une longueur de 16 bits.

On peut voir également la partie des flag. Les plus commun sont : ACK, SYN et FIN

Ces trois flags sont utilisés pour établir et terminer une connexion.

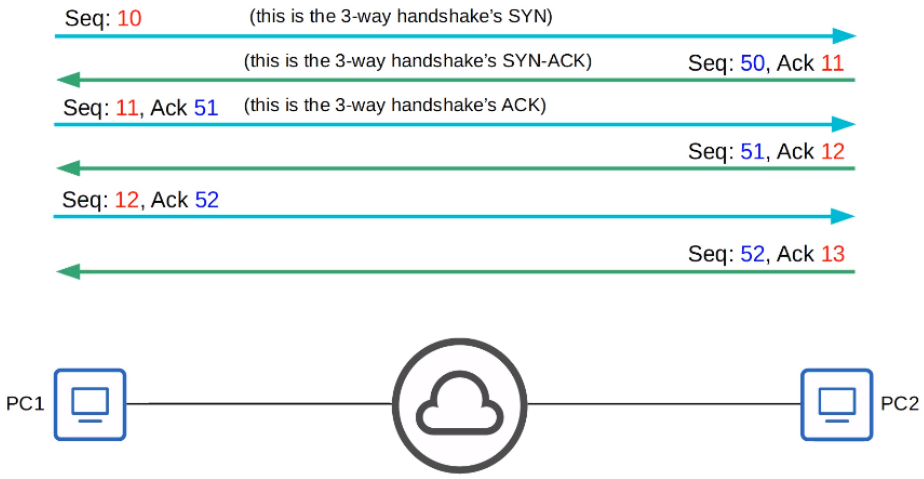
Expliquons à présent plus en détail comment la connexion est établie en utilisant le : « Three-Way Handshake » :

Sur ce schéma le PC1 veut établir une connexion avec le SRV1, pour cela il commence par envoyer un flag SYN, le SRV1 répond par un flag ACK la connexion est établie avec le ACK du PC1 envoyé.



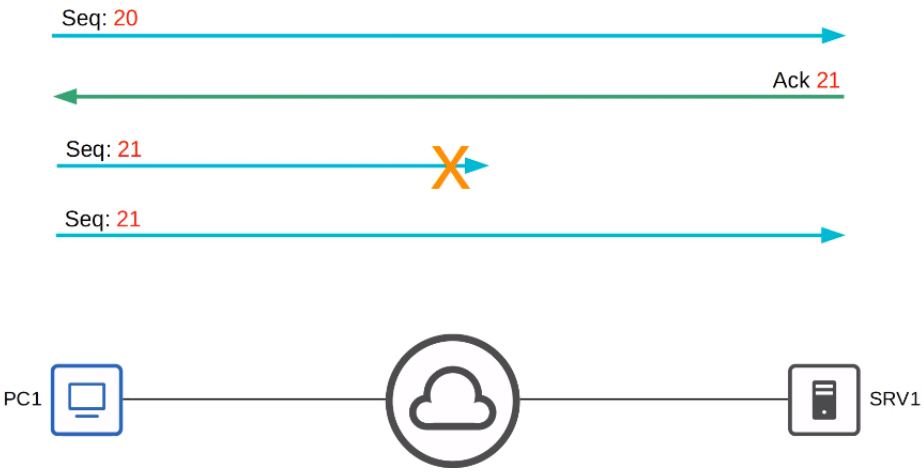
Une autre façon d'établir la connexion est par le moyen du « Four Way Handshake » ou ce sont cette fois 4 connexion nécessaires pour établir la connexion comme suit :

Voici la façon dont TCP permet le séquençage correct des données :



ce sont au départ des séquences de 10 (10 est pris pour exemple) puis au fur et à mesure des échanges le nombre de séquençage augmente de 1.

Le protocole TCP permet de s'assurer que la transmission est bien établie si ce n'est pas bien établie il renvoie le segment comme suit :



A présent que le fonctionnement de TCP est expliqué expliquons le fonctionnement de UDP.

UDP n'est pas orienté pour la connexion, les hôtes n'établissent pas de connexion avec l'hôte de destination avant d'envoyer les données. Les données sont juste envoyées.

UDP ne fournit pas une connexion fiable, Lorsque UDP est utilisé des confirmation (ou ackowledgement en Anglais) ne sont pas reçus. Si un segment est perdus UDP n'a pas de mécanisme pour le retransmettre. Les segment font de leur « mieux » pour être transmis.

UDP ne fournit pas de séquençage, il n'y a pas de nombre dans l'entête UDP. Si le segment arrive en dehors du délais, UDP n'a pas de mécanisme pour le renvoyer.

UDP ne fournit pas de contrôle de flux. UDP n'a pas de mécanisme de contrôle de flux des données comme TCP.

Voici l'entête utilisé pour UDP, image tiré de Wikipédia :

UDP datagram header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

On peut voir qu'il ne contient que quatre partis.

Comparons à présent les deux mécanismes et dans quelles cas chacun est utilisé.

TCP fournit pour sa part beaucoup de fonctionnalités par rapport à UDP mais le coût de séquençage est surélevé.

Pour les applications qui requière une communication fiable (Par exemple télécharger un fichier), TCP est privilégié. (On veut s'assurer d'avoir le fichier en entier)

Pour les applications comme voix en temps réel/Vidéo en direct, UDP est privilégié.

Il y a certaines applications qui utilisent UDP mais qui fournissent de la fiabilité dans la connexion à l'application.

Certaine application utilisent TCP&UDP cela peut dépendre de la situation.

TCP	UDP
Orienté pour établir une connexion	Orienté transfert de donnée
Fiable	Non fiable
Séquençage	Pas de séquençage
Contrôle de flux	Pas de contrôle de flux
Utiliser pour télécharger etc..	Utilisé pour le direct (Streaming, VoIP etc..)

Voici une liste des numéro de ports protocoles les plus utilisés :

TCP

- FTP data (20)

- FTP control (21)

- SSH (22)

- Telnet (23)

- SMTP (25)

- HTTP (80)

- POP3 (110)

- HTTPS (443)

UDP

- DHCP server (67)

- DHCP client (68)

- TFTP (69)

- SNMP agent (161)

- SNMP manager (162)

- Syslog (514)

TCP&UDP

- DNS (53)

Cours 31 : IPv6 Partie 1

Dans ce cours nous allons apprendre le fonctionnement d'IPV6 qui est le futur remplaçant d'IPV4.

Nous commencerons par revoir les bases de l'hexadécimal, nous verrons ensuite pourquoi utilise t-on IPV6 et les base d'IPV6, en dernier temps nous verrons comment configurer les adresses IPV6.

Pourquoi n'a t on pas appelé IPV6 : IPV5 puisque c'est le chiffre qui vient juste après IPV4 ?

Internet Stream Protocol a été développé en 1970 mais n'a jamais été montré au publique, il ne s'appelle pas IPV5 mais sa valeur est de 5 dans la version de l'entête IP. IPV4 utilise la valeur 4.

Pour éviter la confusion il a été appelé IPV6 et utilise la valeur 6 dans la version de l'entête IP

Il existe plusieurs systèmes de comptage :

- le Binaire ne comportant que deux chiffres le 0 et le 1

- le Décimal comportant 10 chiffres (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

- l'Hexadécimal comportant 16 chiffres (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

Decimal	Binaire	Hexadécimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

faisons à présent quelques exercices de conversion de Binaire à Hexadécimal :

11011011

pour se faire on divise le chiffre par deux groupes de 4 chiffres :

1101 et 1011 puis on convertit les deux séparément se qui fait : 1101 = 13 et 1011 = 11

à présent que l'on a les deux groupes en décimal on les convertit en Hexadécimal, se qui fait

13 = D et 11 = B le résultat est donc DB.

Pour 00101111 on fais :

0010 = 2 et 1111 = 15

2 = 2 et 15 = F

2F

Pour 10000001 on fais :

1000 = 8 0001 = 1

8 = 8 et 1 = 1

81

Comment fais t on pour convertir de d'hexadécimal à binaire ?

On fais inverse le processus en faisant par exemple :

EC

E = 14 et C = 12

14 = 1110 et 12 = 1100

11101100

Un autre exemple :

2B

2 = 2 et B = 11

2 = 0010 et 11 = 1011

00101011

dernier exemple :

D7

D = 13 et 7 = 7

13 = 1101 et 7 = 0111

11010111

Pour quoi utiliser IPV6 ?

La raison est qu'il n'existe plus assez d'adresse IPV4, c'est pour cela que l'on a créé IPV6

il y a en tout 4,294,967,296 (2^{32}) adresses IPV4 disponible. Cela peut sembler beaucoup mais dans le monde moderne dans lequel l'on vit cela est vite comblé.

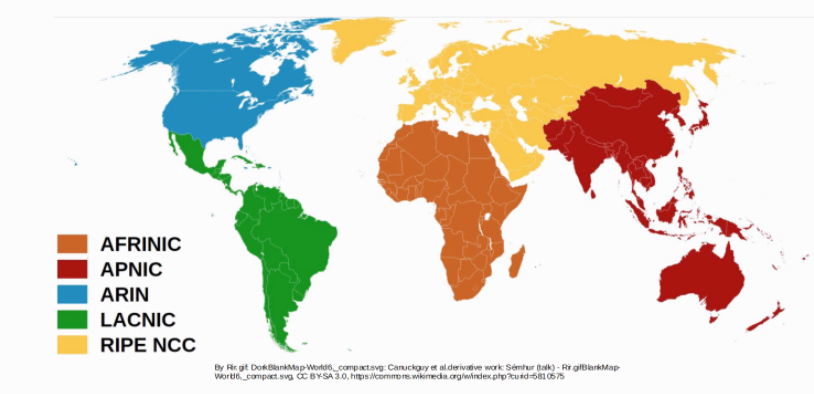
Lorsque IPV4 a été créé il y a 30, le créateur n'avait pas pensé que Internet serait aussi répandu.

Le VLSM les adresses IP privées et le NAT sont utilisés pour conserver l'espace d'adresse IPV4 disponible.

Ces techniques sont utiles mais seulement pour le court termes afin de garder les adresses disponibles, la solution à long terme est de faire la transition vers l'IPV6.

Les adresses IPV4 sont régulés par l'IANA (Internet Assigned Numbers Authority)

IANA distribue les adresses IPV4 vers des RIR (Regional Internet Registries) variés, qui eux même les assignent à des entreprises. Voici une carte des RIR :



Parlons à présent du fonctionnement d'IPV6. Une adresse IPV6 est composé de 128 bits

Pour chaque bit dans IPV6 le nombre d'adresse disponible est à chaque fois multiplié par deux contrairement à l'IPV4 ou le nombre d'adresse disponible est multipliés par quatre.

Cela signifie qu'il y a en tout 340,282,366,920,938,463,374,607,431,768,211,456 adresses IPV6 disponible. A la différence de l'IPV4 les adresses ne sont pas notés en décimal mais en Hexadécimal. Une adresse IPV6 est divisé en 8 groupes de 4 chiffres séparé par des deux points le masque de sous réseau est inscrit en utilisant la notation avec le /

Il y a plusieurs moyens de réduire une adresse IPV6 pour qu'elle soit plus clairement lisible.

Voici quelques règles général :

- Les 0 en première position du groupe de chiffres peuvent être retirés.

Par exemple pour l'adresse : 2001 :0DB8 :000A :001B :20A1 :0020 :0080 :34BD

l'adresse sera égal à 2001 :DB8 :A :1B :20A1 :80 :34BD

- Les groupes de quatre 0 peuvent être remplacés par des double deux points.

Par exemple pour l'adresse : 2001 :0DB :0000 :0000 :0000 :0080 :34BD

l'adresse sera égal à : 2001 :0DB8 ::0080 :34BD

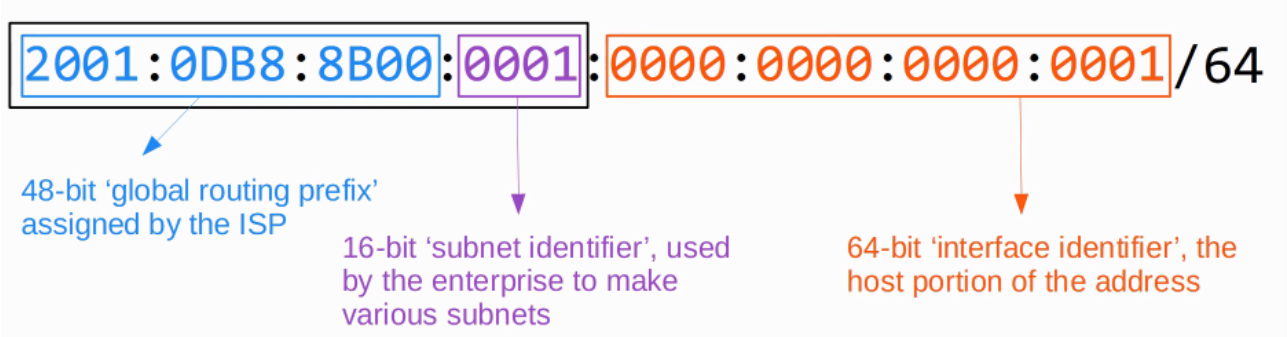
on peut combiner avec la méthode précédente se qui fera : 2001 :DB8 :80 :34BD

Des groupes de quatres 0 ne peuvent être abrégé qu'une seule fois car sinon l'on ne pourrait pas savoir combien de groupe de 0 il y a pour chaque abréviation.

Habituellement une entreprise fais la requête d'une adresse IPV6 depuis son ISP et reçoit un bloque de 48 bit. Le sous réseau est lui d'une longueur total de 64 bits.

Cela signifie qu'une entreprise a 16bits pour dévellopper un sous réseau.

Les 64 bits peuvent être utilisés pour les hôtes.



Voici un exemple d'une adresse IPV6 avec les différentes parties :

En bleue les 48 bits désignés par l'ISP avec en violet utilisé par les entreprise pour faire des sous réseaux variés. La partie orange est utilisé pour identifier la partie Hôte de l'adresse.

Essayons à présent de trouvant la partie du préfixe utilisé par l'entreprise.

Avec l'adresse :

2001 :0DB8 :8B00 :0001 :FB89 :017B :0020 :0011**/93**

On compte jusqu'au 92ème bit qui correspond au 6ème groupe qui est 017B

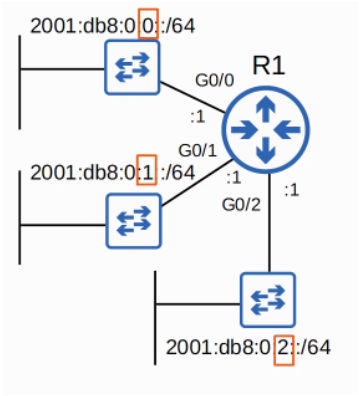
car 16+16+16+16+16=80

le 93ème bit est compris jusqu'au chiffre 7 et est le premier bit du chiffre « B » ce qui donne 11 en décimal et qui est égal à 1011 en binaire seulement le premier bit 1 fait partie du préfixe le reste peut être convertit en 0 se qui fait 1000 ceci signifie que le « B » sera égal à 8.

si l'on réécrit le préfixe réseau utilisé cela sera : 2001 :DB8 :8B00 :1 :FB89 :128 ::/93

Voyons quelles sont les commandes basique pour configurer une adresse IPV6 sur une configuration réseau.

Le réseau utilisé est le suivant :



Les commandes utilisés pour configurer le routeur avec les adresses demandés sont les suivantes :

```
R1(config)#ipv6 unicast-routing
R1(config)#int g0/0
R1(config-if)#ipv6 address 2001:db8:0:0::1/64
R1(config-if)#no shutdown
R1(config-if)#int g0/1
R1(config-if)#ipv6 address 2001:db8:0:1::1/64
R1(config-if)#no shutdown
R1(config-if)#int g0/2
R1(config-if)#ipv6 address 2001:0db8:0000:0002:0000:0000:0000:0001/64
R1(config-if)#no shutdown
```

Ici les adresses ont été configuré de 3 manière différentes avec des abréviations différentes.

Cours 32 : IPV6 Partie 2

Dans ce cours nous allons continuer le cours précédent à propos du fonctionnement d'IPV6.

Commençons par décrire ce qu'est « EUI-64 »

C'est l'acronyme de « Extended Unique Identifier », il s'agit d'une méthode pour convertir une adresse MAC (48 bits) en une interface avec un ID de 64 bit.

Cette identifiant d'interface peut devenir la partie hôte d'une adresse IPV6 en /64

Voici comment convertir une adresse MAC :

1. Diviser l'adresse MAC en deux :

1234 5678 90AB → 1234 56 | 78 90AB

2. Insérer FFFE au milieu

1234 56FF FE78 90AB

3. Inverser le 7ème bit de l'adresse MAC

1234 56FF FE78 90AB

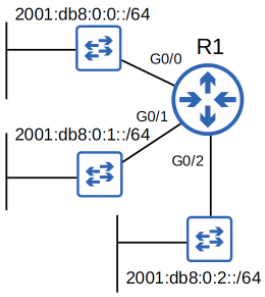
0010 → 0000

Dans l'exemple le 7ème bit est placé au deuxième chiffre qui est le 2 et on inverse le bit concerné.

Voici quelques exemples d'adresses MAC convertit en EUI-64 Interface Identifier :

MAC Address	EUI-64 Interface Identifier
782B CBAC 0867	7A2B CBFF FEAC 0867
0200 4C4F 4F50	0000 4CFF FE4F 4F50
0050 56C0 0001	0250 56FF FEC0 0001
00FF 6BA6 F456	02FF 6BFF FEA6 F456
96AB 6D6B 98AE	94AB 6DFF FE6B 98AE

A présent que l'on sait comment convertir une adresse voici comment configurer une adresse EUI 64 sur un routeur par rapport au schéma suivant :




```
R1(config)#int g0/0
R1(config-if)#ipv6 address 2001:db8::/64 eui-64
R1(config-if)#int g0/1
R1(config-if)#ipv6 address 2001:db8:0:1::/64 eui-64
R1(config-if)#no shutdown
R1(config)#int g0/2
R1(config-if)#ipv6 address 2001:db8:0:2::/64 eui-64
R1(config-if)#no shutdown
```

Expliquons pourquoi doit on inverser le 7ème bit de l'adresse MAC d'origine pour la convertir en EUI-64.

Les adresses MAC sont divisés en deux types :

1. UAA (Universally Administered Address)

- Il s'agit d'adresse assignés à un appareil uniquement par le fabriquant

2. LAA (Locally Administered Address)

- ce sont des adresses manuellement assignés par un administrateur (avec la commande :

mac-address) il n'a pas à être unique.

On peut identifier un UAA ou un LAA avec le 7ème bit d'une adresse MAC appelé le bit U/L (Universal/Local bit) :

→ le bit U/L est à 0 lorsque c'est un UAA

→ le bit U/L est à 1 lorsque c'est un LAA

Dans le contexte d'une adresse IPV6/EUI-64, la signification de U/L est inversé :

Dans le contexte d'une adresse IPV6/EUI-64 la signification de U/L est inversé :

→ Le bit U/L est à 0 lorsque l'adresse MAC d'une interface EUI-64 ID est fais à partir d'un LAA

→ Le bit U/L est à 1 lorsque l'adresse MAC d'une interface EUI-64 ID est fais à partir d'un UAA

EUI-64 n'est pas vraiment un type d'adresse IPV6 c'est une méthode qui permet de générer automatiquement une adresse IPV6 en utilisant un préfixe et une adresse MAC.

Voyons à présent quelles sont les différents types d'adresses IPV6.

- **Global Unicast** : Les adresses IPV6 Global Unicast sont publiques et peuvent être utilisés à travers internet. Il faut s'enregistrer pour les utiliser, car se sont des adresses publique et elles doivent être unique.

Ces adresses sont définis avec le bloc : 2000 ::/3

(Allant de 2000 :: à 3FFF :FFFF :FFFF :FFFF :FFFF :FFFF :FFFF :FFFF)

2001:0DB8:8B00:0001:0000:0000:0000:0001/64

En bleu les 48 bit est le 'global routing prefix' assigné par l'ISP.

En violet les 16 bit est 'identifiant de sous réseau' utilisé par l'entreprise pour varier le sous réseau.

Ces deux parties (Bleue et Violet) forment le préfixe IPV6.

En orange les 64 bit restant est l'interface identifiant ou la partie hôte de l'adresse.

- **Unique Local** : Les adresses IPV6 Unique Local sont des adresses privés qui ne peuvent pas être utilisés sur Internet. Il n'est pas nécessaire de s'enregistrer pour les avoir et qu'elles soient unique.

Ces adresses sont définis avec le bloc : FC00 ::/7

(Allant de FC00 :: à FDFF :FFFF :FFFF :FFFF :FFFF :FFFF :FFFF :FFFF)

Un changement à fais que le 8ème bit soit être définis à 1, donc les premiers chiffres doivent être « FD ».

Voici l'exemple d'une adresse Unique Local :

FD45:93AC:8A8F:0001:0000:0000:0000:0001/64

En marron est indiqué l'adresse Unique Local.

En bleu les 40 bits sont le 'global ID' généré au hazard. En violet les 16bit sont l'identifiant de sous réseau utilisé par l'entreprise pour varier le sous réseau. Ces trois parties (Marron, Bleu et Violet) forment le préfixe IPV6.

En orange les 64 bit restant est l'interface identifiant ou la partie hôte de l'adresse.

- **Link Local :** Les adresses Link-Local sont automatiquement générés sur une interface IPV6 activé. Il faut utiliser la commande : *R1(config-if)#ipv6 enable* sur une interface pour activer l'IPV6.

Est utilisé le bloc d'adresse FE80 ::/10

(Allant de FE80 :: à FEBF :FFFF :FFFF :FFFF :FFFF :FFFF :FFFF :FFFF)

Le standard allant de 54 bits après le FE80/10 doit être à 0, donc on ne verra jamais d'adresse Link Local commençant par FE9, FEA, FEB, mais seulement par FE8.

L'interface ID est généré en utilisant les règles de EUI-64

Link-Local signifie que ces adresses sont utilisés pour la communication sur un seul lien (sous réseau) Les routeurs ne vont pas router les paquets avec une adresse IPV6 de destination link-local.

Link-Local a plusieurs utilisations :

→ appairage de protocoles (OSPFv3 utilise les adresses link-local)

→ Détection des sauts suivants (Next Hop) pour le routage statique

→ Neighbor Discovery Protocol (NDP, remplace ARP pour l'IPV6) utilise les adresses Link-Local pour fonctionner

- **Multicast Adresses** : Pour rappel une adresse Unicast est une adresse allant d'une source vers une autre destination.

Les adresses Broadcast sont des adresses allant d'une source vers toutes les destinations possible.

Les adresses Multicast sont des adresses allant d'une source vers plusieurs destinations précise.

IPV6 utilise le classement FF00 ::/8 pour le multicast

(Allant de FF00 :: à FFFF :FFFF :FFFF :FFFF :FFFF :FFFF :FFFF :FFFF)

IPV6 n'utilise pas le Broadcast (il n'y a pas de broadcast dans IPV6) mais il existe des adresses qui fonctionnent comme Broadcast.

But	Adresse IPV6	Adresse IPV4
Tous les Hôtes (fonctionne comme un Broadcast)	FF02 ::1	224.0.0.1
Tous les routeurs	FF02 ::2	224.0.0.2
Tous les routeurs OSPF	FF02 ::5	224.0.0.5
Tous les OSPF DRs/B-DRs	FF02 ::6	224.0.0.6
Tous les routeurs RIP	FF02 ::9	224.0.0.9
Tous les routeurs EIGRP	FF02 ::A	224.0.0.10

IPV6 définit plusieurs 'scopes' qui indique combien le paquet doit être retransmis.

Les adresses dans le tableau précédent utilisent le « scope » Link-Local (FF02) qui reste dans le sous réseau local.

Il y a plusieurs multicast IPV6 :

→ Interface-local (FF01) : Le paquet ne quitte pas l'appareil local. Peut être utilisé pour envoyer le trafique vers un servive intégré à l'appareil lui même.

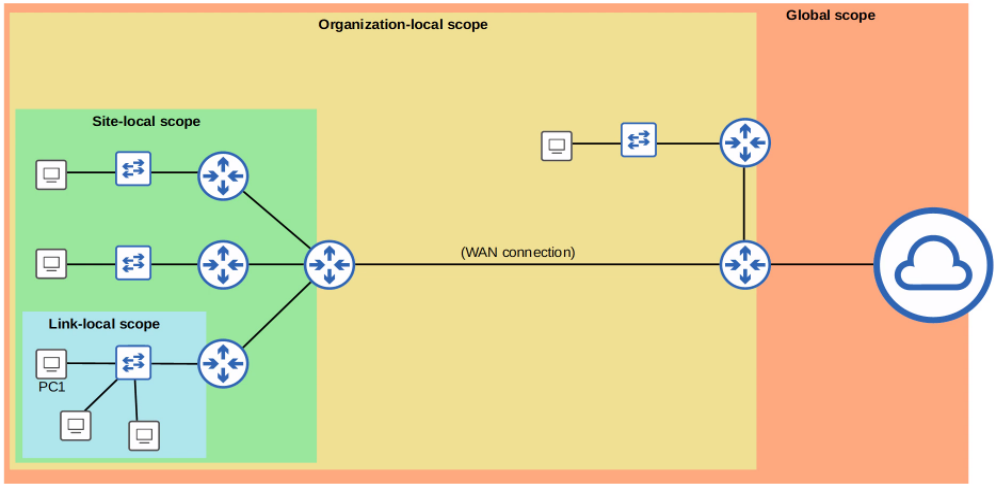
→ Link-Local (FF02) : Le paquet reste dans le sous réseau local. Les routeurs ne peuvent pas router le paquet entre les sous réseaux.

→ Site-local (FF05) : Le paquet peut être transmis par les routeurs. Doit être limité à une seul localisation physique (et non pas retransmis à travers un WAN)

→ Organization-local (FF08) : Le scope est plus élargie par rapport au site-local (une compagnie entière)

→ Global (FF0E) : Sans frontières. Possibilité d'être routé à travers Internet

Voici un schéma pour mieux comprendre l'utilité de chaque scope :



- **Anycast Adresses** : Anycast est une fonction de IPV6, se sont des adresses allant d'une source vers une destination. Plusieurs routeurs sont configurés avec la même adresse IPV6.

→ Ils utilisent le protocole de routage pour avertir de leurs adresses.

→ Lorsqu'un hôte envoie un paquet à une adresse de destination, les routeurs vont retransmettre vers le routeur sur lequel est configuré cette même adresse (basé sur des métriques de routage)

Il n'existe pas de classement spécifique pour n'importe quelle adresse anycast. Utiliser une adresse Unicast regular (global unicast, unique local) et spécifier qu'il a une adresse anycast :

```
R1(config-if)#ipv6 address 2001:db8:1:1::99/128 anycast
```

Il existe d'autres adresses IPV6

- :: = Les adresse IPV6 non spécifiés

→ peuvent être utilisés lorsqu'un appareil ne connais pas ses adresses IPV6

→ le routage IPV6 par défaut est configuré à ::/0

→ équivalent IPV4 de 0.0.0.0

- ::1 = adresse de loopback

→ utilisé pour tester la pile protocole d'un appareil local

→ les messages envoyés vers cette destination sont procédés dans le réseau local, mais ne sont pas envoyés vers d'autres appareils.

→ équivalent IPV4 de 127.0.0.0/8

Cours 33 : Ipv6 (Partie 3)

Dans cette vidéo nous allons continuer le cours sur l'IPV6.

Nous allons revoir une chose que nous avons lors du du précédent cours, nous verrons ensuite comment est composé l'entête IPV6, nous explorerons différents protocoles, les protocoles Neighbor Discovery Protocol (NDP) et SLAAC, nous finirons par apprendre comment configurer un routage statique en IPV6.

Nous allons revoir rapidement comment est composé l'IPV6, tout d'abord commençons par définir ce qu'est qu'une RFC.

Une RFC (Request for Comment) est une publication de l'ISOC (Internet Society) et de ses organisations associés comme l'IETF (Internet Enginneering Task Force), pour définir les normes officiels des protocoles, procédures et spécifications Internet etc.…

Donc si l'on veut en savoir sur le protocol OSPF par exemple il suffit de nous rendre sur l'RFC qui documente le protocole OSPF.

Le titre de l'RFC 5952 est 'A recommendation for IPV6 Address Text Representation' en Français 'Une recommendation pour les textes représentatifs des adresses IPV6'

Avant la publication de cette RFC, les représentation des adresses IPV6 était plus flexibles.

- On pouvait supprimer des 0 premiers

- On pouvait remplacer tous les 0 par des ::

- On pouvait utiliser les majuscules 0xA, B, C, D, E, F ou minuscules 0xa, b, c, d, e, f

La RFC 5952 a permis une standardisation de la représentation des adresses IPV6.

Sa publication a mis en place les changements suivants :

- Les 0 placés en début doivent obligatioment être supprimés

Exemple : 2001 :0db8 :0000 :0001 :0f2a :4fff :fea3 :00b1 → 2001 :db8 :0 :1 :f2a :4fff :fea3 :b1

- Les :: doivent obligatoirement être utilisés lorsque plusieurs groupes de quatres 0 sont placés de manière consécutives.

Exemple : 2001 :0000 :0000 :0000 :0f2a :0000 :0000 :00b1 → 2001 ::f2a :0 :0 :b1

(Lorsqu'il n'y a qu'un groupe de quatres 0 on n'utilise pas les ';;')

- Lorsqu'il y deux fois des groupes de quatres 0 séparés et que l'on peut utiliser deux fois les ' ::' on ne l'utilisera que pour celui le plus à gauche.

Exemple : 2001 :0db8 :0000 :0f2a :0000 :0000 :0000 :00b1 → 2001 :db8 ::f2a :0 :0 :b1

- Les caractère hexadécimale 'a','b','c','d','e' et 'f' doivent obligatoirement être écris en minuscules et non pas en majuscules.

Il n'est pas grave si les caractères sont quelques fois écris en majuscules car aussi sur les routeurs il sont en majuscules alors que ça n'est pas réglementaire.

Voyons à présent de quoi se compose l'entête IPV6 :

Fixed header format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Traffic Class								Flow Label																			
4	32	Payload Length																Next Header								Hop Limit							
8	64	Source Address																															
12	96																																
16	128																																
20	160																																
24	192	Destination Address																															
28	224																																
32	256																																
36	288																																

Voici en comparaison l'entête IPV4 que l'on a vu auparavant dans les précédent cours :

IPv4 header format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

L'entête IPV6 semble plus simple, les choses qui font que l'ipv6 est plus simple est le fais que ce soit un format d'entête fixe. L'entête IPV4 a un format variable de 12 à 60bits tandis que l'IPV6 a une entête fixe de 40bits c'est pour cela qu'il y a une partie pour la longueur du Payload et non pas pour la partie de l'entête comme dans IPV4.

Le processus de l'IPV6 est plus simple pour les routeurs donc les performance sont en conséquences améliorés.

Nous allons à présent détailler chaque partie utilisé :

- Version : d'une longueur de 4bits, il indique la version de l'IP utilisé. La valeur fixé est de 6 (0b0110) pour indiquer IP version 6

- Traffic class : d'une longueur de 8bits, il est utilisé pour la QoS (Quality of Service), pour indiquer la priorité du trafique. Par exemple le trafique IP d'un téléphone, un appel vidéo, etc. il aura une valeur de Traffic Class qui indique la priorité à travers les autres trafiques du réseau.

- Flow Label : d'une longueur de 20bits, il est utilisé pour identifier les flux spécifique de communication (la communication entre une source spécifique et sa destination)

- Longueur du Payload : Il indique la longueur du Payload (le segment de couche 4 encapsulé) en Octets. La longueur de de l'entête IPV6 n'est pas inclus car elle est toujours de 40 octets.

- Next Header : d'une longueur de 8 bits, il permet d'identifier le type de l'entête suivante (l'entête du segment encapsulé), par exemple TCP ou UDP.

Il a la même fonction que la partie « Protocol » de l'entête IPV4

- Hop Limit : d'une longueur de 8bits, la valeur de cette partie est soustraite par 1 à chaque fois qu'un routeur le repartage. S'il atteint 0 le paquet est perdue.

Il a la même fonction que la partie « TTL » de l'entête IPV4.

- Source/Destination : d'une longueur de 128 bits chacun (Source : 128, Destination : 128)

Ces parties contiennent l'adresse IPV6 pour la source du paquet ainsi que sa destination.

Un nœud IPV6 sollicité pour une adresse multicast est calculé à partir d'une adresse unicast.

L'adresse commence par un préfixe fixe par exemple : ff02 :0000 :0000 :0000 :0000 :0001 :ff

puis par l'ajout des 6 derniers chiffres hexadécimal de l'adresse Unicast

Prenons pour exemple l'adresse Unicast : 2001 :0db8 :0000 :0001 :0f2a :4fff :fea3 :00b1

Pour générer un nœud IPV6 les derniers chiffres de l'adresse Unicast vont être utilisé ici ce sera : a3 :00b1

l'adresse final utilisé sera : ff02 ::1 :ffa3 :b1

Pour une adresse Unicast : 2001 :0db8 :0000 :0001 :0489 :4eda :073a :12b8

les 6 derniers chiffres de cette adresse sont **3a :12b8**

L'adresse final utilisé sera donc : ff02 ::1 :ff3a :12b8

```
R1#sh ipv6 int g0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::EF8:22FF:FE36:8500
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8::EF8:22FF:FE36:8500, subnet is 2001:DB8::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF36:8500
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
```

On peut voir que dans la partie encadré en rouge le routeur a joins les deux groupes multicast FF02 ::1 et FF02 ::2

il est à noter que le routeur a également joins l'adresse FF02 ::1 :FF36 :8500 ceci est le nœud sollicité pour l'adresse Multicast, on remarque que les 6 derniers chiffres sont les mêmes que ceux de l'adresse IPV6 utilisé.

Nous allons à présent expliquer comment fonctionne le protocole NDP (Neighbor Discovery Protocol) et son utilité pour IPV6.

Le protocole NDP (Neighbor Discovery Protocol) à des fonctions très variés, et l'unes de ces fonctions est de remplacer ARP, qui n'est plus utilisé dans IPV6.

La fonction ARP de NDP utilise ICMPv6 et les nœud solicités de l'adresse multicast pour apprendre l'adresse MAC des autres hôtes du réseau. Pour rappel le protocole ARP utilise des requête de message en Broadcast. La méthode utilisé pour IPV6 est plus efficace car ce sont les hôtes d'adresses spécifiques qui sont utilisés contrairement à des messages envoyés en Broadcast pour tous les hôtes.

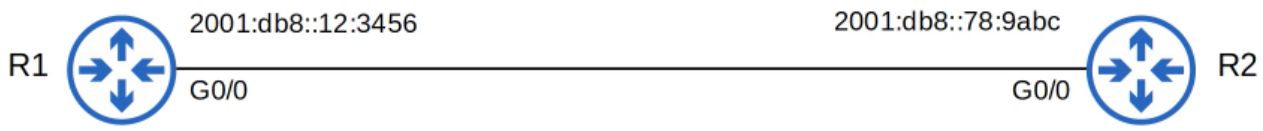
Deux types de messages sont utilisés :

1) Neighbor Solicitation (NS) = ICMPv6 Type 135

2) Neighbor Advertisement (NA) = ICMPv6 Type 136

Voici le fonctionnement basique d'un message Neighbor Solicitation (NS) l'équivalent NDP d'une requête ARP

Nous utiliserons la topologie suivante :



Disons que R1 veut faire un ping de l'adresse R2, pour cela il va d'abord envoyer une requête pour connaître l'adresse MAC

Pour comprendre comment cela est différent d'une requête ARP comparons les différentes adresses utilisés dans une capture Wireshark

```
> Frame 6: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
> Ethernet II, Src: ca:01:09:6d:00:08 (ca:01:09:6d:00:08), Dst: IPv6mcast_ff:78:9a:bc (33:33:ff:78:9a:bc)
> Internet Protocol Version 6, Src: 2001:db8::12:3456, Dst: ff02::1:ff78:9abc
> Internet Control Message Protocol v6
```

On peut voir sur cette capture :

- En rouge : l'adresse IP source : R1 G0/0

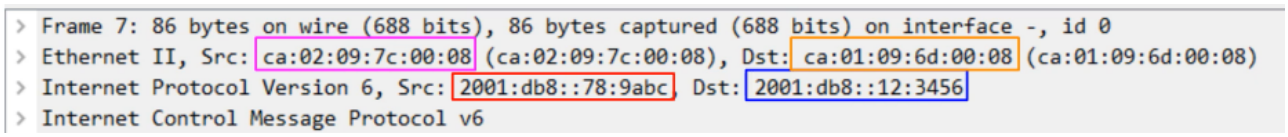
- En bleu : l'adresse IP de destination avec l'adresse du nœud sollicité pour l'adresse Multicast

- En rose : L'adresse source MAC

- En jaune : l'adresse MAC Multicast basé sur l'adresse du nœud sollicité de R2

Voici à présent le fonctionnement basique d'un message Neighbor Advertisement (NA)

Pour répondre à la requête de R1 lui demandant son adresse MAC, R2 va répondre basiquement en lui envoyant son adresse MACvoyons donc les différentes adresses utilisés dans le message dans Wireshark :



- En rouge : l'adresse IP source (ici R2)

- En bleu : l'adresse IP de destination (ici R1)

- En rose : l'adresse MAC source

- En jaune : l'adresse MAC de destination

Voyons comment sont utilisés les Neighbor Table pour IPV6 dans notre exemple, il est possible de l'afficher avec la commande : *show ipv6 neighbor*

```
R1#show ipv6 neighbor
IPv6 Address                              Age Link-layer Addr State Interface
FE80::C802:9FF:FE7C:8                       0 ca02.097c.0008  REACH Gi0/0
2001:DB8::78:9ABC                           0 ca02.097c.0008  REACH Gi0/0
```

```
R2#show ipv6 neighbor
IPv6 Address                              Age Link-layer Addr State Interface
FE80::C801:9FF:FE6D:8                       0 ca01.096d.0008  REACH Gi0/0
2001:DB8::12:3456                           0 ca01.096d.0008  REACH Gi0/0
```

On peut voir d'abord la colonne des adresses IPV6 sur laquelle est contenu l'age qui indique depuis combien de temps en minutes les adresses IPv6 ont été identifiés, le Link-Layer Address montre les adresses MAC, la colonne Interface montre sur quelles interface cela à été appris.

Nous allons expliquer encore une chose à propos du protocole NDP

Une autre fonction de NDP permet aux hôtes de découvrir automatiquement les routeurs sur le réseau local.

Deux messages sont utilisés pour ce processus :

1) Router Solicitation (RS) = ICMPv6 Type 133

→ ils sont envoyés à des adresses Multicast FF02 ::2 (à tous les routeurs)

→ ce message demande à tous les routeurs du réseau local de s'identifier eux même

→ ce message est envoyé lorsqu'une interface est activé et que l'hôte est connecté au réseau

2) Router Advertisement (RA) = ICMPv6 Type 134

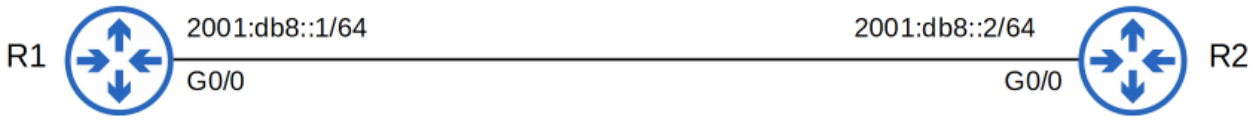
→ ces messages sont envoyés à des adresses multicast FF02 ::1 (à tous les nœud)

→ Le routeur annonce sa présence en plus d'autres informations à propos du réseau local

→ ces messages sont envoyés en réponse aux messages RS

→ ils sont aussi envoyés périodiquement, même si le routeur n'a pas reçu de RS

Nous allons utiliser la topologie suivante pour mieux comprendre :



R2 envoie automatiquement un message RS pour vérifier s'il y a un routeur sur le réseau local

R1 répond en s'identifiant et en envoyant un RA au routeur R2

Nous allons à présent expliquer comment fonctionne SLAAC.

SLAAC est l'acronyme de Stateless Address Auto-Configuration

C'est une autre manière pour configurer des adresses IPV6, en utilisant SLAAC, les hosts utilisent les messages RS/RA pour apprendre le préfixe IPV6 du réseau local (par exemple : 2001 :db8 ::/64) ce qui permet de générer automatiquement une adresse IPV6.

Lorsque l'on utilise la commande : *ipv6 address* ***prefix/prefix-length*** *eui-64* il nous faut manuellement entrer le préfixe. En utilisant la commande :

ipv6 address autoconfig il ne devient plus nécessaire d'entrer le préfixe. L'appareil utilise NDP pour apprendre le préfixe utilisé dans le réseau local.

L'appareil utilisera EUI-64 pour générer l'interface ID, ou bien il sera automatiquement généré.

Voici un exemple de la commande utilisé sur un routeur Cisco :

```
R2(config)#int g0/0
R2(config-if)#ipv6 address autoconfig
R2(config-if)#do show ipv6 interface brief
GigabitEthernet0/0     [up/up]
    FE80::EF8:22FF:FE56:A600
    2001:DB8::EF8:22FF:FE56:A600
GigabitEthernet0/1     [administratively down/down]
    unassigned
GigabitEthernet0/2     [administratively down/down]
    unassigned
GigabitEthernet0/3     [administratively down/down]
    unassigned
```

Une dernière chose que nous allons expliquer à propos de NDP est le DAD (Duplicate Address Detection) qui permet aux hôtes de vérifier si l'autres appareils sur le réseau local utilisent la même adresse IPV6.

A n'importe quelle moment ou une interface IPV6 est activé (avec la commande :

*no shutdown)*ou qu'une adresse IPV6 est configurer sur une interface (en utilisant les méthodes : manuel, SLAAC, etc.) il fait fonctionner DAD.

DAD utilise deux messages utilisés plus tôt : NS et NA

L'hôte va envoyer un NS à sa propre adresse IPV6. S'il ne reçoit pas de réponse, il sais que l'adresse est unique sur le réseau. S'il reçoit une réponse, cela signifie qu'un autre hôte sur le réseau utilise également cette même adresse.

Sur un routeur Cisco on reçoit le message suivant si la même adresse est détecté :

```
*Oct 31 11:28:48.318: %IPV6_ND-4-DUPLICATE: Duplicate address 2001:DB8::1 on GigabitEthernet0/0
```

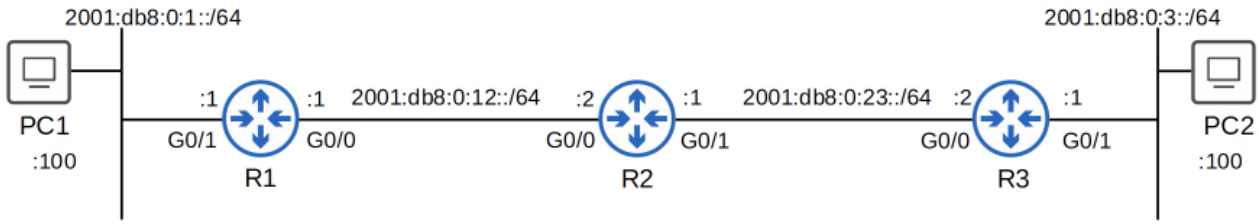
Voyons à présent comment configurer le routage statique d'une adresse IPV6.

Le routage IPV6 fonctionne de la même manière que le routage IPV4, même si les deux processus sont séparés sur le routeur et que les deux tables de routage sont aussi séparés.

Le routage IPV4 est activé par défaut, tandis que le routage IPV6 est désactivé par défaut et doit être activé avec la commande *ipv6 unicast-routing*

Si le routage IPV6 est désactivé, le routeur sera capable d'envoyer et de recevoir le trafique IPV5, mais ne pourra pas « router » le trafique IPV6 (ne partagera le trafique entre les réseau)

Pour démonter le fonctionnement d'IPV6 nous utiliserons la topologie suivante :



Voici la table de routage du routeur R1 :

```
R1#show ipv6 route
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       la - LISP alt, lr - LISP site-registrations, ld - LISP dyn-eid
       lA - LISP away, a - Application
C   2001:DB8:0:1::/64 [0/0]
     via GigabitEthernet0/1, directly connected
L   2001:DB8:0:1::1/128 [0/0]
     via GigabitEthernet0/1, receive
C   2001:DB8:0:12::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L   2001:DB8:0:12::1/128 [0/0]
     via GigabitEthernet0/0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Comme pour l'adresse IPV4 un réseau de routage est automatiquement ajouté pour chaque réseau connecté. (La lettre C pour « connected » est utilisé dans le résultat de la commande)

Aussi un routage en localhost est automatiquement ajouté pour chaque adresse configuré sur le routeur. (La lettre L pour « local » est utilisé dans le résultat de la commande)

Une autre chose à préciser est que les adresses link-local ne sont pas ajoutés à la table de routage.

Voici la commande utilisé pour le routage statique d'une adresse IPV6 :

`ipv6 route` *`destination/prefix-length {next-hop | exit-interface [next-hop]}[ad]`*

La première partie est compréhensible, il faut indiquer l'adresse IP de destination ainsi que la longueur du préfixe dans la deuxième partie entre crochet il faut indiquer soit le next-hop ou bien l'interface de sortie, si l'on indique l'interface de sortie, le next-hop devient à ce moment là facultatif. Il y a ensuite le « ad » pour Administrative Distance qui est optionnel et qui permet d'entrer un routage floating static.

Dans un routage statique « directement attaché » il ne faut préciser que l'interface de sortie.

La commande sera donc la suivante :

`ipv6 route destination/prefix-length exit-interface`

Par exemple pour R1 on utilisera cette commande :

```
R1(config)#ipv6 route 2001:db8:0:3::/64 g0/0
```

Pour un routage statique récursif, seulement le next hop est précisé.

La commande sera donc la suivante :

`ipv6 route destination/prefix-Length next-hop`

Sur R1 on utilisera la commande suivante :

```
R1(config)#ipv6 route 2001:db8:0:3::/64 2001:db8:0:12::2
```

La dernière façon de faire le routage statique est la méthode de la « spécification totale » ou l'interface de sortie et le next-hop sont précisés. La commande sera la suivante :

`ipv6 route destination/prefix-Length exit-interface next-hop`

Sur R1 on utilisera la commande suivante :

```
R1(config)#ipv6 route 2001:db8:0:3::/64 g0/0 2001:db8:0:12::2
```

Une chose à préciser avec la première méthode (directement attaché), dans IPV6 on ne peut pas utiliser un routage statique directement attaché si l'interface est une interface Ethernet, il faut que l'interface doit d'un autre type comme par exemple une interface Serial.

La commande utilisé pour R1 ne sera donc pas fonctionnel.

Voyons à présent quelques exemples pour la configuration :

Un routage en réseau :

```
R1(config)#ipv6 route 2001:db8:0:3::/64 2001:db8:0:12::2
```

Le routage d'un hôte :

```
R2(config)#ipv6 route 2001:db8:0:1::100/128 2001:db8:0:12::1
R2(config)#ipv6 route 2001:db8:0:3::100/128 2001:db8:0:23::2
```

Le routage par défaut :

```
R3(config)#ipv6 route ::/0 2001:db8:0:23::1
```

Une chose à préciser avec la commande suivante :

```
R1(config)#ipv6 route 2001:db8:0:3::/64 FE80::EF8:22FF:FEE6:D300
% Interface has to be specified for a link-local nexthop
R1(config)#
R1(config)#ipv6 route 2001:db8:0:3::/64 g0/0 FE80::EF8:22FF:FEE6:D300
R1(config)#do show ipv6 route
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       la - LISP alt, lr - LISP site-registrations, ld - LISP dyn-eid
       lA - LISP away, a - Application
C   2001:DB8:0:1::/64 [0/0]
     via GigabitEthernet0/1, directly connected
L   2001:DB8:0:1::1/128 [0/0]
     via GigabitEthernet0/1, receive
S   2001:DB8:0:3::/64 [1/0]
     via FE80::EF8:22FF:FEE6:D300, GigabitEthernet0/0
C   2001:DB8:0:12::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L   2001:DB8:0:12::1/128 [0/0]
     via GigabitEthernet0/0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

On peut voir que la commande n'a pas fonctionné au départ car le routeur ne parvient pas à préciser à quelle interface le routeur est connecté et ne peut donc pas préciser le next hop.

C'est pour cela qu'il faut préciser en spécification totale pour cette commande.

Cours 34 : Standard Access Control Lists

Dans ce cours nous allons apprendre le fonctionnement de ACL (Access Control Lists)

Nous verrons seulement la partie de configuration pour l'IPV4 et non pas l'IPV6.

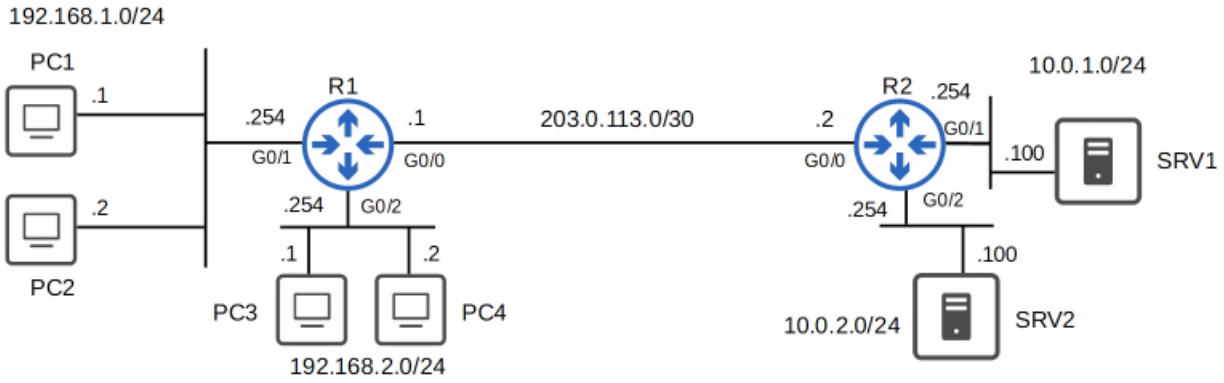
Tout d'abord nous verrons ce que sont les ACLs, la logique de fonctionnement des ACL, les différents types d'ACL, nous verrons après cela comment configurer deux types d'ACL, le Standard numbered ACLs et le Standard named ACLs.

Les ACLs (Access Control Lists) ont différentes usages, dans les prochains cours nous verrons comment les utiliser pour un usage en sécurité.

Les ACLs fonctionnent comme des filtres de paquets qui informent le routeur lorsqu'il faut autoriser ou non le passage du trafique réseau.

Les ACLs peuvent filtrer le trafique basé sur la source/destination de l'adresse IP, source/destination d'un port de couche 4, etc....

Nous utiliserons la topologie réseau suivante pour démontrer comment les ACLs fonctionnent :



A noter sur cette topologie :

- Les hôtes dans 192.168.1.0/24 peuvent accéder au réseau 10.0.1.0/24

- Les hôtes dans 192.168.2.0/24 ne peuvent pas accéder au réseau 10.0.1.0/24

- Les ACLs sont configurés en mode « global config mode » sur le routeur

Elles sont ordonnés en séquences de ACEs (Access Control Entries)

Par exemple pour l'ACL 1 :

1. Si l'IP Source est égal à 192.168.1.0/24 le trafique est autorisé

2. Si l'IP Source est égal à 192.168.2.0/24 le trafique est bloqué

3. Si l'IP Source est n'importe quoi d'autres le trafique est autorisé

Le routeur va exécuter les règles du trafique dans l'ordre, c'est pour cela qu'il est important de configurer le trafique dans l'ordre.

Configurer une ACL en mode « global config » ne rendra pas l'ACL effective automatiquement.

L'ACL doit être appliqué à une interface.

Les ACLs sont appliqués en Inbound ou Outbound (Connexion entrante ou sortante)

Imaginons que l'on veuille que :

- 192.168.1.0/24 ait accès à 10.0.1.0/24

- 192.168.2.0/24 ne puisse pas accéder à 10.0.1.0/24

Si l'on configure l'ACL sur l'interface G0/2 en Outbound (Sortant) du routeur R1, les conditions ne seront pas respectés, car le trafique sera filtré seulement pour les connexions sortantes donc lorsque 192.168.2.0/24 fera un ping vers 10.0.2.0/24 le trafique sera autorisé.

Si l'on configure l'ACL en Inbound (Entrant) le trafique sera filtré mais seulement pour le trafique en entrée donc lorsque 192.168.2.0/24 voudra faire un ping vers l'extérieur le trafique sera bloqué et 192.168.2.0/24 ne pourra communiquer qu'avec un autre PC du même réseau.

Le meilleur endroit ou placer l'ACL est l'interface G0/1 du routeur R2. Car a ce moment toutes les conditions sont respectés.

Donc une ACL est configuré en global config mode mais ils doivent être appliqués à une interface, lorsque cela est fais il faut spécifier une direction pour dire au routeur de vérifier les paquets qui entrent dans l'interface ou qui en sortent.

Les ACLs sont fais de 1 ou plusieurs ACEs.

Lorsque le routeur vérifie le paquet avec l'ACL, il les fais fonctionner les ACEs dans l'ordre, du plus haut vers le bas.

Si le paquet est compatible à unes des ACEs dans l'ACL, le routeur effectuera l'action et arrêtera de faire fonctionner l'ACL. Toutes les entrées à la suite de l'entrée compatible seront ignorés.

Par exemple s'il y a l'ACL suivante :

1. si IP source = 192.168.1.0/24 le trafique est autorisé

2. si IP source = 192.168.0.0/16 le trafique est bloqué

Si l'IP source est 192.168.1.1, le routeur prendra en compte seulement la première règle et autoriser le trafique et ne prendra pas en compte la deuxième.

Si à présent l'ACL est la suivante :

1. si IP source = 192.168.0.0/16 le trafique est bloqué

2. si IP source = 192.168.1.0/24 le trafique est autorisé

Si l'IP source est 192.168.1.1, le routeur va lire la première règle et bloquer le trafique sans prendre en compte la deuxième règle.

C'est pourquoi il est important de mettre les ACL dans l'ordre.

Une autre chose à préciser est que maximum une ACL peut être appliqué à une seule interface par direction.

Donc une ACL en Inbound (Entrante) et une ACL en Outbound (Sortante)

A présent voyons ce qu'il se passe si un paquet n'est en concordance avec aucune des ACL ?

Par exemple si une ACL est configuré avec les règles :

1. si l'IP source est 192.168.1.0/24 le trafique est autorisé

2. si l'IP source est 192.168.0.0/16 le trafique est bloqué

Si le routeur reçoit un paquet avec IP source de 10.0.0.1, cela n'est en concordance avec aucunes des règles de l'ACL. Par défaut le routeur bloquera le paquet, il ne le repartagera pas.

C'est ce que l'on appelle « implicit deny » ou « blocage implicite » en Français.

Maintenant que l'on a compris le fonctionnement des ACL, expliquons plus en détails les différents types d'ACL qui sont possibles.

Il y a deux types d'ACLs :

- Les ACL Standard : elles se basent sur l'adresse IP source *uniquement*, et sont composés de :

→ Standard Numbered ACLs

→ Standard Named ACLs

- Les ACL étendus : elles se basent sur l'adresse IP source/destination, port source/destination, etc..

et sont composés de :

→ Extended Numbered ACLs

→ Extended Named ACLs

Commençons par expliquer le fonctionnement des Standard Numbered ACLs :

Les Standard ACL se basent uniquement sur l'adresse IP source du paquet.

Les Numbered ACLs sont identifiés avec un chiffre (exemple : ACL1, ACL2, etc.…)

Différents types d'ACLs ont différents classements de nombre qui peuvent être utilisés :

→ Standard ACL peuvent utiliser 1-99 et 1300-1999

Voici un tableau qui présente le classement des ACL en nombre.

Protocol	Range
Standard IP	1-99 and 1300-1999
Extended IP	100-199 and 2000-2699
Ethernet type code	200-299
Ethernet address	700-799
Transparent bridging (protocol type)	200-299
Transparent bridging (vendor code)	700-799
Extended transparent bridging	1100-1199
DECnet and extended DECnet	300-399

Xerox Network Systems (XNS)	400-499
Extended XNS	500-599
AppleTalk	600-699
Source-route bridging (protocol type)	200-299
Source-route bridging (vendor code)	700-799
Internetwork Packet Exchange (IPX)	800-899
Extended IPX	900-999
IPX Service Advertising Protocol (SAP)	1000-1099

Voici la commande pour configurer une ACL standard numbered :

`R1(config)#access-list` *number*`{deny |permit}` *ip wilcard-mask*

Par exemple :

```
R1(config)#access-list 1 deny 1.1.1.1 0.0.0.0
```

Lorsque l'on veut ajouter une ACL avec un masque en /32 il n'est nécessaire de spécifier le masque 0.0.0.0 le routeur le configurera par défaut. Donc la commande précédente pourrait aussi être :

```
R1(config)#access-list 1 deny 1.1.1.1
```

Il existe une autre manière de configurer une ACL avec un masque en /32,en ajoute « host » entre l'adresse et l'autorisation (deny ou permit), la commande sera donc :

```
R1(config)#access-list 1 deny host 1.1.1.1
```

A présent que nous avons configuré l'ACL, il nous faut ajouter une règle pour autoriser tous les autres trafiques car sinon aucun trafique ne sera fonctionnel. On lance donc la commande :

```
R1(config)#access-list 1 permit any
```

Une autre possibilité pour autoriser tout le trafique restant serait d'utiliser la commande :

```
R1(config)#access-list 1 permit 0.0.0.0 255.255.255.255
```

Il est aussi possible d'ajouter une remarque sur une ACL par exemple avec la commande :

```
R1(config)#access-list 1 remark ## REMARQUE A AJOUTER ##
```

il est possible d'afficher les ACL du routeur en lançant unes de ces commandes :

```
R1(config)#do show access-lists
```

on peut aussi lancer cette commande :

```
R1(config)#do show ip access-lists
```

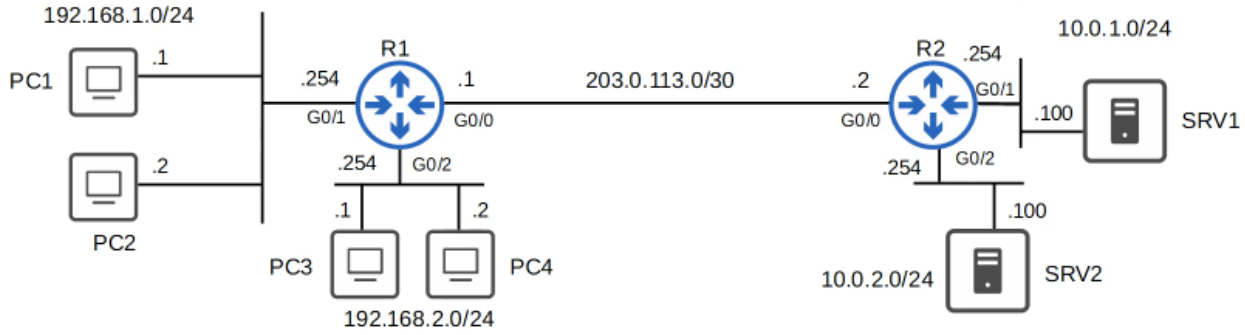
ou aussi celle ci :

```
R1(config)#do show running-config \textbar{} include access-lists
```

Pour appliquer l'ACL à une interface on lance la commande :

`R1(config-if)#ip access-group` *number* `{in |out}`

Voyons à présent comment appliquer ces commandes sur un réseau définie :



avec les règles suivantes que l'on veut établir :

- PC1 peut accéder à 192.168.2.0/24

- Les autres PCs dans 192.168.1.0/24 ne peuvent pas accéder à 192.168.2.0/24

Il faudra appliquer les commandes suivantes sur le routeur R1 :

```
R1(config)#access-list 1 permit 192.168.1.1
R1(config)#access-list 1 deny 192 .168.1.0 0.0.0.255
R1(config)#access-list 1 permit any
R1(config)#interface g0/2
R1(config-if)#ip access-group 1 out
```

Les standard ACL doivent être appliqués le plus proche de la destination possible.

Le résultat de la commande pour voir l'ACL sera le suivant :

```
R1#show access-lists
Standard IP access list 1
    10 permit 192.168.1.1
    20 deny   192.168.1.0, wildcard bits 0.0.0.255
    30 permit any
R1#
```

Si le PC1 veut faire un ping de 192.168.2.1

Il prendra donc les ACL dans l'ordre en vérifiant d'abord la première règle qui est de permettre toute entrée venant de 192.168.1.1

comme ici l'IP source est 192.168.1.1 le routeur va autoriser le trafique.

A présent si le PC2 veut faire un ping vers PC3

Le routeur commencera par vérifier la première règle qui n'est pas en concordance

puis la deuxième règle qui dis que tout trafique provenant de 192.168.1.0/24 doit être bloqué.

Ici comme l'adresse IP source est 192.168.1.2 le trafique est bloqué.

Voyons à présent le fonctionnement des Standard Named ACLs.

Les Standard ACLs ne font fonctionner le trafique que en fonction de l'adresse IP source du paquet.

Les Named ACLs sont identifiés avec des noms (par exemple : « BLOCK JOE »)

Les Standard Named ACLs sont configurés en entrant en mode 'standard named ACL config mode', et en configurant chaque entrée dans ce mode config.

On utilise la commande suivante :

`R1(config)#ip access-list standard acl-name`

Une fois dans le mode Standard Named ACL on peut lancer la commande suivante :

`R1(config-std-nacl)#[entry-number] {deny | permit} ip wilcard-mask`

Un exemple de configuration de l'ACL sur le routeur R1 serait le suivant :

```
R1(config)#ip access-list standard BLOCK_BOB
R1(config-std-nacl)#5 deny 1.1.1.1
R1(config-std-nacl)#10 permit any
R1(config-std-nacl)#remark ## CONFIGURE LE 03 AOUT 2023 ##
R1(config-std-nacl)#interface g0/0
R1(config-if)#ip access-group BLOCK_BOB in
```

Essayons de configurer le Routeur pour que les conditions suivantes soit utilisés :

- Les PC dans 192.168.1.0/24 ne peuvent pas accéder à 10.0.2.0/24

- Le PC3 ne peut pas accéder à 10.0.1.0/24

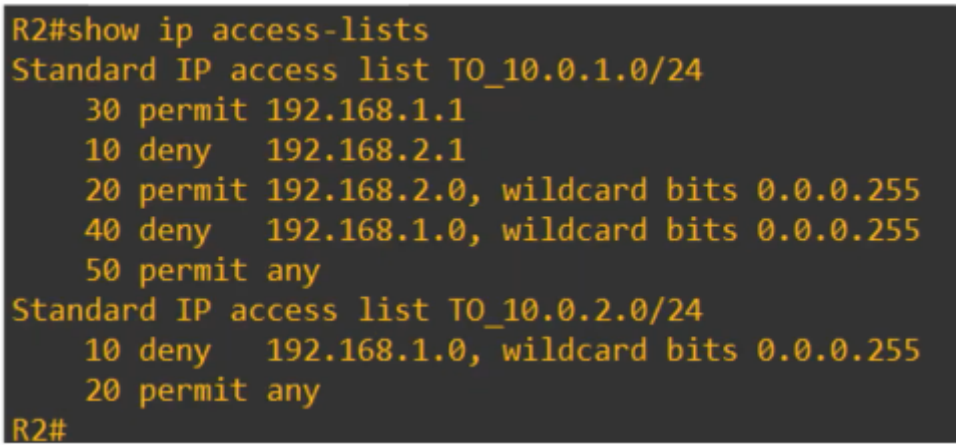
- Les autres PC dans 192.168.2.0/24 peuvent accéder à 10.0.1.0/24

- Les autres PC dans 192.168.1.0/24 ne peuvent pas accéder à 10.0.1.0/24

les commandes à utiliser pourraient être les suivantes :

```
R2(config)#ip access-list standard TO\_10.0.2.0/24
R2(config-std-nacl)#deny 192.168.1.0 0.0.0.255
R2(config-std-nacl)#permit any
R2(config-std-nacl)#interface g0/2
R2(config-if)#ip access-group TO\_10.0.2.0/24 out
R2(config-std-nacl)#deny 192.168.2.1
R2(config-std-nacl)#permit 192.168.2.0 0.0.0.255
R2(config-std-nacl)#permit any
R2(config-std-nacl)#interface g0/1
R2(config-if)#ip access-group TO\_10.0.1.0/24 out
```

Voici un résultat de la commande *`show ip access-lists`*



Une question que l'on pourrait se poser pourrait être pourquoi les ACL n'apparaissent pas dans l'ordre que celui ou l'on a lancer les commandes ?

La réponse est que le routeur réordonne les entrées en /32 d'abord, car cela permet d'améliorer le traitement de l'ACL, cela ne changera pas l'effet de l'ACL.

Cela s'applique pour les Standard Named et les Standard Numbered ACLs.

Par contre le logiciel Packet Tracer ne fera pas cela.

Cours 35 : Extended Access Control Lists

Dans ce cours nous allons continuer le cours sur les ACLs mais cette fois sur les Extented ACLs. Ce que nous avons vu à propos des ACLs dans le cours précédent est équivalent pour les Extended ACLs, la seule différence est que les les Extended ACLs peuvent ajouter plus de filtres par rapport aux ACLs Standard qui peut se baser uniquement sur l'adresse IP source d'un Paquet.

Durant ce cours nous allons donc voir comment configurer d'une autre manière les Numbered ACLs (ce qui peut s'appliquer pour les ACL standard et Extended), nous verrons également comment modifier une ACLs déjà configuré et en dernier temps nous verrons comment fonctionne les Extended Numbered et Named ACLs

Dans le cours précédent nous avons vu que les Numbered ACLs sont configurés en mode « global config »

```
R1(config)# access-list 1 deny 192.168.1.1
R1(config)# access-list 1 permit any
```

Nous avons également appris que les « Named ACLs » (ACLs només) sont configurés avec des sous commandes dans un mode de config différent.

```
R1(config)# ip access-list standard BLOCK_PC1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

Seulement aussi dans des IOS moderne il est aussi possible de configurer des numbered ACLs de la même manière que les Named ACLs comme dans cette cette exemple ci dessous configuré de la même manière que l'exemple ci dessus :

```
R1(config)# ip access-list standard 1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

Ce sont juste deux manière différentes de configurer des Numbered ACLs, si l'on lance la commande « running-config » l'ACL apparaîtra comme si elle avait été configuré en utilisant la méthode traditionnel comme on peut le voir sur les commandes suivantes

```
R1(config)#ip access-list standard ?
  <1-99>       Standard IP access-list number
  <1300-1999>  Standard IP access-list number (expanded range)
  WORD         Access-list name

R1(config)#ip access-list standard 1
R1(config-std-nacl)#deny 192.168.1.1
R1(config-std-nacl)#permit any
R1(config-std-nacl)#
R1(config-std-nacl)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 permit any
R1(config-std-nacl)#
```

Les avantages du mode de configuration Named ACL sont les suivant :

1. Il est possible de facilement supprimer une entrée dans l'ACL avec la valeur : « no » avant la commande. Voici comment cela fonctionne :

```
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.3.0, wildcard bits 0.0.0.255
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#no 30
R1(config-std-nacl)#
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    40 permit any
R1(config-std-nacl)#
```

Voici en comparaison les commandes à lancer lorsque l'on utilise la méthode Standard Numbered ACLs :

```
R1(config)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.3.0, wildcard bits 0.0.0.255
    40 permit any
R1(config)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 deny   192.168.1.2
access-list 1 deny   192.168.3.0 0.0.0.255
access-list 1 permit any
R1(config)#no access-list 1 deny 192.168.3.0 0.0.0.255
R1(config)#do show access-lists
R1(config)#do show running-config | section access-list
R1(config)#
```

On peut voir ici que l'on a utilisé la même même méthode qu'auparavant pour supprimer l'entrée, mais que cette fois lorsque l'on veut afficher les ACLs avec la commande « show acces-lists » il n'y a aucun résultat. La raison est que ce n'est pas que l'entrée qui a été supprimé mais l'ACL entière. Lorsque l'on configure ou modifie des numbered ACLs depuis le mode global config mode, on ne peut pas seulement supprimer une entrée de manière individuel, mais on peut supprimer l'ACL en entier.

Ceci était le premier avantage d'utiliser le mode Named ACL.

2. Il y a un autre avantage qui est que l'on peut insérer de nouvelles entrées entre d'autres entrées en spécifiant le numéro de séquence comme on peut le voir dans l'exemple ci dessous :

```
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#30 deny 192.168.2.0 0.0.0.255
R1(config-std-nacl)#
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.2.0, wildcard bits 0.0.0.255
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 deny   192.168.1.2
access-list 1 deny   192.168.2.0 0.0.0.255
access-list 1 permit any
```

dans l'exemple l'entrée numéro 30 a été ajouté.

Il y a également la possibilité de reséquencer pour modifier l'ACL. La commande est :

***ip access-list resequence** acl-id starting-seq-num increment »*

Voici un exemple pour mieux comprendre :

```
R1(config)#do show access-lists
Standard IP access list 1
    1 deny   192.168.1.1
    3 deny   192.168.3.1
    2 deny   192.168.2.1
    4 deny   192.168.4.1
    5 permit any
R1(config)#
R1(config)#ip access-list resequence 1 10 10
R1(config)#
R1(config)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.3.1
    30 deny   192.168.2.1
    40 deny   192.168.4.1
    50 permit any
```

On peut voir ici que la commande

qui a été utilisé a permis de réorganiser l'ACL en commençant par la première entrée qui est : 1 qui a pris pour valeur 10 et sur laquelle est implémenté à chaque entrée la

valeur de 10 en plus.

On voit le résultat avec la commande « show access-lists »

Maintenant que nous avons vu les avantages d'utiliser les Named ACLs nous allons voir comment fonctionne un Extended ACLs

Comme précisé lors de l'introduction, le fonctionnement des Extended ACLs est le même que les Standard ACLs elles peuvent être numbered ou named comme les ACLs standard.

- Les ACLs Numbered utilisent les classements suivants : 100 - 199, 2000 - 2699

- Les Extended ACLs s'activent de la même manière que les Standard ACL du haut vers le bas.

La différence avec les Standard ACLs est qu'elles peuvent utiliser plus de paramètres de filtrage, donc elles sont plus complexe et plus précises que les ACLs Standard.

Nous nous baserons sur les paramètres principaux qui sont : la couche 4 protocole/port, l'adresse IP source, et l'adresse de destination.

Pour configurer une Extended Numbered ACL depuis le mode Global Config la commande est :

```
R1(config)#access-list number [permit | deny] protocol src-ip dest-ip
```

il faut s'assurer que le nombre soir bien compris dans le classement précisé auparavant

(100 - 199, 2000 – 2699)

ensuite on peut lancer la commande :

```
R1(config-ext-nacl)#[seq-num] [permit | deny] protocol src-ip dest-ip
```

Voici un exemple de la configuration d'une Extended ACL

```
R1(config)#ip access-list extended EXAMPLE
R1(config-ext-nacl)#deny ?
  <0-255>        An IP protocol number
  ahp            Authentication Header Protocol
  eigrp          Cisco's EIGRP routing protocol
  esp            Encapsulation Security Payload
  gre            Cisco's GRE tunneling
  icmp           Internet Control Message Protocol
  igmp           Internet Gateway Message Protocol
  ip             Any Internet Protocol
  ipinip         IP in IP tunneling
  nos            KA9Q NOS compatible IP over IP tunneling
  object-group   Service object group
  ospf           OSPF routing protocol
  pcp            Payload Compression Protocol
  pim            Protocol Independent Multicast
  sctp           Stream Control Transmission Protocol
  tcp            Transmission Control Protocol
  udp            User Datagram Protocol
```

On peut voir que lorsque l'on veut lancer la commande deny avec un « ? » pour afficher les options on peut voir une liste des protocol.

En toute première ligne on peut voir le Protocole IP, cela permet d'identifier le numéro protocole qui est encapsulé dans l'entête IP comme TCP ou UDP. On peut aussi utiliser directement le nom du protocole parmi ceux listés.

Si l'on veut utiliser le numéro du protocole voici quelques exemples :

1 : ICMP

6 : TCP

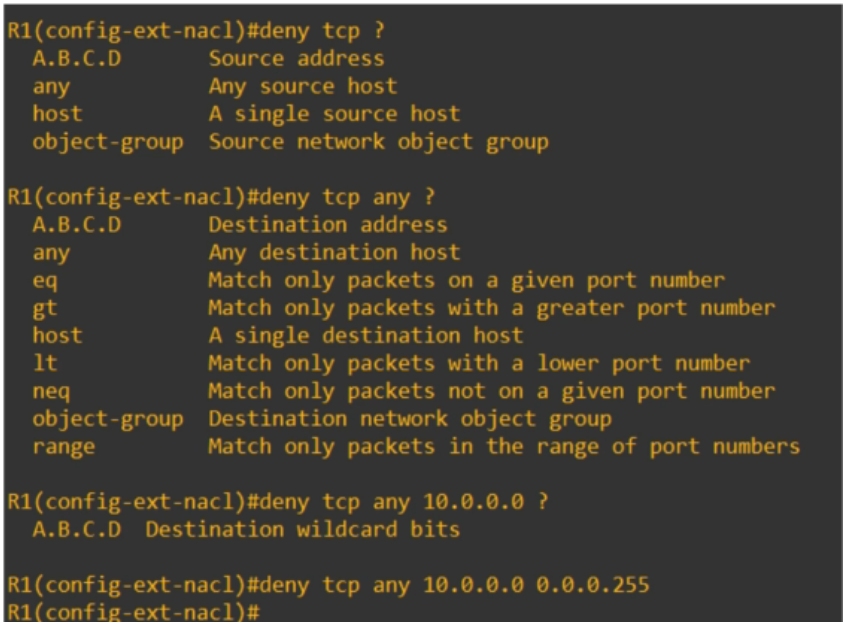
17 : UDP

88 : EIGRP

89 : OSPF

Donc il possible par exemple de bloquer des messages OSPF sur une interface par exemple ou bien renier un paquet ICMP pour bloquer le ping.

Voyons un exemple pour voir comment ajouter une adresse IP source et une adresse IP de destination à cette entrée d'ACL.



Dans l'exemple ci dessus a été sélectionné le protocole TCP. Donc n'importe quelle paquet IP avec un segment TCP va correspondre à cette partie de l'entrée.

Il reste tout de même nécessaire de spécifier l'adresse IP source et de destination pour les faire correspondre.

Il est aussi à noter que dans l'Extended ACLs pour spécifier la source ou destination /32 il faut utiliser l'option **host** ou spécifier le masque de sous réseau. On ne peut pas juste écrire l'adresse sans ces autres paramètres, avec les ACLs Standard c'est possible mais pas avec les Extended.

C'est pour cela que dans l'exemple a été utilisé l'option « any » pour faire correspondre toutes les adresses IP source.

A la suite a été spécifié l'adresse IP de destination, il y a plusieurs options possible mais nous allons voir surtout les options « Destination Adresse » « Any » « Host ».

Dans l'exemple a été spécifié l'option avec l'adresse de destination 10.0.0.0 et le masque de sous réseau /24.

Nous allons à présent voir plusieurs exemples de configuration d'ACLs :

1. Permettre tout le trafique :

```
R1(config-ext-nacl)#permit ip any any
```

2. Empêcher 10.0.0.0/16 d'envoyer le trafique UDP à 192.168.1.1/32

```
R1(config-ext-nacl)#deny udp 10.0.0.0 0.0.255.255 host 192.168.1.1
```

3. Empêcher 172.16.1.1/32 de pinguer l'host dans 192.168.0.0/24

```
R1(config-ext-nacl)#deny icmp host 172.16.1.1 192.0.0 0.0.0.255
```

Lorsque l'on veut faire correspondre TCP/UDP, on peut optionnellement spécifier la source et/ou le port de destination à faire correspondre.

La commande est la suivante :

***R1(config-ext-nacl)#deny tcp** src-ip dest-ip*

Si l'on veut spécifier le port source ou de destination il faut le spécifier après l'adresse IP source.

***R1(config-ext-nacl)#deny tcp** src-ip **eq** src-port-num **eq** dest-ip*

Les différentes options possible qui peuvent remplacer « eq » sont les suivantes :

- « *eq 80* » (equal) c'est égal au port 80 donc cela va faire correspondre la source TCP du port 80.

- « *gt 80* » (greater than). Par exemple gt 80 va faire correspondre tous les ports qui ont une valeur supérieur à 80.

- « *lt 80* » (less than) pour faire correspondre les ports inférieurs à 80

- « *neq 80* » (not equal) pour faire correspondre tous les autres port à l'exception du port 80.

- « *range 80 100* » pour faire correspondre le classement des ports compris entre 80 et 100.

Voici une liste du nom des différents protocoles utilisés :

TCP
- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)

UDP
- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)

TCP & UDP
- DNS (53)

Il est à noter que lorsque l'on spécifie le protocole, l'adresse IP source, l'adresse Ip de destination, etc. Le paquet doit correspondre à toutes les valeurs pour correspondre l'entrée de l'ACL. Même si cela correspond à tous les autres paramètres à l'exception d'un seul le paquet ne correspondra pas à cette entrée de l'ACL.

Donc en résumé l'Extended ACL permet d'être très précis dans le filtrage des paramètres.

Voici quelques exemples d'ACLs :

1. permettre le trafique de 10.0.0.0/16 d'accéder au serveur à 2.2.2.2/32 en utilisant HTTPS.

```
R1(config-ext-nacl)#permit tcp 10.0.0.0 0.0.255.255 2.2.2.2 0.0.0.0 eq 443
```

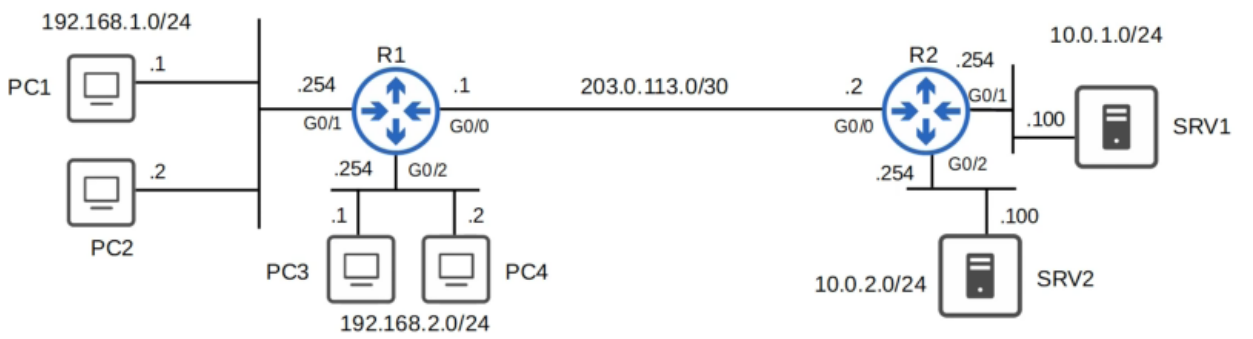
2. Empêcher tous les hôtes d'utiliser le port source UDP de 20000 à 30000 d'accéder au serveur 3.3.3.3/32

```
R1(config-ext-nacl)#deny udp any range 20000 30000 host 3.3.3.3
```

3. Permettre à l'hôte dans 172.16.1.0/24 d'utiliser le port source TCP supérieur à 9999 d'accéder à tous les ports TCP du serveur 4.4.4.4/32 excepté le port 23.

```
R1(config-ext-nacl)#permit tcp 172.16.1.0 0.0.0.255 gt 9999 host 4.4.4.4 neq 23
```

A présent nous allons configurer le réseau suivant pour que les conditions édictés ci dessous sous bien respectés.



Les conditions de configurations sont les suivantes :

- Les hôtes dans 192.168.1.0/24 ne peuvent pas utiliser HTTPS pour accéder au SRV1

- Les hôtes dans 192.168.2.0/24 ne peuvent pas accéder à 10.0.2.0/24

- Les hôtes dans 192.168.1.0/24 ou 192.168.2.0/24 peut pinger 10.0.1.0/24 ou 10.0.2.0/24

Voici les commandes que nous allons utiliser pour que la première partie de conditions :

```
R1(config)#ip access-list ectended HTTP\_SRV1
R1(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100 eq 443
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/1
R1(config-if)#ip access-group HTTP\_SRV1 in
```

Il est à noter que les Extended ACLs doivent s'appliquer le plus proche possible de la source, pour limiter le trajet du trafique avant qu'il ne soit renié.

(Standard ACLs sont plus spécifiques, donc s'ils s'appliquent plus proche de la source il y a un risque de bloquer plus de trafique que voulu).

Pour la deuxième partie des conditions voici les commandes à utiliser :

```
R1(config)#ip access-lis extended BLOCK_10.0.2.0/24
R1(config-ext-nacl)#deny ip 192.168.2.0 0.0.0.255 10.0.2.0 0.0.0.255
R1(config-ext-nacl)#permit ip any any
```

Pour la troixième partie des conditions voici les commandes à utiliser :

```
R1(config)#ip access-list extended BLOCK_ICMP
R1(config-ext-nacl)#deny icmp 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
R1(config-ext-nacl)#deny icmp 192.168.1.0 0.0.0.255 10.0.2.0 0.0.0.255
R1(config-ext-nacl)#deny icmp 192.168.2.0 0.0.0.255 10.0.1.0 0.0.0.255
R1(config-ext-nacl)#permit ip nay any
R1(config-ext-nacl)#interface g0/0
```

R1(config-if)#ip access-group BLOCK_ICMP out

Voici les résultats des commandes pour afficher les commandes :

```
R1#show access-lists
Extended IP access list BLOCK_10.0.2.0/24
    10 deny ip 192.168.2.0 0.0.0.255 10.0.2.0 0.0.0.255
    20 permit ip any any
Extended IP access list BLOCK_ICMP
    10 deny icmp 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
    20 deny icmp 192.168.1.0 0.0.0.255 10.0.2.0 0.0.0.255
    30 deny icmp 192.168.2.0 0.0.0.255 10.0.1.0 0.0.0.255
    40 permit ip any any
Extended IP access list HTTP_SRV1
    10 deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100 eq 443
    20 permit ip any any
```

```
R1#show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 203.0.113.1/30
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is BLOCK_ICMP
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
```

Cours 36 : CDP & LLDP

Dans ce cours nous verrons ce que sont les deux couches 2 de découvertes de protocoles qui sont CDP et LLDP.

Nous ferons une introduction à la couche 2 de découverte de protocole, puis nous verrons en détail comment fonctionne les protocole Cisco Discovery Protocol (CDP) et Link Layer Discovery Protocol (LLDP)

La couche 2 de découverte des protocoles comme CDP et LLDP servent à partager et découvrir des informations à propos des appareils voisins (connectés).

Il sont appelés couche 2 de découverte de protocole puisque les protocoles eux mêmes fonctionnent avec la couche couche 2 et n'utilisent pas d'adresse IP.

En voyant des captures Wireshark on pourra observer qu'il n'est pas présent de paquets IP à l'intérieur de la trame envoyés par CDP et LLDP.

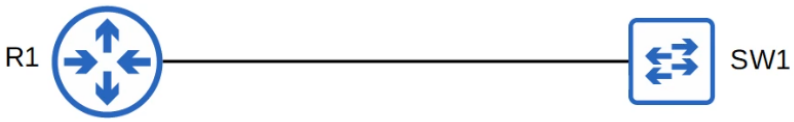
Bien qu'ils soient des protocoles de couche 2 de découverte de protocoles, ils peuvent être utilisés pour partager des informations de couche 3 comme les adresses IP. Les informations partagés incluent les nom d'hôtes, les adresses IP, les types d'appareils, etc..

CDP est un protocole propriétaire Cisco développé par Cisco pour les appareils Cisco. donc si le réseau utilisé seulement des appareils Cisco CDP est adapté, par contre si le réseau est composé de plusieurs vendeurs comme Cisco, Juniper, Palo Alto, il faudra utiliser le protocole : LLDP.

LLDP est un protocole de l'industrie standard, IEEE 802.1AB.

Puisque ces protocoles partagent des informations à propos des appareils du réseau, ils peuvent être considérés comme risqués en sécurité et ne sont pas souvent utilisés, les administrateurs ou ingénieurs réseau peuvent décider s'ils veulent les utiliser ou non dans le réseau.

Pour comprendre comment ces protocoles fonctionnent voici deux appareils R1 et SW1 :



R1 envoie de manière périodique des trames à SW1 en lui donnant des informations comme le nom du routeur R1, le type d'appareils, l'ID d'interface, l'adresse IP, etc...

SW1 de même et envoie aussi périodiquement des trames à R1. A noter que SW1 n'envoie pas d'informations comme l'adresse IP à R1 puisqu'il s'agit d'un Switch son interface n'a pas d'adresse IP.

Voyons plus en détail le fonctionnement de CDP.

CDP est donc un protocole Cisco propriétaire, il est activé par défaut sur les appareils Cisco (Routeurs, Switchs, les pare feu, les téléphones IP, etc..).

Les messages CDP sont envoyés périodiquement en multicast à l'adresse MAC 0100 0CCC CCCC

Puisque les messages utilisent une adresse MAC il faudra penser au message redirigés à plusieurs appareils mais en vérité ça n'est pas le cas. Lorsqu'un appareil reçoit un message CDP il procède donc le message mais ne le repartage pas aux autres appareils. Donc ça n'est que les appareils connectés directement qui sont des voisins CDP.

Par défaut les messages CDP sont envoyés toutes les 60 secondes à toutes les interfaces actives.

Ce sont des messages qui contiennent des informations comme le hostname, l'adresse IP, etc...

Lorsque l'appareil reçoit ces messages CDP depuis l'appareil voisin il ajoute une entrée de l'appareil dans sa table de voisin.

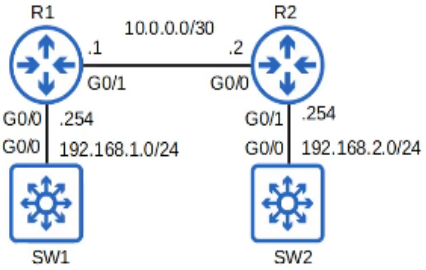
Si un voisin est déconnecté il y a un temps d'attente de 180 secondes, puis si le voisin n'est plus détecté il est supprimé de la table de voisins CDP.

Cela fait que les tables de voisins CDP n'ont pas d'entrées d'anciens voisins qui ne sont plus existants.

Il y a deux versions de CDP, Version 1 et Version 2, la version 2 est utilisé par défaut.

La version CDP Version 1 est ancienne donc il n'y a probablement pas besoin de l'utiliser.

Les principales différences entre les V1 et V2 est qu'il y a quelques fonctionnalités avancés comme l'habilité d'identifier un VLAN natif.



Pour démontrer l'utilisation de CDP nous utiliserons le réseau suivant :

Deux routeurs et deux commutateurs multicouches sont utilisés nous n'utilisons pas les fonctions de couches 3 sur les commutateurs.

Voici les commandes basiques à utiliser pour vérifier la configuration CDP :

```
R1#show cdp
```

```
R1#show cdp interface
GigabitEthernet0/0 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/2 is administratively down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/3 is administratively down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds

 cdp enabled interfaces : 4
 interfaces up          : 2
 interfaces down        : 2
```

```
R1#show cdp trafic
R1#show cdp interfaces
```

```
R1#show cdp
Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is  enabled
R1#
R1#show cdp traffic
CDP counters :
        Total packets output: 105, Input: 112
        Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
        No memory: 0, Invalid packet: 0,
        CDP version 1 advertisements output: 0, Input: 0
        CDP version 2 advertisements output: 105, Input: 112
R1#
```

la commande *show cdp* permet d'afficher la fréquence de temps d'envois de messages CDP, il est de 60 secondes par défaut, et de 180 secondes de temps d'attente de réponse (holdtime)

Par défaut cette commande affiche aussi quelle version de CDP est utilisé, la version 2 est celle utilisé par défaut.

Il est à noter que si CDP n'est pas activé sur un appareil on recevra le message suivant :

```
R1#show cdp
% CDP is not enabled
R1#
```

La commande ***show cdp traffic*** permet d'afficher combien de paquets et d'avertissements CDP l'appareil a envoyé et reçu.

La commande ***show cdp interface*** permet d'afficher des informations basiques à propos de chacune des interfaces. Il est aussi possible de spécifier certaines interface en entrant la commande mais lorsque l'on entre la commande sans préciser l'interface, on reçoit des informations sur toutes les interfaces.

Sur le résultat de la commande de la capture d'écran on peut voir l'information « encapsulation ARPA », il s'agit d'un type d'encapsulation Ethernet connu comme Ethernet 2.

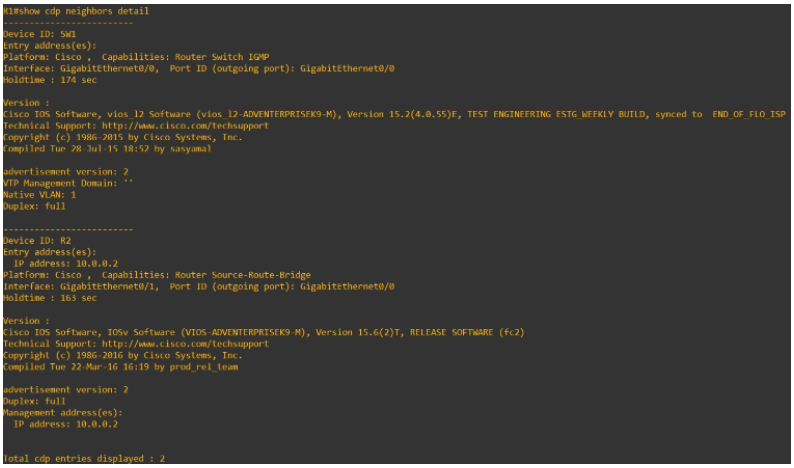
Pour afficher la table des voisins CDP on lance la commande :

```
R1#show cdp neighbors
```

```
R1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW1              Gig 0/0           153           R S I           Gig 0/0
R2               Gig 0/1           146            R B            Gig 0/0

Total cdp entries displayed : 2
R1#
```

Ces commandes permettent d'afficher des informations essentiels à propos de CDP, pour afficher des informations additionnels on peut utiliser la commande :

```
R1#show cdp neighbors detail
```



Comme on peut le voir plus d'informations apparaît pour chaque appareils voisins.

Par exemple le nom de l'OS, le type de VLAN (lorsqu'un commutateur) etc...

Si l'on veut afficher des informations détaillés à propos d'un seul appareil, il est possible de lancer la commande :

```
R1#show cdp entry R2
```

```
R1#show cdp entry R2
-------------------------
Device ID: R2
Entry address(es):
  IP address: 10.0.0.2
Platform: Cisco ,  Capabilities: Router Source-Route-Bridge
Interface: GigabitEthernet0/1,  Port ID (outgoing port): GigabitEthernet0/0
Holdtime : 178 sec

Version :
Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(2)T, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Tue 22-Mar-16 16:19 by prod_rel_team

advertisement version: 2
Duplex: full
Management address(es):
  IP address: 10.0.0.2
```

CDP est activé par défaut sur l'appareil mais aussi sur chacunes des interfaces par défaut.

Pour activer/désactiver CDP on peut lancer la commande :

```
R1(config)#[no] cdp run
```

Pour activer/désactiver une interface spécifique on peut lancer la commande :

```
R1(config-if)#[no] cdp enable
```

Il est possible de configurer le temps de CDP en lançant la commande :

```
R1(config)#cdp timer {seconds}
```

Pour configurer le temps d'attente CDP on lance la commande :

```
R1(config)#cdp holdtime {seconds}
```

Pour activer/désactiver CDPv2 on lance la commande :

```
R1(config)#[no] cdp advertise-v2
```

Voyons à présent plus en détail le fonctionnement pour la configuration du protocole LLDP (Link Layer Discovery Protocol)

LLDP est un protocole de l'industrie standard (IEEE 802.1AB)

CDP était le protocole originel et LLDP a été inventé plus tard pour qu'il y ait une version standard au niveau industriel.

Ce protocole est désactivé par défaut sur les appareils Cisco, donc il faut l'activer manuellement pour l'utiliser.

Un appareil peut lancer CDP et LLDP en même temps donc il n'y pas nécessité à en désactiver un.

Les messages LLDP sont périodiquement envoyés à l'adresse MAC multicast : 0180.C200.000E

Lorsqu'un appareil reçoit des messages LLDP, il procède le message mais ne le partage pas aux autres appareils voisins.

Donc seulement les appareils directement connectés peuvent devenir des voisins LLDP.

Par défaut les messages LLDP sont envoyés toutes les 30 secondes, et toutes les 120 secondes pour les messages d'attente de réponses.

LLDP a également un temps additionnel appelé le « délai de réinitialisation ».

Si LLDP est activé de manière global ou bien sur une interface spécifique le temps aura un délai de l'initialisation actuelle de LLDP et le temps sera de 2 secondes par défaut.

Voici les commandes nécessaires dans la configuration de LLDP.

Puisque LLDP est désactivé par défaut, il faut l'activer sur toutes les interfaces. La configuration est légèrement différente de CDP.

Pour activer LLDP on lance la commande :

```
R1(config)#lldp run
```

C'est la même commande que pour CDP sauf que l'on remplace cdp par lldp. Si l'on veut le désactiver on lance la commande :

```
R1(config)#no lldp run
```

Pour activer LLDP sur une interface spécifique (tx) on utilise la commande suivante :

```
R1(config-if)#lldp transmit
```

Cette commande lancé fera que l'interface commencera à lancer des messages LLDP.

Par contre lorsqu'il recevra un message LLDP, il le rejettera.

Pour activer LLDP en réception il faut lancer une autre commande qui est :

```
R1(config-if)#lldp receive
```

Pour configurer le temps d'envois des messages de LLDP on lance la commande suivante :

```
R1(config)#lldp timer {secondes}
```

Pour configurer le temps d'attente de réponse entre les messages on lance la commande :

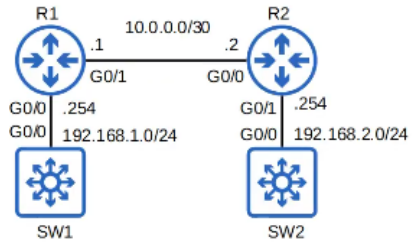
```
R1(config)#lldp holdtime {secondes}
```

Pour réinitialiser le temps LLDP on lance la commande :

```
R1(config)#lldp reinit {secondes}
```

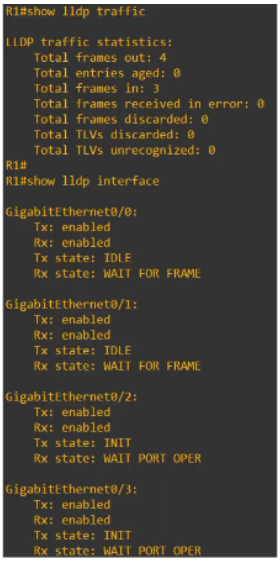
Ces commandes sont similaires à la configuration CDP la différence se situe dans la configuration des interfaces ou il faut activer la transmission et la réception dans LLDP.

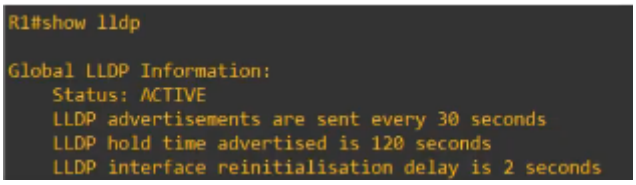
Sur le réseau suivant LLDP a été activé sur les interfaces.



Voici le résultat des commandes pour afficher la configuration des interfaces :

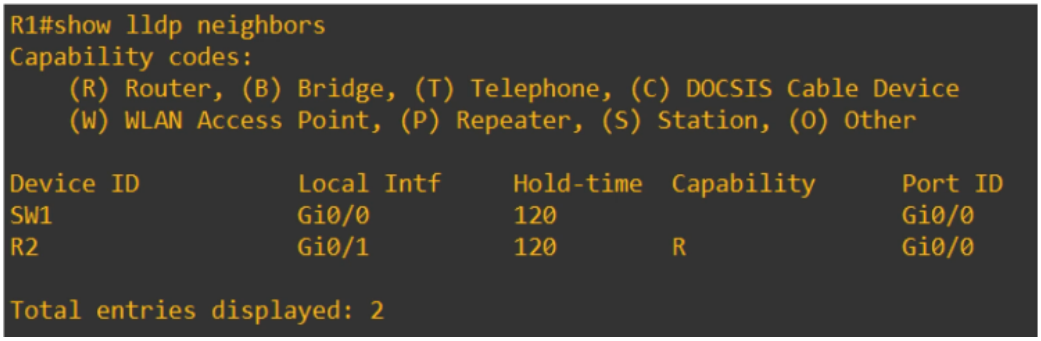
```
R1#show lldp
R1#show lldp trafic
R1#show lldp interface
```





Voici la commande pour afficher la table de voisins sur LLDP :

```
R1#show lldp neighbors
```



Le résultat de la commande affiche des informations similaires à la commande pour afficher la table de voisins CDP

Pour afficher en détail la configuration de la table de voisin on lance la commande suivante :

```
R1#show lldp neighbors detail
```

```
R1#show lldp neighbors detail
------------------------------------------------
Local Intf: Gi0/0
Chassis id: 0c04.41d2.1a00
Port id: Gi0/0
Port Description: GigabitEthernet0/0
System Name: SW1

System Description:
Cisco IOS Software, vios_l2 Software (vios_l2-ADVENTERPRISEK9-M), Version 15.2(4.0.55)E, TEST ENGINEERING ESTG_WEEKLY BUILD, synced to  END_OF_FLO_ISP
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compi

Time remaining: 99 seconds
System Capabilities: B,R
Enabled Capabilities - not advertised
Management Addresses - not advertised
Auto Negotiation - not supported
Physical media capabilities - not advertised
Media Attachment Unit type - not advertised
Vlan ID: - not advertised

------------------------------------------------
Local Intf: Gi0/1
Chassis id: 0c04.418d.a400
Port id: Gi0/0
Port Description: GigabitEthernet0/0
System Name: R2

System Description:
Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(2)T, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Tue 22-Mar-16 16:19 by prod_rel_team

Time remaining: 92 seconds
System Capabilities: B,R
Enabled Capabilities: R
Management Addresses:
    IP: 10.0.0.2
```

Cette commande permet d'afficher quelques informations de plus par rapport à CDP, comme par exemple : la section « System Capabilities : B,R » permet de dire que le système voisin peut faire office de pont mais aussi de routeur. (C'est un commutateur niveau 3)

La section « Enabled capabilities » indique quelle mode de fonctionnement est activé sur le système voisin.

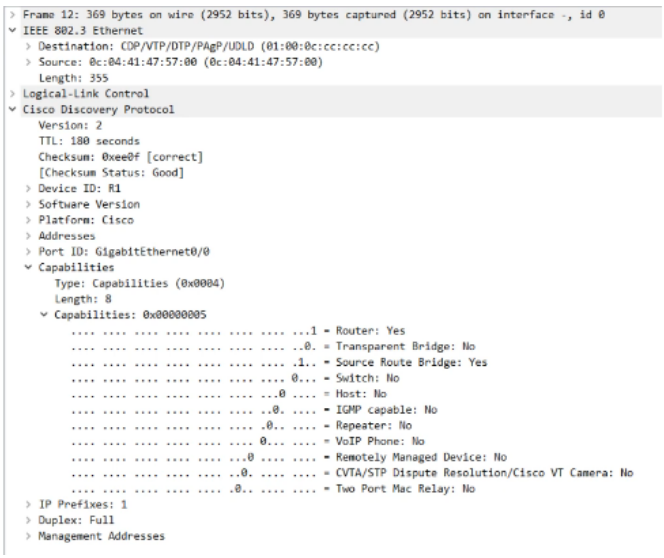
Il est possible de n'afficher les détail de l'interface de seulement un appareil spécifié la commande est la suivante :

```
R1#lldp entry SW1
```

Dans ce cas les informations du SW1 sont affichés :

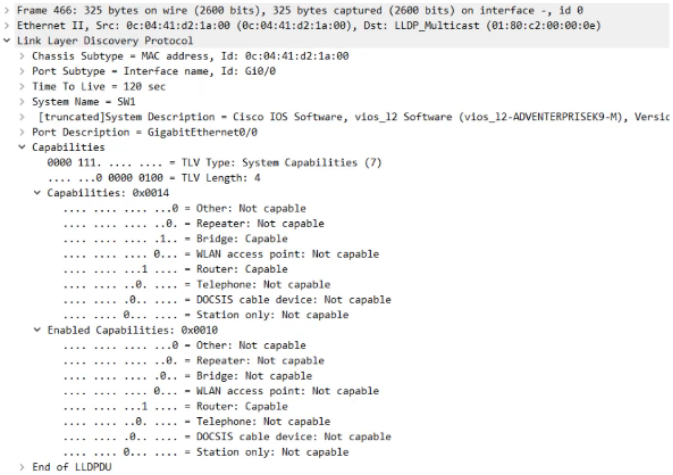
```
R1#show lldp entry SW1

Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
------------------------------------------------
Local Intf: Gi0/0
Chassis id: 0c04.41d2.1a00
Port id: Gi0/0
Port Description: GigabitEthernet0/0
System Name: SW1

System Description:
Cisco IOS Software, vios_l2 Software (vios_l2-ADVENTERPRISEK9-M), Version 15.2(4.0.55)E, TEST ENGINEERING ESTG_WEEKLY BUILD, synced to  END_OF_FLO_ISP
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compi

Time remaining: 119 seconds
System Capabilities: B,R
Enabled Capabilities: R
Management Addresses - not advertised
Auto Negotiation - not supported
Physical media capabilities - not advertised
Media Attachment Unit type - not advertised
Vlan ID: - not advertised
```

Voici une capture Wireshark de CDP :



On peut voir le TTL qui est de 180 seconde, c'est le même que celui configuré avec la commande *lldp holdtime*

Voici une capture Wireshark de LLDP :



On peut voir affiché ici les fonctions possibles sur SW1 qui sont le routage et le pont (ou Bridge)

Cours 37 : Network Time Protocole (NTP)

Dans ce cours nous verrons le fonctionnement du protocole Network Time Protocole (NTP) qui permet la synchronisation du temps de manière précise entre tous les appareils.

Nous expliquerons pourquoi le temps est un élément important pour les appareils d'un réseau.

Nous ferons la configuration manuelle du temps sans utiliser NTP.

Nous expliquerons ensuite les points basique du protocole NTP ainsi que comment le configurer.

Tous les appareils ont une horloge interne (Routeurs, Switchs, PC, etc...)

Dans les IOS Cisco on peut afficher le temps en lançant la commande : *show clock*

```
R1#show clock
*00:16:00.857 UTC Sat Dec 26 2020
```

Sur le résultat de la commande précédente on peut voir que la commande a été lancé à 12 :16 AM

0 secondes et 847 millisecondes, le Samedi 26 Décembre 2020.

La zone de temps est celle par défaut qui est UTC (Coordinated Universal Time)

Si l'on lance la commande : *show clock detail*

on peut voir la source sur laquelle est tiré l'heure :

```
R1#show clock detail
*00:19:49.411 UTC Sat Dec 26 2020
Time source is hardware calendar
```

Sur le résultat de la commande précédente on peut voir que l'heure est tiré du calendrier matériel ou aussi l'horloge interne de l'appareil.

Lorsque l'appareil est configuré avec l'horloge interne l'appareil aura un décalage de temps, donc ça n'est pas la meilleure source pour le temps.

La raison principale pour laquelle il est important d'avoir un temps précis sur un appareil est afin d'avoir des logs et une journalisation précise pour résoudre les problèmes.

Syslog est le protocole utilisé pour garder les logs de l'appareil.

La commande pour afficher les logs d'un appareil est : *show logging*

```
R2#show logging
!output abbreviated!
*Dec 27 00:50:20.005: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.122.192 on GigabitEthernet0/0 from LOADING to FULL,
Loading Done
*Dec 27 01:06:38.653: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done
*Dec 27 01:07:07.311: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done
*Dec 27 01:08:29.924: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from FULL to DOWN, Neighbor
Down: Dead timer expired
*Dec 27 01:09:10.714: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done

R2#show clock
*01:17:06.706 UTC Sun Dec 27 2020
```

On peut voir que l'interface 0/1 de l'appareil est passé de l'état de LOADING à FULL à 01 :09 :10

Avec le protocole OSPF

Sur un autre appareil R3 on lance la même commande afin d'afficher les logs :

```
R3#show logging
!output abbreviated!
May 23 16:24:17.320: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from LOADING to FULL, Loading
Done
May 23 16:25:08.758: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from FULL to DOWN, Neighbor
Down: Interface down or detached
May 23 16:25:10.714: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to down
May 23 16:25:11.716: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to down
May 23 16:26:14.976: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
May 23 16:26:15.977: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
May 23 16:26:20.618: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from LOADING to FULL, Loading
Done

R3#show clock
16:30:37.020 UTC Fri May 23 2008
```

Sur cet appareil on peut voir que l'heure n'est pas du tout identique à celle de l'appareil R2

Il sera donc difficile de faire une relation entre les logs de ces deux appareils c'est aussi pour cela qu'un temps précis est important sur les appareils.

Voyons comment configurer manuellement l'heure sur un appareil.

On peut manuellement configurer le temps d'un appareil avec la commande : ***clock set***

```
R2#clock set ?
  hh:mm:ss  Current Time

R2#clock set 14:30:00 ?
  <1-31>  Day of the month
  MONTH   Month of the year

R2#clock set 14:30:00 27 ?
  MONTH  Month of the year

R2#clock set 14:30:00 27 Dec ?
  <1993-2035>  Year

R2#clock set 14:30:00 27 Dec 2020 ?
  <cr>

R2#clock set 14:30:00 27 Dec 2020
R2#show clock detail
14:30:05.887 UTC Sun Dec 27 2020
Time source is user configuration
```

On peut voir sur les commandes précédentes que l'heure a été configuré pour : 14 :30 :05.887 le 27 Décembre 2020

On affiche ensuite l'heure avec la commande : ***show clock detail***

Toutes ces commandes ont été exécutés à partir du mode global et non pas mode config.

Une chose à noter est que le calendrier matériel (horloge interne) est le temps par défaut. Le temps du matériel et du logiciel, sont deux horloges différentes et peuvent être configurés séparément.

```
R2#calendar set 14:35:00 ?
  <1-31>  Day of the month
  MONTH   Month of the year

R2#calendar set 14:35:00 27 ?
  MONTH  Month of the year

R2#calendar set 14:35:00 27 Dec ?
  <1993-2035>  Year

R2#calendar set 14:35:00 27 Dec 2020 ?
  <cr>

R2#calendar set 14:35:00 27 Dec 2020
R2#show calendar
14:35:07 UTC Sun Dec 27 2020
```

On peut configurer manuellement l'horloge matériel avec la commande : ***calendar set***

On peut vouloir synchroniser l'horloge « clock » et le calendrier « calendar »

Pour synchroniser le calendrier à l'horloge du temps on lance la commande :

```
R1#clock update-calendar
```

```
R2#show clock
14:38:14.301 UTC Sun Dec 27 2020
R2#show calendar
00:00:03 UTC Sun Dec 27 2020
R2#clock update-calendar
R2#show clock
14:38:22.181 UTC Sun Dec 27 2020
R2#show calendar
14:38:23 UTC Sun Dec 27 2020
```

On peut voir qu'en lançant la commande l'horloge du calendrier est à présent celle de l'horloge interne.

Pour synchroniser l'horloge du temps au calendrier on lance la commande :

```
R1#clock read-calendar
```

```
R2#show clock
00:00:15.788 UTC Mon Sep 6 1993
R2#show calendar
14:55:07 UTC Sun Dec 27 2020
R2#clock read-calendar
R2#show clock
14:55:12.522 UTC Sun Dec 27 2020
R2#show calendar
14:55:15 UTC Sun Dec 27 2020
```

Cette fois ci on peut voir qu'en lançant la commande l'horloge interne est à présent celle du calendrier.

Voyons à présent comment configurer la zone de temps.

Pour configurer la zone de temps on utilise la commande : *clock timezone*

```
R2(config)#do show clock
15:13:33.985 UTC Sun Dec 27 2020
R2(config)#clock timezone ?
  WORD  name of time zone

R2(config)#clock timezone JST ?
  <-23 - 23>  Hours offset from UTC

R2(config)#clock timezone JST 9 ?
  <0-59>  Minutes offset from UTC
  <cr>

R2(config)#clock timezone JST 9
R2(config)#do show clock
00:13:45.414 JST Mon Dec 28 2020
R2(config)#do clock set 15:15:00 Dec 27 2020
R2(config)#do show clock
15:15:02.129 JST Sun Dec 27 2020
```

On peut voir que sur les commandes précédentes le temps à été configuré sur la zone JST il faut préciser ensuite le nombre d'heures de différences par rapport à UTC (ici 9h puisqu'il y a 9h de différence entre la zone Japon et UTC)

Il y a encore une autre chose à prendre en compte dans la configuration du temps, il s'agit du temps d'été car il peut y avoir certaines heures ou l'heure recule ou bien avance etc...

Il est possible de configurer les appareils Cisco pour qu'ils procèdent à cela automatiquement.

Par exemple l'heure d'été au Canada commence le second Dimanche de mars à 2 :00 et termine le premier Dimanche de Novembre à 2 :00.

Pour configurer cela on lance la commande : *clock summer-time*

```
R2(config)#clock summer-time ?
  WORD  name of time zone in summer
R2(config)#clock summer-time EDT ?
  date       Configure absolute summer time
  recurring  Configure recurring summer time
R2(config)#clock summer-time EDT recurring ?
  <1-4>  Week number to start
  first  First week of the month
  last   Last week of the month
  <cr>
R2(config)#clock summer-time EDT recurring 2 ?
  DAY  Weekday to start
R2(config)#clock summer-time EDT recurring 2 Sunday ?
  MONTH  Month to start
R2(config)#clock summer-time EDT recurring 2 Sunday March ?
  hh:mm  Time to start (hh:mm)
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 ?
  <1-4>  Week number to end
  first  First week of the month
  last   Last week of the month
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 ?
  DAY  Weekday to end
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 Sunday ?
  MONTH  Month to end
R2(config)#$r-time EDT recurring 2 Sunday March 02:00 1 Sunday November ?
  hh:mm  Time to end (hh:mm)
R2(config)#$ recurring 2 Sunday March 02:00 1 Sunday November 02:00 ?
  <1-1440>  Offset to add in minutes
  <cr>
R2(config)#$ recurring 2 Sunday March 02:00 1 Sunday November 02:00
```

On commence par spécifier la zone de temps avec EDT dans notre cas.

Puis on indique si l'on souhaite que l'appareil fasse l'action seulement à la date préciser (date) ou bien tous les ans (recurring). On indique ensuite à quelle semaine du mois devra être appliqué (ici la deuxième semaine du mois) on indique le jour de la semaine (Dimanche) puis le mois (Mars) et l'heure (2 :00). On précise ensuite la fin du temps d'été (1er Dimanche du mois de Novembre à 2 :00 dans le cas du Canada) avec d'abord la semaine (1) puis le jour (Dimanche) le mois (Novembre) et l'heure (2 :00).

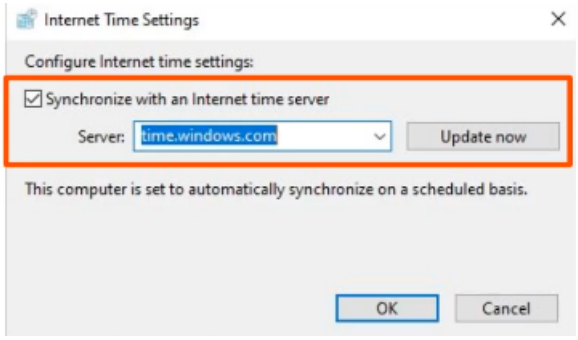
Voyons à présent comment fonctionne le protocole Network Time Protocole (NTP)

Comme on a pu l'expliquer la configuration du temps manuellement n'est pas très précise et peut demander du temps si elle était faite manuellement à chaque fois pour tous les appareils.

Le protocole NTP permet de synchroniser automatiquement le temps à travers le réseau.

Sur un ordinateur Windows on peut voir que l'heure est synchronisé sur le serveur :

time.windows.com



Le serveur Windows de synchronisation du temps provient de la même source que le DNS de Google comme on peut le voir avec la commande nslookup sur Windows.

```
C:\Users\user>nslookup time.windows.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    time.microsoft.akadns.net
Address:  20.43.94.199
Aliases:  time.windows.com


C:\Users\user>nslookup time.google.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    time.google.com
Addresses:  2001:4860:4806::
          2001:4860:4806:c::
          2001:4860:4806:8::
          2001:4860:4806:4::
          216.239.35.12
          216.239.35.8
          216.239.35.4
          216.239.35.0
```

Les clients NTP font la requête du temps depuis le serveur NTP.

NTP permet une précision du temps à 1millisecondes près si le serveur NTP est sur le même LAN ou de près de 50millisecondes si connecté à un serveur NTP à travers WAN ou Internet.

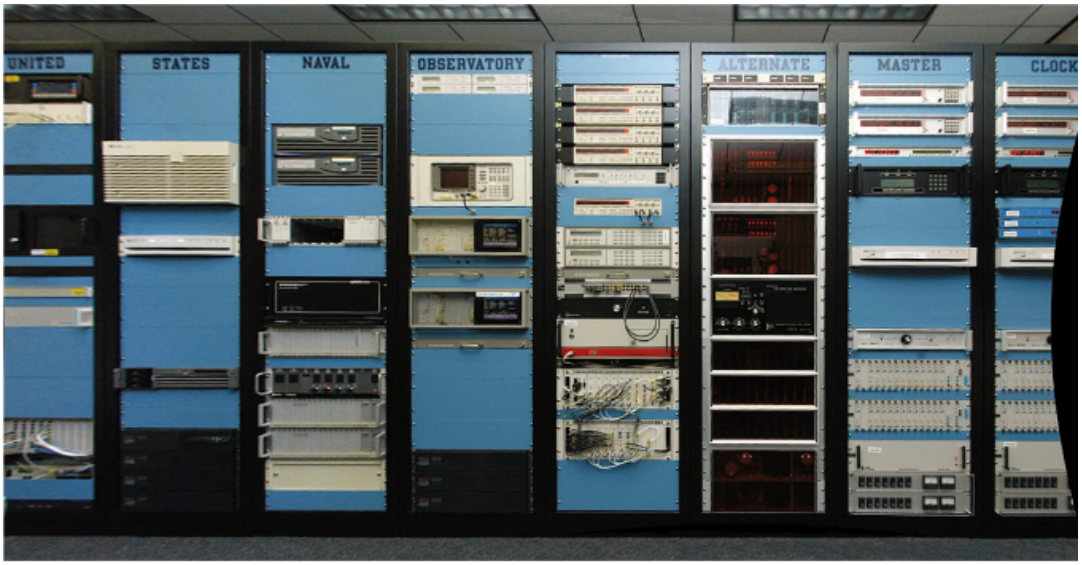
Certains serveurs NTP peuvent être « meilleurs » que d'autres. La distance d'un serveur NTP depuis l'horloge de référence est appelé stratum.

NTP utilise le port UDP 123 pour communiquer.

Une horloge de référence est un appareil qui possède un temps très précis avec un horloge atomique ou une horloge par GPS.

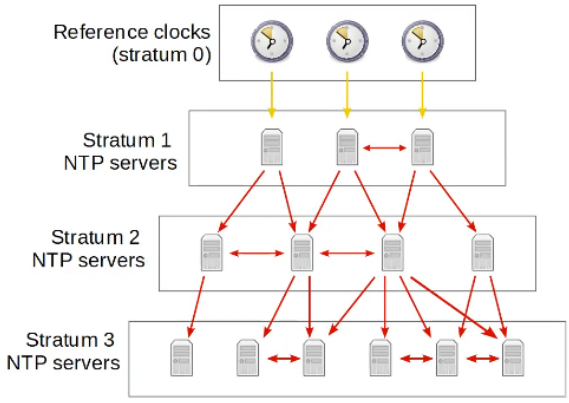
Les horloge de référence sont stratum 0 dans le hiérarchie NTP.

Les serveurs NTP directement connectés à l'horloge de référence sont stratum 1



Voici à quoi ressemble une horloge de référence.

Les appareils Cisco ne peuvent pas synchroniser directement avec une horloge stratum 0 mais ils peuvent synchroniser leurs horloge avec un serveur NTP de stratum 1



Voici un schéma des différents niveaux des serveurs. Il y a ici 4 niveau qui vont jusqu'au stratum 3

Le stratum 15 est le niveau maximum, il n'y a pas de niveau inférieur à celui ci après cela l'appareil ne sera pas synchronisé. Des appareils peuvent aussi s'appairer avec d'autres situés dans le même stratum pour qu'ils fournissent un temps plus précis, ceci est appelé le mode « symétrique actif ».

Les appareils Cisco peuvent opérer en 3 modes :

- Mode serveur

- Mode Client

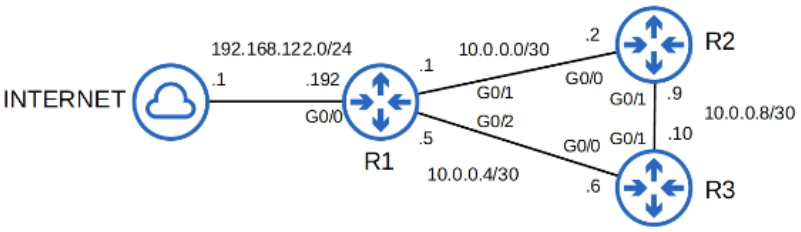
- Mode Symétrique actif

Ils peuvent être dans les trois modes en même temps. Un client NTP peut être synchronisé à plusieurs serveurs NTP.

Les serveurs NTP qui reçoivent leurs temps directement depuis l'horloge de référence (stratum 1) sont aussi appelé les serveurs primaire.

Les serveurs NTP qui reçoivent leurs temps depuis l'horloge d'autres serveurs (stratum 2 et plus) sont aussi appelé les serveurs secondaires, ils fonctionnent en mode client et serveur en même temps.

Nous verrons comment configurer NTP en utilisant le réseau suivant :



Comme on peut le voir le serveur Google peut utiliser les adresses IP suivantes :

```
C:\Users\user>nslookup time.google.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    time.google.com
Addresses:  2001:4860:4806::
          2001:4860:4806:c::
          2001:4860:4806:8::
          2001:4860:4806:4::
          216.239.35.12
          216.239.35.8
          216.239.35.4
          216.239.35.0
```

216.239.35.12

216.239.35.8

216.239.35.4

216.239.35.0

C'est pour cela qu'on lance les commandes suivantes afin de configurer le routeur R1 avec le serveur NTP de Google, pour cela on lance les commandes :

```
R1(config)#ntp server 216.239.35.0
```

```
R1(config)#ntp server 216.239.35.0
R1(config)#ntp server 216.239.35.4
R1(config)#ntp server 216.239.35.8
R1(config)#ntp server 216.239.35.12
```

Afin d'afficher les serveurs NTP utilisés pour la synchronisation on peut utiliser la commande :

```
R1#show ntp associations
```

```
R1#show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~216.239.35.0    .GOOG.           1     43     64    17 62.007 1401.54  0.918
+~216.239.35.8    .GOOG.           1     43     64    17 64.220 1416.65  0.939
+~216.239.35.4    .GOOG.           1     47     64    17 57.669 1402.11  0.916
+~216.239.35.12   .GOOG.           1     39     64    17 62.229 1409.03  0.960
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

Le serveur de synchronisation est ici le premier avec l'étoile devant.

Une autre commande utile pour afficher le statut de NTP est la commande :

```
R1#show ntp status
```

```
R1#show ntp status
Clock is synchronized, stratum 2, reference is 216.239.35.12
nominal freq is 1000.0003 Hz, actual freq is 999.5003 Hz, precision is 2**14
ntp uptime is 295800 (1/100 of seconds), resolution is 1001
reference time is E393F0A9.1F758C5B (05:50:33.122 UTC Mon Dec 28 2020)
clock offset is 1343.7280 msec, root delay is 49.13 msec
root dispersion is 2275.31 msec, peer dispersion is 3.44 msec
loopfilter state is 'SPIK' (Spike), drift is 0.000499999 s/s
system poll interval is 64, last update was 173 sec ago.
```

On peut voir affiché que le serveur est synchronisé à un serveur stratum 2.

On synchronise le serveur NTP en utilisant la commande :

```
R1(config)#do show clock detail
06:56:32.315 UTC Mon Dec 28 2020
Time source is NTP
R1(config)#do show calendar
05:23:06 UTC Mon Dec 28 2020
R1(config)#clock timezone JST 9
R1(config)#ntp update-calendar
R1(config)#do show clock detail
15:57:33.078 JST Mon Dec 28 2020
Time source is NTP
R1(config)#do show calendar
15:57:36 JST Mon Dec 28 2020
```

```
R1(config)#ntp update-calendar
```

Dans le réseau nous utiliserons le routeurs R1 pour qu'il soit le serveur NTP de R2.

Il faut tout d'abord configure une interface loopback. Pour cela on lance les commandes suivantes :

```
R1(config)#interface loopback0
R1(config-if)#ip address 10.1.1.1 255.255.255.255
R1(config-if)#exit
R1(config)#ntp source loopback0
```

En temps normal R2 synchronise son temps directement avec sa connexion avec R1, mais si la connexion est interrompue avec R2 après un câble déconnecté, R2 pourra toujours se connecter au serveur R1 en passant par le routeur R3, car l'interface loopback aura permis que l'interface soit sur tout le réseau.

Une fois l'interface loopback configuré on peut configurer le client R2 au serveur R1, pour cela on lance la commande :

```
R2(config)#ntp server 10.1.1.1
```

On affiche le serveur NTP avec la commande :

```
R2(config)#do show ntp associations
```

Pour afficher le statut du client NTP on lance la commande :

```
R2(config)#do show ntp status
```

```
R2(config)#ntp server 10.1.1.1
R2(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.1.1.1        216.239.35.12    2      0     64     1  7.038 -13.128 3937.5
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R2(config)#do show ntp status
Clock is synchronized, stratum 3, reference is 10.1.1.1
...
```

On fini par configurer le Routeur R3 avec les commandes suivantes :

```
R3(config)#ntp server 10.1.1.1
R3(config)#ntp server 10.2.2.2
```

Ici a été configuré les serveurs NTP : R1 et R2 sur le routeur R3.

On affiche ensuite les serveur NTP configurés avec :

```
R3(config)#do show ntp associations
```

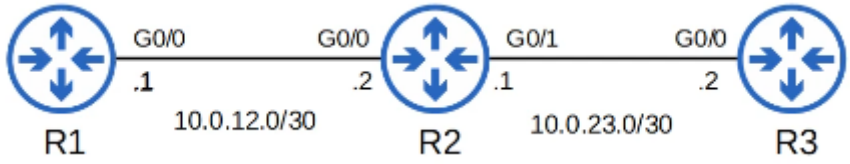
```
R3(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.1.1.1        216.239.35.0     2      1     64     0  0.000   0.000 15937.
 ~10.2.2.2        10.1.1.1         3      1     64     0  0.000   0.000 15937.
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

Ici le serveur privilégié est le premier avec l'adresse du routeur R1, cela est dû au fait que le stratum est le plus bas (stratum 2) que le routeur R2 (stratum 3)

Nous avons vu comment configurer les routeurs sur un serveur NTP connecté sur Internet. Nous allons voir comment configurer s'il n'y a pas de serveur NTP connecté à Internet.

Pour cela nous allons configurer un routeur comme serveur NTP.

Nous utiliserons le réseau suivant



:

Nous utiliserons le Routeur R1 pour qu'il serve de serveur NTP pour les deux autres routeurs.

Pour cela on lance la commande : *ntp master*

```
R1(config)#ntp master ?
  <1-15>  Stratum number
  <cr>

R1(config)#ntp master
R1(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~127.127.1.1     .LOCL.           7      2     16   377  0.000   0.000  0.292
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R1(config)#do show ntp status
Clock is synchronized, stratum 8, reference is 127.127.1.1
...
```

Lorsque l'on affiche la configuration du serveur NTP on peut voir que l'horloge de référence est en loopback (donc sa propre adresse)

On peut voir que lorsque l'on affiche le statut du serveur NTP qu'il est au niveau stratum 8

Le stratum par défaut d'un ntp master est 8.

On configure ensuite les deux autres routeurs pour qu'il synchronise sur le serveur NTP qui est le routeur R1, pour cela on utilise les commandes : *ntp server*

```
R2(config)#ntp server 10.0.12.1

R2(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.0.12.1       127.127.1.1      8      2     64     1  5.263  62.494 187.64
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
R3(config)#ntp server 10.0.12.1

R3(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.0.12.1       127.127.1.1      8     45     64    17 21.534 -21.440  0.976
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

On peut configurer les routeurs pour qu'ils fonctionnent en mode symétrique actif, cela permet à ce que la synchronisation continue entre les deux appareils au cas ou par exemple le Routeur R1 serait dysfonctionnel.

On utilise pour cela la commande : *ntp peer*

```
R2(config)#ntp peer 10.0.23.2

R2(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.0.12.1       127.127.1.1      8     60     64    17 24.040 206.682  0.987
 ~10.0.23.2       10.0.12.1        9     33     64     0  0.000   0.000 15937.
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
R3(config)#ntp peer 10.0.23.1

R3(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.0.12.1       127.127.1.1      8     11     64    37 12.605  -7.406 63.575
 ~10.0.23.1       10.0.12.1        9      1     64     0  0.000   0.000 15937.
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

Il est possible de configurer l'authentification NTP même s'il s'agit d'une fonction optionnel.

Cela permet à ce que le client NTP synchronisent uniquement au serveurs voulue.

Pour configurer l'authentification NTP on lance les commandes suivante :

On commence par activer l'authentification NTP :

`ntp authenticate`

On crée ensuite les clef d'authentification NTP avec la commande :

`ntp authentication-key` *`key-number`* `md5` *`key`*

On spécifie le numéro de clé de confiance avec la commande :

`ntp trusted-key` *`key-number`*

On spécifie quelle clé utiliser pour le serveur avec la commande :

`ntp server ip-address key key-number`

Voici comment configurer l'authentification NTP sur le serveur NTP R1, puis comment les routeurs R2 et R3 s'y connectent :

```
R1(config)#ntp authenticate
R1(config)#ntp authentication-key 1 md5 jeremysitlab
R1(config)#ntp trusted-key 1
```

```
R2(config)#ntp authenticate
R2(config)#ntp authentication-key 1 md5 jeremysitlab
R2(config)#ntp trusted-key 1
R2(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.2 key 1
```

```
R3(config)#ntp authenticate
R3(config)#ntp authentication-key 1 md5 jeremysitlab
R3(config)#ntp trusted-key 1
R3(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.1 key 1
```

Cours 38 : Domain Name Service (DNS)

Dans ce cours nous verrons le fonctionnement du protocole Domain Name Service (DNS).

DNS est un protocole permettant de rendre l'accès à différentes ressources comme internet plus simple aux humains.

Par exemple le nom de domaine « youtube.com » permet d'accéder à Youtube de manière instantané au d'utiliser l'adresse IP de Youtube. Des noms comme « Youtube.com » ou bien « google.com » sont plus simple à se souvenir plutôt qu'une adresse IP.

Nous verrons tout d'abord le but de DNS et les fonctions basique de DNS. Puis nous verrons comment configurer un DNS sur un IOS Cisco.

Commençons par comprendre l'intérêt d'utiliser le protocole DNS.

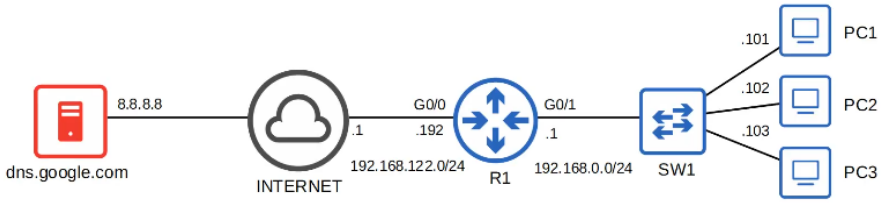
DNS est utilisé pour « résoudre » ou convertir des noms lisible par l'humain (par exemple : Google.com) en adresses IP.

On entre un nom comme Google.com et il est convertit en une adresse IP.

Les machines comme les PC n'utilisent pas de noms, mais des adresses (comme par exemple IPV4/IPV6). Les noms sont aussi plus simple pour nous à utiliser et se souvenir plutôt que des adresses IP. Par exemple comme savoir l'adresse IP de Youtube.com ?

Lorsque l'on écrit « youtube.com » que un navigateur web, l'appareil va demander un serveur DNS pour l'adresse IP de « youtube.com ».

Le serveur DNS que l'appareil utilise peut être configuré manuellement ou appris par DHCP.



Sur un appareil Windows on peut afficher la configuration du serveur DNS avec la commande :

`ipconfig /all` commande lancé depuis le terminal.

```
C:\Users\user>ipconfig /all

[output omitted]

Ethernet adapter ローカル エリア接続：

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82579LM Gigabit Network Connection
   Physical Address. . . . . . . . . : 78-2B-CB-AC-08-67
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.0.101(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
   DNS Servers . . . . . . . . . . . : 8.8.8.8
   NetBIOS over Tcpip. . . . . . . . : Enabled

[output omitted]
```

Ici on peut voir que le serveur DNS est configuré sur l'adresse : 8.8.8.8

Pour afficher comment le serveur DNS fonctionne on lance la commande :

`nslookup youtube.com`

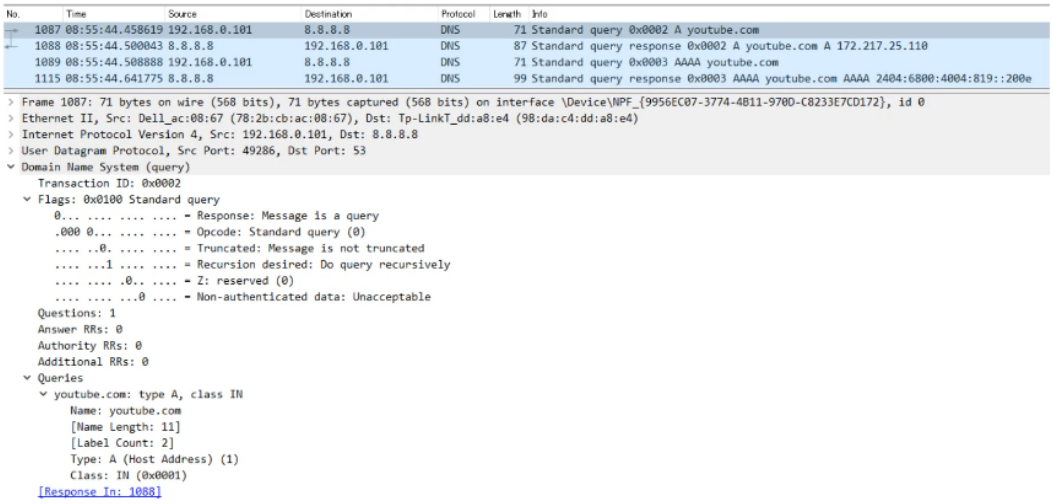
```
C:\Users\user>nslookup youtube.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    youtube.com
Addresses:  2404:6800:4004:819::200e
          172.217.25.110

C:\Users\user>ping youtube.com

Pinging youtube.com [172.217.25.110] with 32 bytes of data:
Reply from 172.217.25.110: bytes=32 time=10ms TTL=117
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117

Ping statistics for 172.217.25.110:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 10ms, Average = 7ms
```

Ici on peut voir que l'adresse IP de Youtube est : 172.217.25.110

Dans le cas du réseau présenté auparavant le PC1 a envoyé la requête au serveur DNS de Google 8.8.8.8 qui a répondu à la requête en transmettant l'adresse IP.

Le routeur R1 ne fonctionne ni comme serveur DNS ou client, il va seulement retransférer les paquets. Aucune configuration DNS n'est requise sur R1.

On peut aussi utiliser Wireshark afin de capturer le trafique à travers la commande nslookup.



Il y a 4 messages :

1. Le PC1 qui fais la demande au serveur DNS de Google

2. Le serveur DNS de Google répond qu'il s'agit d'une réponse à la requête de PC1.

3. Le PC1 renvois ensuite une requête avec les lettre AAAA

4. Le serveur Google répond à la destination de PC1 avec les lettres AAAA

L'enregistrement DNS « A » est utilisé pour cartographier les noms sur des adresses IPV4.

L'enregistrement DNS « AAAA » est utilisé pour cartographier les noms sur des adresses IPV6.

Comme on peut le voir sur la première requête le protocole utilise ici UDP.

Les requêtes et réponses DNS utilisent UDP. TCP est utilisé pour les messages DNS de plus de 512 bytes. Dans la plupart des cas le port 53 est utilisé.

Voyons comment fonctionne le cache DNS.

Les appareils sauvegardent leurs serveur DNS dans un cache DNS local. Cela signifie qu'ils n'ont pas besoin de faire la requête du serveur à chaque fois qu'il veulent avoir accès à une destination particulière.

Pour afficher le cache DNS on lance la commande : *ipconfig /displaydns*

```
C:\Users\user>ipconfig /displaydns

[output omitted]

www.youtube.com
    ----------------------------------------
    Record Name . . . . . : www.youtube.com
    Record Type . . . . . : 5
    Time To Live  . . . . : 98
    Data Length . . . . . : 8
    Section . . . . . . . : Answer
    CNAME Record  . . . . : youtube-ui.l.google.com

[output omitted]

Record Name . . . . . : youtube-ui.l.google.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 98
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 172.217.25.110

[output omitted]
```

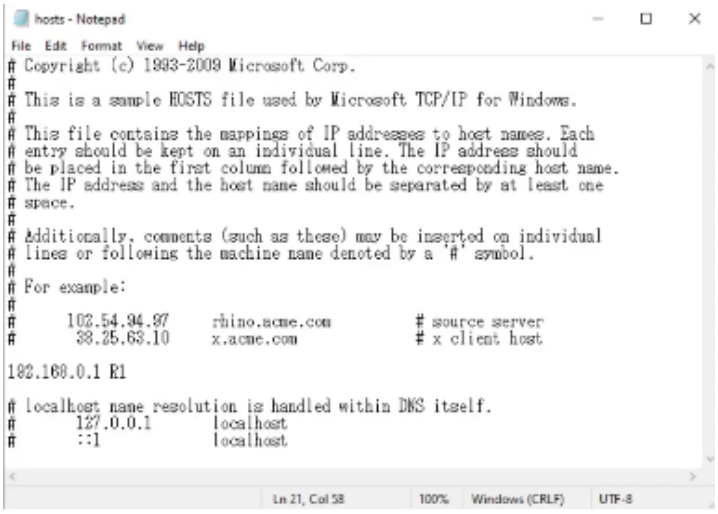
Voici une autre commande qui permet de nettoyer le cache DNS : *ipconfig /flushdns*

```
C:\Users\user>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\user>ipconfig /displaydns

Windows IP Configuration


C:\Users\user>
```

Il est possible d'accéder à la configuration de l'hôte en allant dans le répertoire :

`Windows > System32 > drivers > etc`

Le nom du ficher est hosts. Son contenu ressemble à cela :



Voyons à présent comment configurer un serveur DNS.

Pour les hôtes dans un réseau pour utiliser DNS, il n'est pas nécessaire de configurer le DNS sur un routeur. Ils vont simplement retransférer les messages comme n'importe quelle autre paquet.

Le routeur Cisco peut lui même peut être configuré comme serveur DNS même si cela est rare.

Si un serveur DNS interne est utilisé il s'agit habituellement d'un serveur Windows ou Linux.

Un routeur Cisco peut lui aussi être configuré comme client DNS.

Voici les commandes à utiliser afin de configurer le routeur R1 en tant que serveur DNS :

```
R1(config)#ip dns server

R1(config)#ip host R1 192.168.0.1
R1(config)#ip host PC1 192.168.0.101
R1(config)#ip host PC2 192.168.0.102
R1(config)#ip host PC3 192.168.0.103

R1(config)#ip name-server 8.8.8.8

R1(config)#ip domain lookup
```

On configure une liste de hostname/adresse IP avec : *ip host*

On configure le serveur DNS que R1 va faire la requête si l'enregistrement de la requête n'est pas dans la table d'hôtes.

La commande *ip domain lookup* permet d'activer R1 pour qu'il fasse fonctionner les requêtes DNS.

Disons que le PC1 veut faire un ping de PC2.

On peut voir que le serveur est configuré sur l'adresse IP de R1 :

```
C:\Users\user>ipconfig /all

[output omitted]

   IPv4 Address. . . . . . . . . . . : 192.168.0.101(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
   DNS Servers . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

[output omitted]

C:\Users\user>ping PC2 -n 1

Pinging PC2 [192.168.0.102] with 32 bytes of data:
Reply from 192.168.0.102: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.102:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

On lance le ping avec la commande ping PC2 -n 1

Le PC1 va faire la requête à R1 afin de connaître l'adresse IP de PC2 qui va lui indiquer l'adresse : 192.168.0.102 et le PC1 va ensuite effectuer le ping de l'adresse.

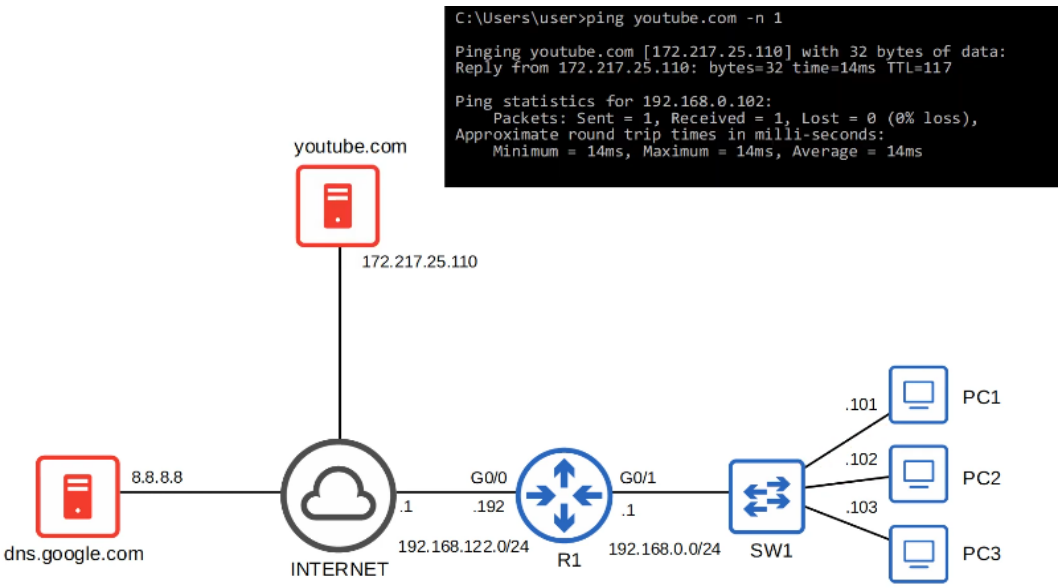
On ajoute le serveur youtube.com dans le réseau précédent.

En faisant un ping vers youtube.com, le PC1 doit connaître son adresse IP, pour cela il va faire la requête au routeur R1. Le routeur n'ayant pas d'entrée pour youtube.com celui ci va faire la requête à son propre serveur DNS qui est celui de Google avec l'adresse IP : 8.8.8.8

Le serveur Google répond en lui donnant l'adresse IP de youtube.com

Le routeur R1 peut à présent répondre au PC1 et lui indiquer l'adresse IP du serveur youtube.

Le PC1 peut à présent faire un ping vers le serveur youtube.



Pour afficher la configuration des hôtes DNS on lance la commande : *show host*

```
R1#show hosts
Default domain is not set
Name/address lookup uses domain service
Name servers are 8.8.8.8

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port  Flags      Age Type   Address(es)
youtube.com               None  (temp, OK)  0   IP    172.217.25.110
R1                        None  (perm, OK)  4   IP    192.168.0.1
PC1                       None  (perm, OK)  1   IP    192.168.0.101
PC2                       None  (perm, OK)  4   IP    192.168.0.102
PC3                       None  (perm, OK)  4   IP    192.168.0.103
```

On peut par exemple voir ci dessus l'enregistrement du DNS de youtube.com sur le routeur R1.

Il y a le flag « temp » pour temporaire qui indique que l'enregistrement est temporaire et qu'il devra être réappris. Lorsque l'enregistrement est fais manuellement il est enregistré de manière permanente.

Afin de configurer un routeur en tant que client DNS on lance les commandes suivantes :

(Le serveur DNS est ici celui de Google)

```
R1(config)#ip name-server 8.8.8.8
```

La commande suivante est normalement lancé par défaut, il est tout de même préférable de la lancer pour être certain qu'elle est bien configuré.

```
R1(config)#ip domain lookup
```

On peut aussi configurer un nom de domaine par défaut avec la commande :

```
R1(config)#ip domain name jeremysitlab.com
```

Le domaine sera appliqué à tous les hostnames qui n'ont pas de domaine spécifié.

Par exemple avec la commande : *ping pc1* deviendra la commande : *ping.pc1.jeremysitlab.com*

```
R1(config)#do ping youtube.com
Translating "youtube.com"
% Unrecognized host or address, or protocol not running.

R1(config)#ip name-server 8.8.8.8

R1(config)#ip domain lookup

R1(config)#do ping youtube.com
Translating "youtube.com"...domain server (8.8.8.8) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.217.25.110, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/13 ms

R1(config)#ip domain name jeremysitlab.com
```

Cours 39 : Dynamic Host Configuration Protocol (DHCP)

Dans ce cours nous verrons le fonctionnement du protocole Dynamic Host Configuration Protocol (DHCP). Tout comme le protocole DNS, le protocole DHCP est important à connaître et est très répandu. DHCP permet aux clients DHCP (comme les appareils de type PC, téléphones, etc...) d'automatiquement apprendre quelle adresse IP utiliser depuis un serveur DHCP au lieu d'avoir à le configurer manuellement.

Dans ce cours nous verrons l'intérêt d'utiliser DHCP, puis les fonctions basique de DHCP et dernier temps comment configurer DHCP sur un IOS Cisco.

DHCP est utilisé pour permettre aux hôtes d'automatiquement/dynamiquement « apprendre » des aspects variés de leurs configuration réseau comme l'adresse IP, le masque de sous réseau, la passerelle par défaut, le serveur DNS, etc...

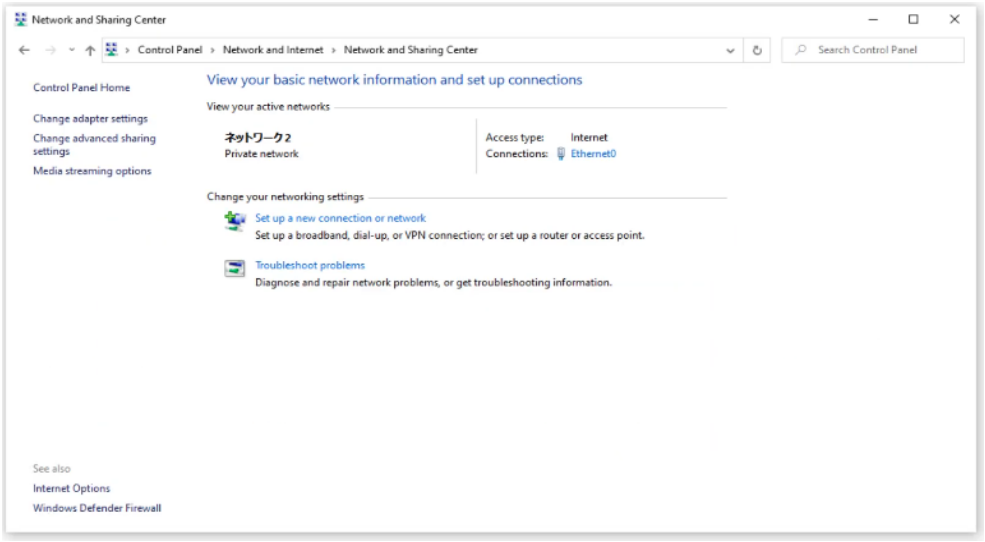
sans avoir à le configurer manuellement ou de manière statique.

C'est une partie important des réseaux modernes. Par exemple lorsque l'on connecte un téléphone/ordinateur au Wifi, le réseau ne demande pas quelle est l'adresse IP, le masque de sous réseau et la passerelle à attribuer à cette appareil, cela ce fais automatiquement grâce à DHCP.

DHCP est principalement utilisé pour des appareils clients comme les PC, téléphones etc...

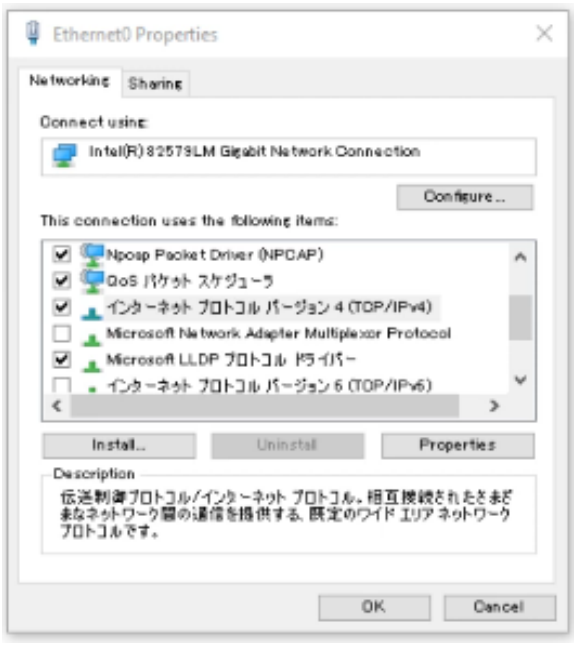
Les appareils comme les routeurs, serveurs, etc.. sont d'habitude configurés manuellement, ils ont une adresse fixe pour bien fonctionner car si la passerelle par défaut change cela ne sera pas idéal.

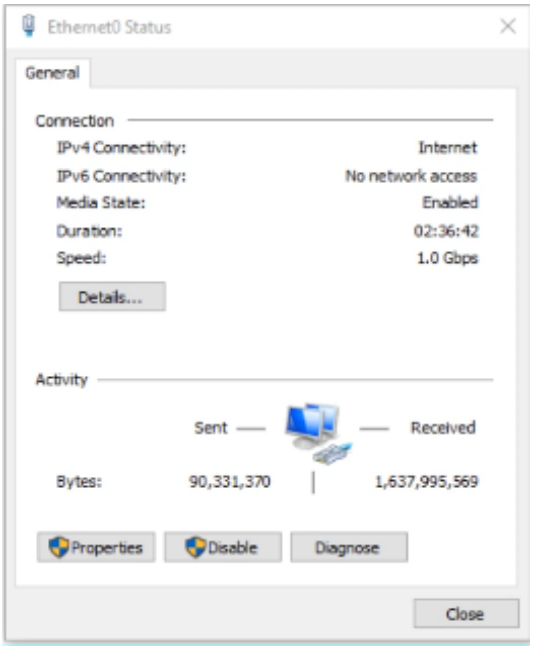
Dans de petits réseaux (comme des réseaux de maison) le routeur fonctionne habituellement comme serveur DHCP pour les hôtes du LAN. Dans de larges réseaux comme les réseaux d'entreprises, le serveur DHCP fonctionne comme serveur Windows/Linux.



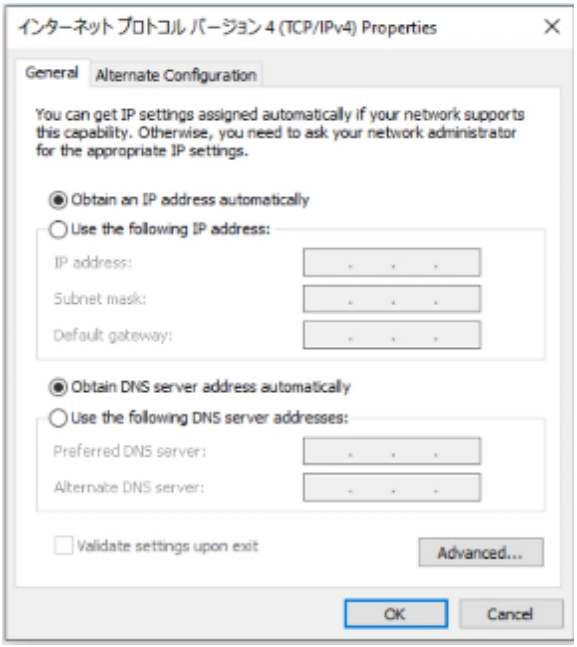
On peut voir ci dessous que la connexion de l'ordinateur est faite à partir du câble Ethernet0

Voici les fenêtres que l'on peut observer lorsque l'on ouvre la configuration :





Lorsque l'option est sur « Obtenir une adresse IP automatiquement » cela signifie que l'appareil va utiliser le protocole DHCP.



On peut observer la configuration du serveur DHCP en lançant la commande : *ipconfig /all*

```
C:\Users\user>ipconfig /all

[output omitted]

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82579LM Gigabit Network Connection
   Physical Address. . . . . . . . . : 78-2B-CB-AC-08-67
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.0.167(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Saturday, January 23, 2021 12:02:04 PM
   Lease Expires . . . . . . . . . . : Saturday, January 23, 2021 2:02:05 PM
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DNS Servers . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

[output omitted]
```

Ici la parenthèse (preferred) indique que celle ci est l'adresse préféré du serveur.

Le serveur DHCP donne des sortes de « bails » et période d'attribution d'une adresse IP.

On peut voir par exemple que le bail obtenu sur la configuration ci dessus à été le Samedi 23 Janvier 2021 à 12 :02 :04 PM et que celui ci expirera le Samedi 23 Janvier 2021 à 2 :02 :05 PM

On peut également obtenir des informations comme la passerelle par défaut, le serveur DHCP ainsi que le serveur DNS.

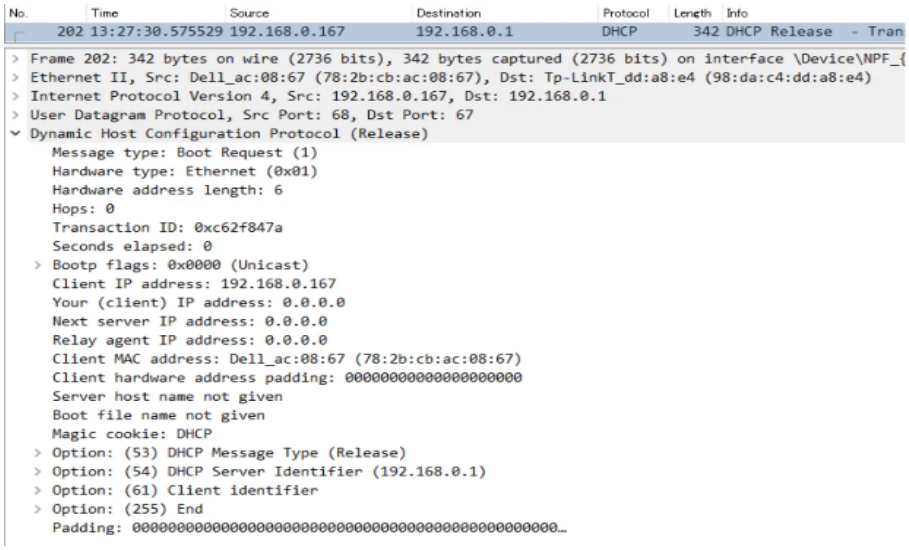
```
C:\Users\user>ipconfig /release

Windows IP Configuration

[output omitted]

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Default Gateway . . . . . . . . . :

[output omitted]
```

Il est possible de renouveler l'attribution de l'adresse IP en lançant la commande :

ipconfig /release

Lorsque la commande est lancé le PC indique au routeur que l'adresse n'est plus utilisé et qu'il est possible de l'assigner à un autre client. En utilisant Wireshark on peut observer l'échange qui est fais entre le PC et le

routeur pour libérer l'adresse :



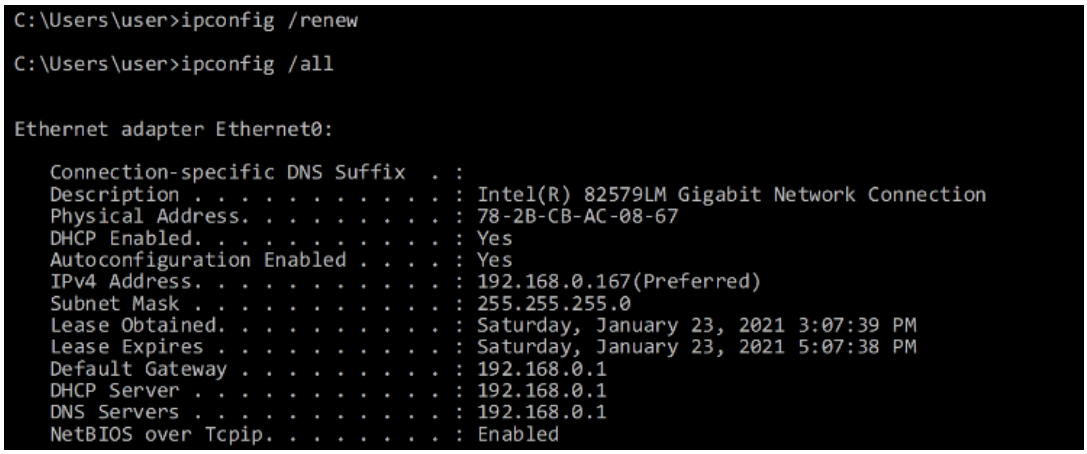
Comme on peut le voir ci dessus, le port source est 68, le port de destination est 67.

Cela est dû au fais que le serveur DHCP utilise le port 67, le client DHCP utilise quant à lui le port 68.

On peut aussi voir les option indiqué dans le message avec l'information de « Release » qui indique que le client a libérer l'adresse.

On utilise la commande : *ipconfig /renew*

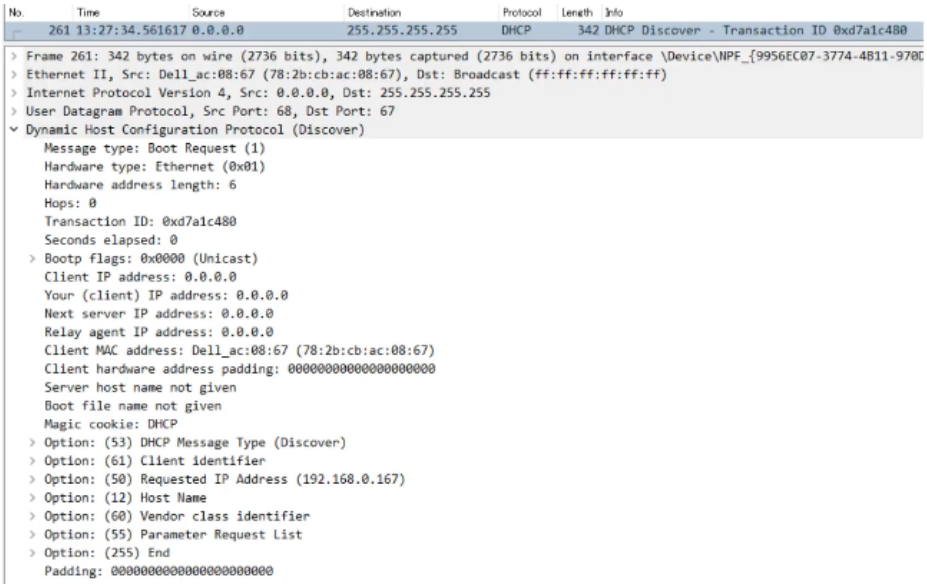
Afin de renouveler l'adresse IP utilisé en une nouvelle.



Cette commande implique 4 messages :

- Le premier message est le DHCP Discover qui est un message en Broadcast du client qui lui permet de savoir s'il existe un serveur DHCP dans le réseau pour qu'il lui attribue une adresse IP.

Avec Wireshark on peut observer le message envoyé par le client :



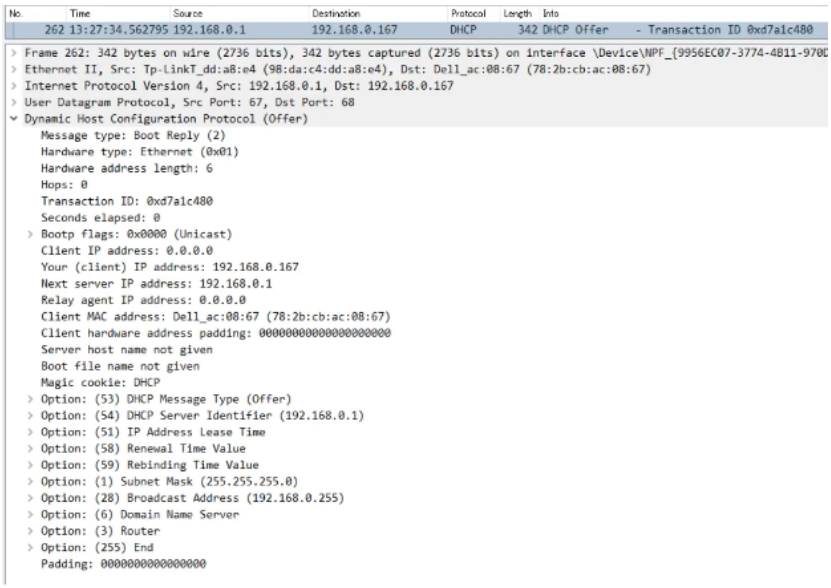
On peut observer la destination qui est en Brodcast.

Dans la section « Options » on peut voir que le type de message envoyé est un message DHCP Discover.

Le PC fais aussi la requête de l'adresse IP 192.168.0.167 puisque auparavant il lui avait aussi été attribué cette même adresse donc il en refais la demande, si l'adresse est disponible elle lui sera attribué sinon une autre adresse IP lui sera attribué.

- Le second message est le DHCP Offer, ce message est envoyé par le serveur DHCP au client afin de lui proposer une adresse IP à utiliser ainsi que d'autres informations comme la passerelle par défaut, le serveur DNS, etc...

Avec Wireshark on peut observer le message envoyé par le serveur :

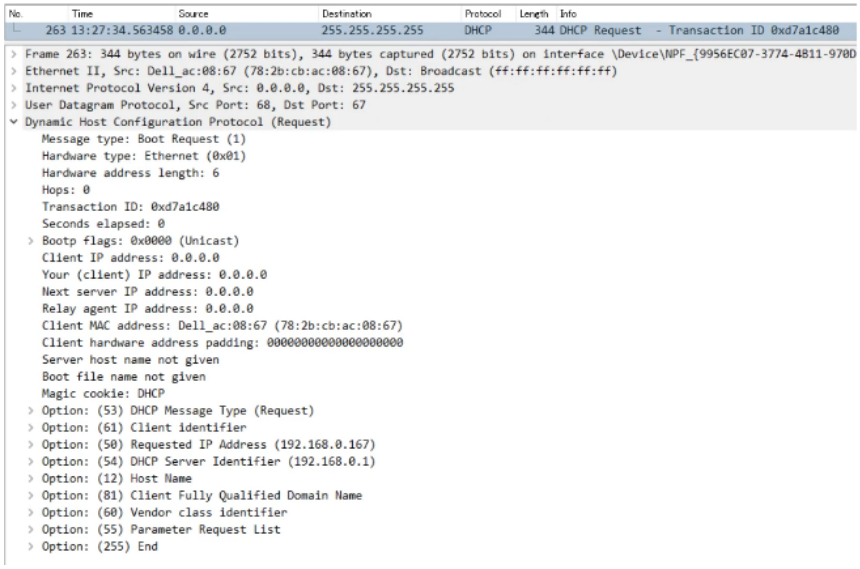


Le message est envoyé en Unicast et peut être envoyé en Broadcast car quelques fois le client peut ne pas réceptionner le message Unicast.

Dans la section « Options » on peut voir que le type de message envoyé est un message DHCP Offer.

- Le troisième message est le DHCP Request, ce message est envoyé par le client DHCP au serveur afin de confirmer l'utilisation de l'adresse IP que celui ci lui a proposé avec le message DHCP Offer.

Avec Wireshark on peut observer le message envoyé par le serveur :

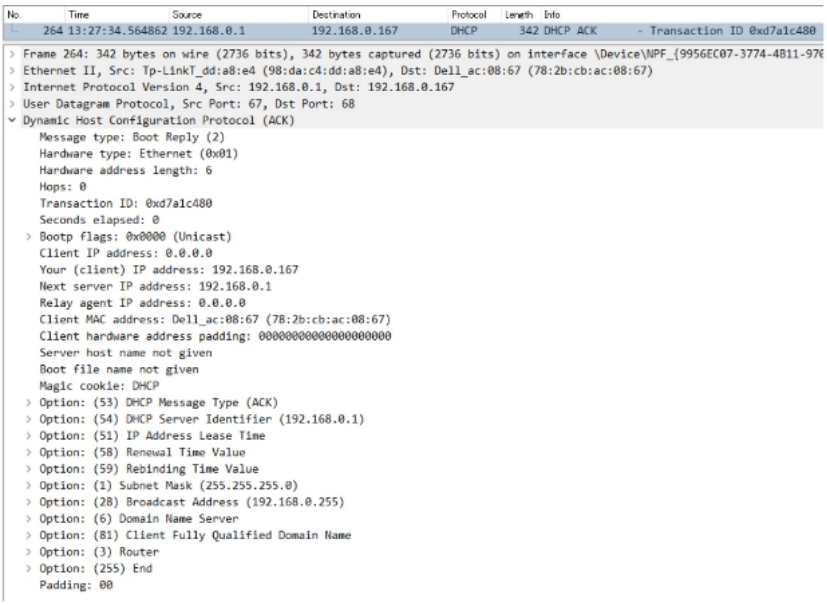


Le message est envoyé en Broadcast.

On peut voir que dans les options le type de message est ici un DHCP request.

Ainsi que l'adresse IP du serveur DHCP.

- Le quatrième et dernier message est un DHCP Ack (Acknowledgment) ou message de confirmation. Ce message est envoyé depuis le routeur au client.



On peut observer le message DHCP réceptionné sur Wireshark :

Ici il s'agit d'un message envoyé en Unicast mais il peut aussi être envoyé en Broadcast.

Dans les options du message on peut observer que le type de message envoyé est de type ACK.

Il est possible de résumer le type de message envoyés avec l'acronyme DORA pour :

- **D**iscover : message de type Broadcast

- **O**ffer : message de type Broadcast ou Unicast

- **R**equest : message de type Unicast

- **A**ck : message de type Broadcast ou Unicast

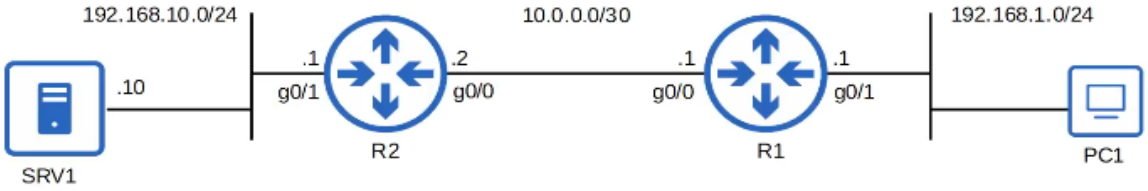
Voyons le principe du relais DHCP, certains ingénieurs réseaux peuvent choisir de configurer chacun des routeurs pour fonctionner comme serveur DHCP pour ses LAN connectés.

De grandes entreprises choisissent souvent d'utiliser un serveur DHCP centralisé.

Si le serveur est centralisé, il ne recevra pas de message DHCP du client en Broadcast (les messages en Broadcast ne quittent pas le sous réseau local)

Pour fixer cela il est possible de configurer un routeur pour fonctionner comme agent relais DHCP.

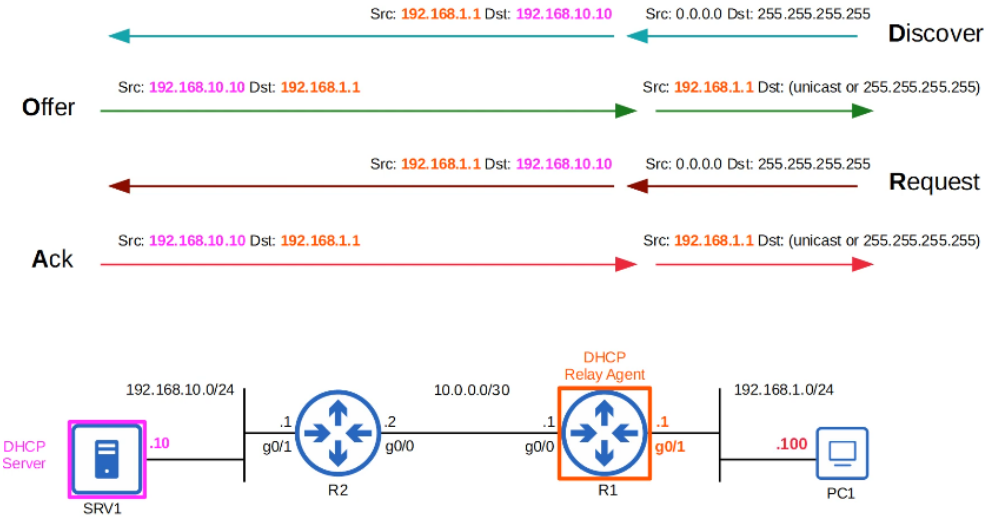
Le routeur redistribuera les messages DHCP du client à un serveur distant comme message Unicast.



Utilisons le réseau suivant afin de démontrer cela visuellement :

SRV1 est un serveur DHCP et R1 est un agent relai DHCP. PC1 distribue un message DHCP Discover pour recevoir une adresse IP, R1 relai le message à SRV1.

Puis SRV1 répond en envoyant un message DHCP Offer vers l'adresse du routeur R1. R1 relai le message vers le PC1 en Unicast ou en Broadcast. Le PC1 répond un DHCP Request en Broadcast au routeur R1 qui lui même relai le message vers le serveur DHCP SRV1. Finalement SRV1 envoie un message DHCP Ack pour confirmer le message à R1 qui relaie le message au PC1.



Voici un schéma pour mieux résumer cela :

Voyons à présent comment configurer un routeur en tant que serveur DHCP dans l'IOS Cisco.

```
R1(config)#ip dhcp excluded-address 192.168.1.1 192.168.1.10

R1(config)#ip dhcp pool LAB_POOL

R1(dhcp-config)#network 192.168.1.0 ?
  /nn or A.B.C.D  Network mask or prefix length
  <cr>

R1(dhcp-config)#network 192.168.1.0 /24

R1(dhcp-config)#dns-server 8.8.8.8

R1(dhcp-config)#domain-name jeremysitlab.com

R1(dhcp-config)#default-router 192.168.1.1

R1(dhcp-config)#lease 0 5 30
```

On utilise les commandes suivantes pour la configuration du Routeur R1 :

On configure tout d'abord un classement d'adresses qui ne seront pas utilisés par les clients DHCP, on utilise pour cela la commande suivante pour exclure les adresses du rang :

ip dhcp excluded-address suivi de la plage d'adresses à exclure.

On utilise ensuite la commande suivante afin de créer un pool DHCP : *ip dhcp pool*

Un pool DHCP est basiquement un nom pour chacun des réseaux que le serveur DHCP va utiliser lors des requêtes effectués par les client. On y configure le serveur DNS, le nom de domaine, etc..

On configure ensuite les sous réseaux qui devront être attribués aux clients (à l'exception des adresses exclus) la commande est : *network* suivi de l'adresse IP.

On configure le serveur DNS avec la commande : *dns-server* suivi de l'adresse du serveur DNS

On spécifie le nom de domaine avec la commande : *domain-name* suivi du nom de domaine

On configure l'adresse de la passerelle par défaut avec la commande : *default-router* suivi de l'adresse du routeur.

On configure ensuite le temps d'allocation avec la commande : *lease* suivi du temps en jours, heures puis minutes ou bien aussi avec la commande : *lease infinite* pour des adresses attribué dans un temps indéfini.

La commande suivante est très utile pour permettre d'afficher les clients qui utilisent le serveur DHCP :

`show ip dhcp binding`

```
R1#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
192.168.1.11        0100.0c29.e727.39       Jan 24 2021 10:52 AM    Automatic
```

On peut voir affiché l'adresse MAC et la date d'expiration de l'attribution de l'adresse.

```
C:\Users\user>ipconfig /all

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : jeremysitlab.com
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection #2
   Physical Address. . . . . . . . . : 00-0C-29-E7-27-39
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.1.11(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Saturday, January 24, 2021 2:22:35 PM
   Lease Expires . . . . . . . . . . : Saturday, January 24, 2021 7:52:35 PM
   Default Gateway . . . . . . . . . : 192.168.1.1
   DHCP Server . . . . . . . . . . . : 192.168.1.1
   DNS Servers . . . . . . . . . . . : 8.8.8.8
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

On affiche la configuration du PC qui est configuré par le serveur DHCP.

On peut voir affiché le nom de domaine, l'adresse IP qui lui a été attribué, ainsi que les dates d'attribution et d'expiration des adresses.

Pour configurer l'agent relais DHCP dans l'IOS Cisco on utilise les commandes suivantes :

ip helper-address suivi de l'adresse IP du serveur DHCP

```
R1(config)#interface g0/1

R1(config-if)#ip helper-address 192.168.10.10

R1(config-if)#do show ip interface g0/1
GigabitEthernet0/1 is up, line protocol is up
  Internet address is 192.168.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is 192.168.10.10

[output omitted]
```

Afin de configurer le routeur pour qu'il soit lui même client DHCP on lance les commandes suivantes sur un routeur R2 : *ip address dhcp*

```
R2(config)#interface g0/1

R2(config-if)#ip address dhcp

R2(config-if)#do sh ip interface g0/1
GigabitEthernet0/1 is up, line protocol is up
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by DHCP

[output omitted]
```

Cours 40 : Simple Network Management Protocol (SNMP)

Dans ce cours nous verrons le fonctionnement de SNMP (Simple Network Management Protocol)

Nous verrons tout d'abord le fonctionnement de SNMP, puis les différentes versions disponibles de SNMP, nous verrons ensuite les différents types de messages que SNMP utilise pour fonctionner.

Nous verrons comment configurer basiquement SNMP.

SNMP est un protocole de type standard industrie et le protocole a été originellement publié en 1988.

Le RFC 1065 structure et identifie la gestion des informations pour TCP/IP basé sur Internet.

Le RFC 1066 permet la gestion des informations de base pour la gestion de base du réseau pour TCP/IP basé sur Internet.

Le RFC 1067 est un protocole de gestion du réseau simple (Simple Network Management Protocole ou SNMP)

Les 3 RFC précédentes forment la version 1 de SNMP.

Le terme Simple dans le nom du protocole ne signifie pas qu'il s'agit d'un protocole simple, la version 1 est même plus compliqué que la dernière version actuelle qui est la version 3.

SNMP peut être utilisé pour suivre le statut d'appareils, faire des changement de configuration etc...

Il y a deux type d'appareils dans SNMP.

1) Les appareils gérés :

- Ce sont les appareils utilisés gérés en utilisant le protocole SNMP.

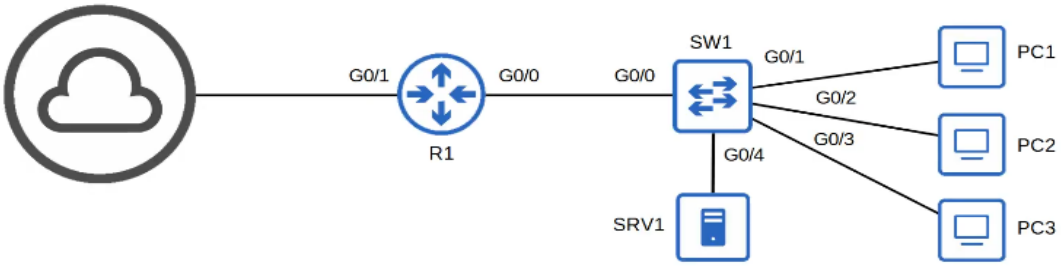
- Par exemple, les appareils du réseau comme les routeurs et commutateurs.

2) Network Management Station (NMS) (en français : Les stations de gestion du réseau)

- Ce sont l'appareil ou les appareils qui gèrent d'autres appareils.

- C'est aussi le « serveur SNMP ».

Voyons comment fonctionne SNMP, pour cela nous utiliserons le réseau suivant :



Le serveur 1 est ici le NMS. Les appareils gérés sont ici le commutateur et le routeur.

Il y a 3 principales opérations en utilisant SNMP.

1) les appareils gérés peuvent notifier le NMS des évènements.

Par exemple la connexion entre le Switch1 et le PC1 est interrompue, le SW1 enverra un message au NMS lui indiquant que la connexion est interrompue.

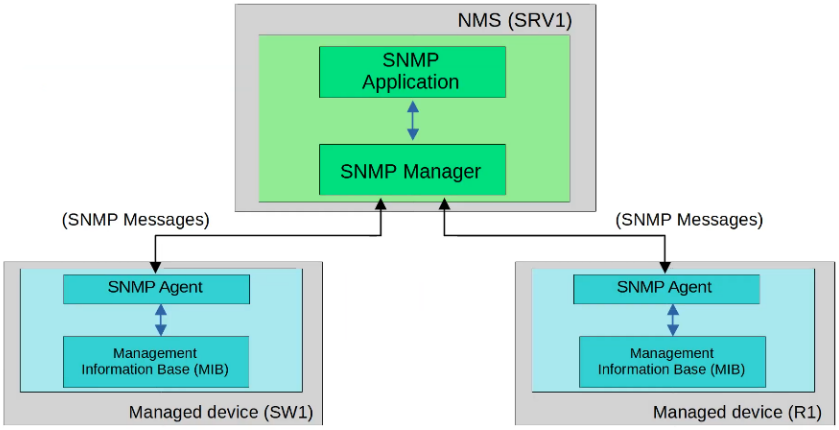
2) Le NMS peut aussi demander aux appareils gérés des informations à propos de leur statut actuel.

Par exemple le NMS peut demander au SW1 ainsi qu'au routeur R1 quelle est son utilisation du CPU. R1 répond en lui indiquant son statut d'utilisation.

3) Le NMS peut demander aux appareils gérés de changer leurs configurations.

Par exemple le NMS peut demander au Routeur R1 de changer son adresse IP pour une autre.

Le routeur R1 répond ensuite au NMS et lui indique que l'adresse IP est bien la nouvelle configuré.



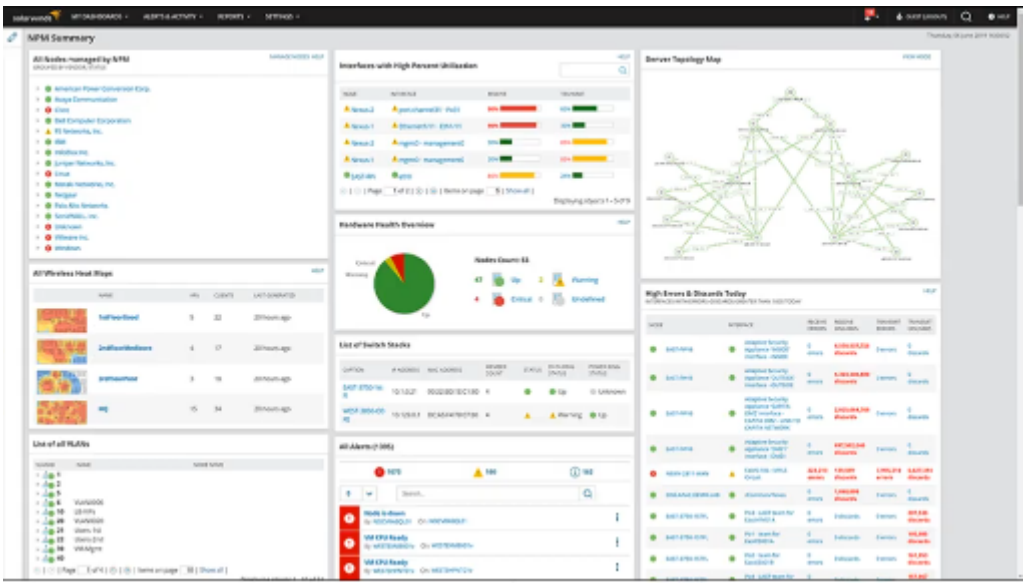
Il est possible de résumer le fonctionnement du réseau avec le schéma ci dessous :

au niveau du NMS ou serveur 1 on peut retrouver le gestionnaire SNMP qui est le logiciel du NMS qui interagit avec l'appareil géré.

Il reçoit des notifications, envois des requêtes pour des informations, envois des changement de configuration, etc...

Il y a aussi l'application SNMP qui fournit une interface pour l'administrateur réseau qui interagit avec. Cela affiche des alertes, des statistiques, des chartes, etc...

Une application SNMP ressemble à cela :



Il y a plusieurs choix d'applications disponibles dans l'utilisation du protocole SNMP, l'application ci dessus provient de SolarWind.

Ensuite au second niveau on peut voir différents appareils qui sont les appareils gérés (SW1 et R1)

Il y a plusieurs entités présentes pour chaque appareil, tout d'abord l'agent SNMP, qui est le logiciel qui se lance dans les appareils gérés et qui interagit avec le gestionnaire SNMP du NMS.

Il envoie/réceptionne des notifications de messages du NMS.

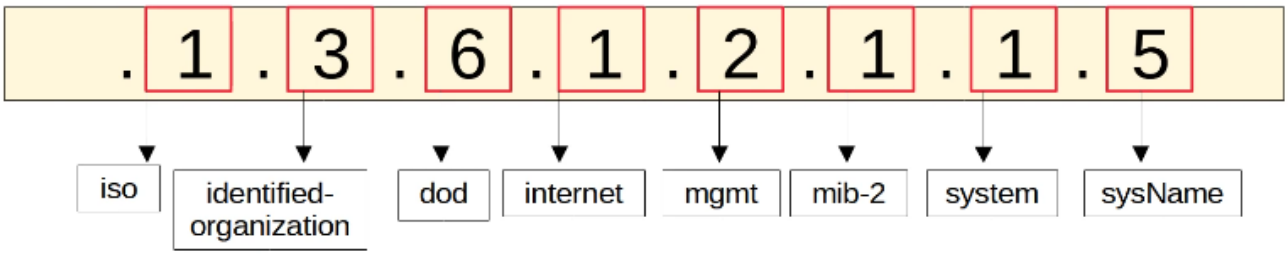
Un autre composant important est le Management Information Base (MIB) c'est la structure qui contient les variables gérés par SNMP.

Chacune de ces variables est identifié avec un ID d'Objet. (OID pour Object ID)

Des exemples de variables sont le statut de l'interface, le trafique, l'utilisation CPU, la température, etc...

Des messages SNMP entre le MIB et le gestionnaire SNMP sont échangés.

Les ID d'objets SNMP sont organisés en une hiérarchie de structure.



Par exemple le NMS peut envoyer un message SNMP à SW1 afin de connaître la valeur de son OID. Le SW1 peut lui répondre que l'OID demandé est SW1.

Il est possible d'avoir plus d'informations concernant l'OID sur le site : oid-info.com

Voyons à présent quelles sont les principales Version qui existent du protocole SNMP.

Plusieurs versions de SNMP ont été proposés ou développé, il existe cependant 3 versions principales qui sont les suivantes :

- SNMPv1 : Il s'agit de la version originale de SNMP

- SNMPv2c : Il permet au NMS de recevoir un grand montant d'informations en une seule requête, donc cela est plus efficace.

La lettre « c » fais référence au « community string » utilisé comme mot de passe dans SNMPv1 (il fallait sélectionner une lettre pour mot de passe), et supprimé de SNMPv2, puis rajouté une nouvelle fois pour SNMPv2c.

- SNMPv3 : Il s'agit de la version la plus sécurisé de SNMP qui supporte des mot de passe crypté et une authentification. Si possible cette version devrait être utilisé.

Voyons à présent le type de messages utilisé :

Classe de message	**Description**	**Messages**
Read	Les messages envoyés par le NMS pour lire des informations depuis les appareils gérés. (Pour savoir par exemple l'utilisation CPU)	Get GetNext GetBulk
Write	Les messages envoyés par le NMS pour changer des informations sur les appareils gérés (par exemple l'adresse IP)	Set
Notification	Les messages envoyés par les appareils gérés pour alerter le NMS d'un évènement particulier	Trap Inform
Response	Les messages envoyés en réponse d'un message/requête précédent	Response

Voyons rapidement en détail chacun de ces messages :

- Get : Il s'agit d'une requête envoyé depuis le gestionnaire à l'agent pour récupérer la valeur de la variable de l'OID ou de plusieurs variables. L'agent envoie ensuite un message « Response » avec le valeur actuelle de chaque variable.

Par exemple le NMS demande quelle est le statut de l'interface G0/1 au Switch. Celui ci répond que l'interface est « Up » ou active.

- GetNext : Il s'agit d'une requête envoyé depuis le gestionnaire vers l'agent pour découvrir les variables disponibles dans le MIB.

Par exemple il demande quelle est le prochain OID, donc ce message peut servir à savoir quelles OID sont disponibles.

- GetBulk : Il s'agit d'une version plus efficace des messages GetNext.

- Set : Il s'agit d'une requête envoyé depuis le gestionnaire vers l'agent pour changer la valeur d'une ou de plusieurs variables. L'agent envoie un message de Response avec la nouvelle valeur.

Par exemple le NMS envoie un message au SW1 lui demandant de changer de nom d'hôte pour SW10 et celui ci répond avec son nouveau nom d'hôte.

- Trap : Il s'agit d'une notification envoyé depuis l'agent vers le gestionnaire. Le gestionnaire n'envoie pas de message de réponse qui confirme qu'il a reçut le Trap, donc ces messages sont « peu fiables ».

Par exemple la connexion entre R1 et SW1 est interrompue, le SW1 envoie un message de Trap au NMS pour lui indiquer que la connexion est interrompu.

- Inform : est un message de notification qui confirme avec un message de réponse.

A l'origine utilisé pour la communication entre plusieurs gestionnaire, mais après des mis à jour cela permet aux agents d'envoyer des messages "Inform aux gestionnaires aussi.

- Response : Il s'agit de messages envoyés en réponse à d'autres.

Les ports utilisés par SNMP sont :

- Agent SNMP : UDP 161

- Gestionnaire SNMP : UDP 162

Voyons comment configurer le protocole SNMPv2c.

Dans le même réseau qu'auparavant on lance les commandes suivantes depuis le routeur R1 :

```
R1(config)#snmp-server contact jeremy@jeremysitlab.com
R1(config)#snmp-server location Jeremy's House

R1(config)#snmp-server community Jeremy1 ro

R1(config)#snmp-server community Jeremy2 rw

R1(config)#snmp-server host 192.168.1.1 version 2c Jeremy1

R1(config)#snmp-server enable traps snmp linkdown linkup
R1(config)#snmp-server enable traps config
```

On configure tout d'abord l'adresse de contact et la localisation de l'appareil, il s'agit de commandes optionnel : ***snmp-server contact*** et ***snmp-server location***

On configure ensuite le SNMP community string (la lettre qui servira de mot de passe) :

snmp-server community Jeremy1 ro signifie read only pour que l'utilisateur Jeremy1 n'envoie pas de messages de type Set.

snmp-server community Jeremy2 rw signifie read/write pour que l'utilisateur Jeremy2 utilise des messages de type Set.

On tape ensuite le mot de passe à utiliser.

On spécifie ensuite l'adresse du NMS avec la commande :

```
snmp-server host 192.168.1.1 version 2c jeremy1
```

Ici l'adresse du serveur NMS est 192.168.1.1 on indique la version SNMP à utiliser ici la version 2c et l'utilisateur pouvant se connecter au NMS, ici l'utilisateur Jeremy1.

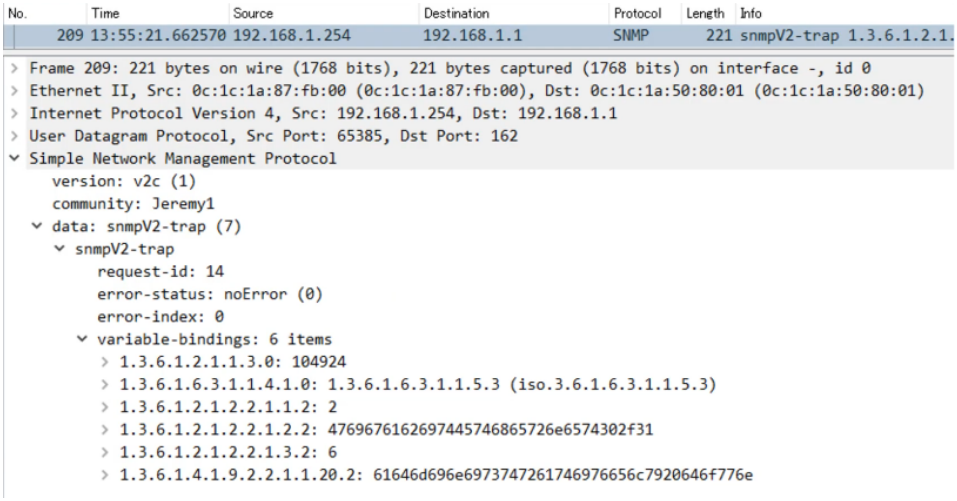
On configure le type de messages Trap qui seront envoyés au NMS :

```
snmp-server enable traps snmp linkdown linkup
```

```
snmp-server enable traps config
```

Dans cette configuration lorsque qu'un appareil change de statut de « up » à « down » un message Trap est envoyé.

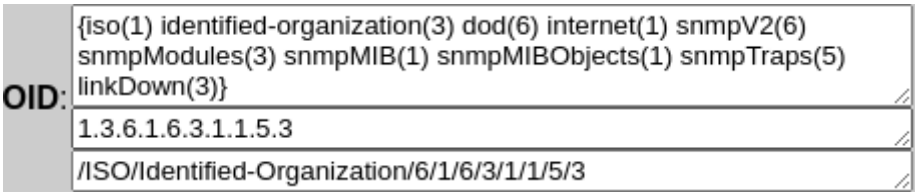
Voyons plus en détail avec Wireshark les messages envoyés lorsque l'interface G0/1 est interrompue.



On peut ici voir les messages OID qui sont envoyés :

Le message OID envoyé est : 1.3.6.1.6.3.1.1.5.3

Si l'on vérifie sur le site : `http://oid-info.com/` on pourra voir que le message correspondant est de type Trap qui permet de notifier le NMS que l'interface est à l'arrêt ou « down » en Anglais.



Sur Wireshark on peut aussi observer la version du protocole utilisé ici c'est la version v2c.

Le « community » est ici Jeremy1 le « community string » avec pour règle ro configuré auparavant.

Il est à préciser qu'avec les protocole SNMPv1 et SNMPv2c il n'y a pas de cryptage. Le contenu et le community sont envoyés en texte clair. Il n'y a pas de sécurité, les paquets peuvent facilement être capturés et lus.

Cours 41 : Syslog

Dans ce cours nous verrons le fonctionnement de Syslog qui est un protocole utilisé pour garder les différents évènement ou logs qui se passent dans un appareil. Par exemple lorsque les interfaces qui s'allument ou s'éteignent, la relation avec les voisins OSPF, etc...

Les message d'évènements peuvent apparaître en temps réel dans le CLI de l'appareil pour informer de l'importance des évènements qui arrivent et ils peuvent aussi être stockés dans l'appareil ou dans un serveur externe pour être examiné plus tard. Ces logs sont très importantes donc comprendre Syslog est essentiel pour des administrateurs réseau et ingénieurs.

Nous verrons tout d'abord comment fonctionne Syslog, le format des messages utilisés par Syslog, puis les facilités Syslog et les niveaux de sévérités et pour finir la configuration de Syslog.

Syslog est un protocole de l'industrie Standard pour la journalisation d'évènements de message.

Sur des appareils réseau, Syslog peut être utilisé pour enregistrer des évènements comme le statut de changement d'interfaces (marche/arrêt), le changement dans le statut des voisins OSPF (marche/arrêt), les redémarrage système, etc...

Les messages peuvent aussi apparaître dans le CLI, sauvegardés dans la RAM de l'appareil, ou envoyé vers des serveurs Syslog externes.

Par exemple en utilisant la commande : ***no shutdown*** on peut voir les messages apparaître

```
R1(config)#int g0/0
R1(config-if)#no shutdown
R1(config-if)#
*Feb 11 03:02:55.304: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Feb 11 03:02:56.305: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
```

La journalisation est importante lorsque l'on cherche à résoudre un problème, ou examiner les causes d'un incident, etc...

Syslog et SNMP peuvent tout deux être utilisés pour suivre et résoudre les problèmes d'un appareil. Ils sont complémentaires, mais leurs fonctionnalités sont différentes.

Voyons le format des messages de type Syslog :

```
seq:time stamp: %facility-severity-MNEMONIC:description
```

On peut voir plusieurs parties dans le message :

- seq : le numéro de séquence indique l'ordre de séquence des messages.

- time : le temps indique l'heure à laquelle le message a été généré.

- %facility : est une valeur qui indique quelle processus de l'appareil a généré le message.

- severity : est un nombre qui indique la sévérité de l'évènement enregistré. Il y a 8 niveaux de sévérité.

- MNEMONIC : c'est un code court pour le message qui indique ce qu'il se passe.

- description : détaille des informations à propos des évènements reportés.

Voici dans un tableau plus détaillé les différents niveau de sévérité de Syslog :

Niveau	Mot clé	Description
0	Emergency	Système inutilisable
1	Alert	Une action doit être faite immédiatement
2	Critical	Conditions critiques
3	Error	Conditions d'erreurs
4	Warning	Conditions d'avertissement

5	Notice	Normal mais condition signifiante (Notification)
6	Informationnel	Messages d'information
7	Debugging	Message de niveau de débuggage

Le RFC n'a pas défini de manière définitive les niveau de sévérité donc chaque vendeur peut avoir ses propres normes dans les différents niveaux de sévérités.

Le RFC 5424 (Protocole Syslog) explique : « Because severities are very subjective, a relay or collector should not assume that all originators have the same definition of severity. »

en Français : « Parce que les sévérités sont très subjectives, un relais ou un collecteur ne doit pas supposer que tous les initiateurs ont la même définition de la gravité. »

Plus simplement cela signifie que l'on ne peut pas s'attendre à ce que un message de « Warning » dans un routeur Cisco ait la même signification qu'un routeur Juniper ayant un message au niveau « Warning » chaque vendeur interprète chaque niveau différemment.

Voyons plusieurs exemple de messages Syslog :

```
*Feb 11 03:02:55.304: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
```

On peut voir d'abord la date et l'heure à laquelle le message est généré, le 11 Février à 3 :02 :55.304, il n'y a pas ici de numéro de séquence avant le temps.

Le facility, est « LINK » et le niveau de sévérité est 3, le MNEMONIC est « UPDOWN » qui signifie que l'interface a changé d'état.

La description nous dis ce qu'il se passe. L'interface Gigabitethernet0/0 à changé d'état pour passer à marche.

Voici un message de type OSPF qui est passé à l'état de FULL.

```
*Feb 11 05:04:39.606: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.2 on GigabitEthernet0/0 from
LOADING to FULL, Loading Done
```

Le Facility est OSPF, le niveau de sévérité est de 5, le MNEMONIC est ADJCHG pour Adjancency change.

Voici un autre message dans lequel la séquence de numéro est activé.

```
000043: *Feb 11 05:06:43.331: %SYS-5-CONFIG_I: Configured from console by jeremy on console
```

Dans ce cas le facility est SYS pour le système, le sévérité est 5 (notification), le MNEMONIC est CONFIG_I, avec une description : « configured from console by jeremy on console »

On reçoit ce type de message lorsque l'on quitte le mode global config et que l'on retourne au mode privilégié.

Un dernier type de message dans lequel le temps a été changé pour passer de UTC à JST

```
*Feb 11 07:27:23.346: %SYS-6-CLOCKUPDATE: System clock has been updated from 07:27:23 UTC Thu Feb
11 2021 to 16:27:23 JST Thu Feb 11 2021, configured from console by jeremy on console.
```

Le facility est SYS, le niveau de sévérité est cette fois 6 (Informational). Le MNEMONIC est CLOCKUPDATE.

Voici un résumé des différentes localisations dans lesquels des messages Syslog peuvent être affichés.

- La ligne de console : Les messages Syslog s'affichent dans le CLI lorsque l'appareil est connecté par le port console. Par défaut, tous les messages (niveau 0 à 7) apparaissent.

- La ligne VTY : les messages Syslog apparaissent dans le CLI lorsque connecté par Telnet/SSH. Ces messages sont désactivés par défaut.

- Mémoire Tampon : Les messages Syslog sont sauvegardés dans la RAM. Par défaut, tous les messages (Niveau 0 à 7) sont affichés.

On peut voir les messages avec la commande : *`show logging`*

- Serveur externe : On peut configurer l'appareil pour envoyer des messages Syslog vers un serveur externe.

Les serveurs Syslog écoutent les messages sur le port UDP 514.

Voici les commandes pour configurer Syslog, on peut configurer le niveau avec le chiffre correspondant (par exemple pour un niveau 6 le chiffre *6* ou bien le mot clé *informational*).

pour configurer la ligne de console on lance la commande :

```
R1(config)#logging console 6
```

Pour configurer la ligne VTY on lance la commande :

```
R1(config)#logging monitor informational
```

Pour configurer la mémoire tampon on lance la commande suivante, on configure la taille du tampon en bytes puis le niveau de sévérité en chiffre ou avec le mot clé.

```
R1(config)#logging buffered 8192 6
```

Pour configurer un serveur externe on lance les commandes suivantes, pour la dernière commande le niveau est indiqué avec le mot clé ou bien le chiffre.

```
R1(config)#logging 192.168.1.100
R1(config)#logging host 192.168.1.100
R1(config)#logging trap debugging
```

Même si *`logging monitor`* est activé par défaut, les messages Syslog n'apparaîtront pas pas lorsque connecté par Telnet ou SSH.

Pour afficher les messages il faut pour cela utiliser la commande :

```
R1#terminal monitor
```

Cette commande doit être utilisé chaque fois que l'on se connecte à un appareil par Telnet ou SSH

Par défaut les messages d'évènements sont affichés dans le CLI alors que l'on est entrain de taper une commande, par exemple ici on peut voir que le message apparaît alors que la commande n'a pas été terminé.

```
R1(config)#exit
R1#show ip in
*Feb 11 09:38:41.607: %SYS-5-CONFIG_I: Configured from console by jeremy on
consoleterface brief
```

Pour empêcher cela on peut utiliser les commandes :

```
R1(config)#line console 0
R1(config-line)#logging synchronous
```

Cela causera qu'une nouvelle ligne s'affiche si l'écriture est interrompue par un message par exemple :

```
R1(config)#exit
R1#show ip int
*Feb 11 09:41:00.554: %SYS-5-CONFIG_I: Configured from console by jeremy on console
R1#show ip int
```

Il est possible de configurer le temps avec les commandes :

Pour configurer l'heure en fonction de la date et de l'heure on lance la commande :

```
R1(config)#service timestamps log datetime
```

Pour configurer l'heure en fonction de l'heure de lancement du système on lance la commande :

```
R1(config)#service timestamps log uptime
```

On active les numéros de séquence avec la commande :

```
R1(config)#service sequence-numbers
```

Syslog et SNMP sont tout deux utilisés pour le monitoring et la résolution de problèmes d'appareils.

Ils sont complémentaires, mais leurs fonctionnalités sont différentes.

Syslog est utilisé pour la journalisation des messages.

- Les évènements qui concernent le système sont catégorisés basé sur la facility/sévérité et journalisation.

- Sont utilisés pour la gestion du système, l'analyse et la résolution.

- Les messages sont envoyés depuis les appareils au serveur. Le serveur ne peut pas activement retirer d'information depuis les appareils (comme le Get SNMP) ou modifier les variables (comme le Set SNMP)

SNMP est utilisé pour récupérer et organiser des informations à propos des appareils SNMP gérés.

- Les adresses IP, le statut actuelle de l'interface, la température, l'utilisation du CPU, etc...

- Les serveurs SNMP peuvent utiliser Get pour requêter les clients et utiliser Set pour modifier les variables des clients.

Cours 42 : Secure Shell (SSH)

Dans ce cours nous verrons le fonctionnement du protocole Secure Shell (SSH) qui est utilisé pour se connecter à un appareil et le configurer par la ligne de commande (CLI). Une option pour se connecter à un appareil et le configurer est par le moyen d'un port console. Il est aussi possible de se connecter à un appareil à distance avec l'adresse IP par le moyen de SSH.

Dans ce cours nous verrons le fonctionnement de console port security, puis nous verrons ce qu'est la couche 2 commutateur de gestion IP qui ne route pas les paquets et ne construit pas de table de routage. Il est tout de même possible de configurer et gérer une adresse IP de gestion pour ces appareils pour pouvoir y accéder à distance.

Nous verrons ensuite le fonctionnement de Telnet qui est un protocole similaire à SSH.

Et nous verrons en dernier temps le fonctionnement de SSH.

Tout d'abord voyons le fonctionnement de console port security. Par défaut aucun mot de passe n'est requis pour accéder au CLI d'un appareil Cisco IOS par le port console, il est possible de configurer un mot de passe sur la ligne de console. Après cela l'utilisateur devra entrer un mot de passe pour accéder au CLI par le port console.

Pour cela il faut configurer l'appareil avec les commandes suivantes :

```
R1(config)#line console 0

R1(config-line)#password ccna

R1(config-line)#login

R1(config-line)#end
R1#exit


R1 con0 is now available

Press RETURN to get started.

User Access Verification

Password:

R1>
```

- *`line console 0`* : sert à se connecter à la ligne de console puisqu'il n'y a qu'une seule ligne de console ou en d'autre terme qu'il n'est possible pas qu'il y ait qu'une seule connexion à la fois le numéro sera donc toujours 0 (à part dans le cas ou il est possible de connecter plusieurs utilisateurs en ligne de console)

- *`password ccna`* : sert à configurer le mot de passe « ccna »

- *`login`* : indique à l'appareil qu'il est requis que l'utilisateur entre le mot de passe configuré pour accéder au CLI par le port console.

Avec cette configuration un mot de passe est à présent requis pour pouvoir se connecter à la ligne de commande de l'appareil.

On remarque que lorsque l'on écrit le mot de passe lors de la connexion il n'apparaît pas en clair sur celui ci, cela permet à ce qu'il ne soit pas visible pour plus de sécurité.

Alternativement il est possible de configurer la ligne de console pour que l'utilisateur se connecte avec l'un des noms utilisateurs configurés sur l'appareil.

On utilise pour cela les commandes suivante :

```
R1(config)#username jeremy secret ccnp

R1(config)#line console 0

R1(config-line)#login local

R1(config-line)#end
R1#exit



R1 con0 is now available


Press RETURN to get started.

User Access Verification

Username: jeremy
Password:
R1>
```

- *username jeremy secret ccnp* : sert à créer l'utilisateur jeremy et pour mot de passe associé ccnp

- *line console 0* : sert à configurer la ligne de console comme expliqué auparavant

- *login local* : indique à l'appareil qu'il est requis que l'utilisateur entre le nom d'utilisateur et mot de passe configurés localement pour accéder au CLI par le port console.

On peut aussi configurer l'appareil avec les commandes suivantes pour configurer son interface :

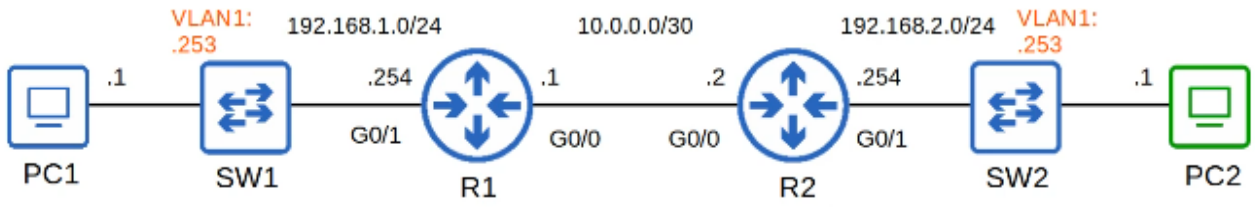
```
line con 0
 exec-timeout 3 30
 password ccna
 logging synchronous
 login local
```

- *exec-timeout 3 30* : permet de déconnecter l'utilisateur après 3 minutes et 30 secondes d'inactivité.

Avec cette configuration il est requis pour l'appareil d'utiliser un nom d'utilisateur et un mot de passe pour se connecter au CLI puisque la commande login local est lancé en priorité par rapport à une connexion direct en utilisant un simple mot de passe.

Voyons à présent le fonctionnement de la couche 2 du Switch et de la gestion des IP. Cette couche ne fais pas fonctionner le routage et ne construit pas de table de routage. Elles ne font pas de routage pour les IP. Il est possible d'assigner une adresse IP à un SVI pour permettre de se connecter à distance au CLI du Switch (en utilisant Telnet ou SSH)

Pour démontrer les différents exemples nous utiliserons le réseau suivant :



On commence par configurer les adresses IP et les vlan avec les commandes suivantes :

```
SW1(config)#interface vlan1
SW1(config-if)#ip address 192.168.1.253 255.255.255.0
SW1(config-if)#no shutdown
SW1(config-if)#exit

SW1(config)#ip default-gateway 192.168.1.254
```

On configure tout d'abord l'adresse IP sur le SVI de la même manière qu'un switch Multicouche et on active l'interface si nécessaire.

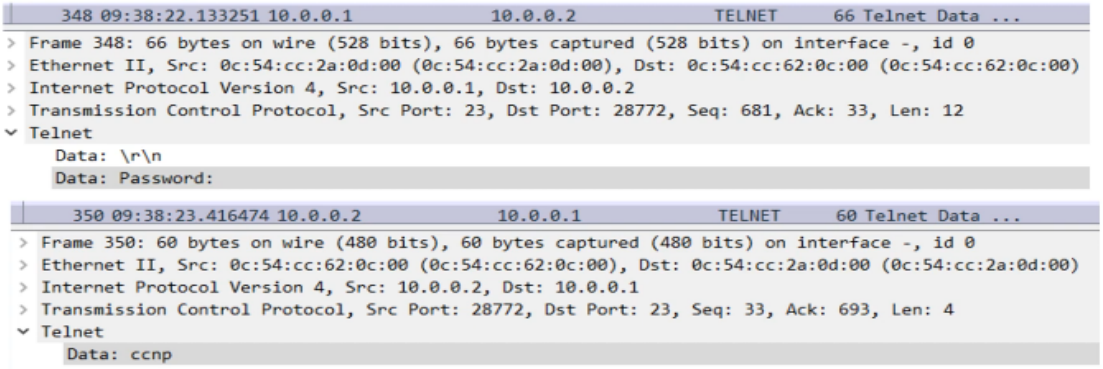
On configure ensuite la passerelle du switch par défaut. Dans ce cas le PC2 n'est pas dans la même LAN que SW1. Si SW1 n'a pas de passerelle par défaut il ne pourra pas communiquer avec le PC2.

Voyons le fonctionnement de Telnet, il s'agit d'un protocole plus très utilisé puisque pas très sécurisé mais il est bien de le connaître avant de voir SSH.

Telnet (Teletype Network) est un protocole utilisé pour accéder à distance au CLI d'un hôte distant.

Telnet a été développé en 1969 et a été largement remplacé par SSH qui est plus sécurisé.

SSH a quant à lui été développé en 1995, Telnet envoie des données en texte clair dans cryptage.



Lorsque l'on utilise Telnet on peut capturer les paquets avec Wireshark :

On voit que dans la capture on peut identifier le mot de passe qui n'est pas crypté et qui est utilisé par Telnet, le mot de passe entré est ici ccnp.

Le serveur Telnet auquel l'appareil essaie de se connecter écoute pour Telnet sur le port TCP 23.

Voici les commandes utilisés pour configurer Telnet sur le SW1 :

```
SW1(config)#enable secret ccna
SW1(config)#username jeremy secret ccna
SW1(config)#access-list 1 permit host 192.168.2.1

SW1(config)#line vty 0 15

SW1(config-line)#login local

SW1(config-line)#exec-timeout 5 0

SW1(config-line)#transport input telnet

SW1(config-line)#access-class 1 in
```

- *enable secret ccna* : avec cette commande si un mot de passe n'est pas configuré, il ne sera pas possible de se connecter en mode privilèged exec mode en utilisant Telnet.

- *username jeremy secret ccna* : sert à configurer un nom d'utilisateur/mot de passe

- *access-list 1 permit host 192.168.2.1* : permet de configurer un ACL pour limiter quelle appareil peut se connecter à la ligne VTY

- *line vty 0 15* : l'accès Telnet/SSH est configuré sur la ligne VTY. Il y a 16 lignes disponible, donc jusqu'à 16 utilisateurs peuvent se connecter en même temps (VTY est l'acronyme de Virtual TeleType)

- *login local* : sert à ce que l'utilisateur se connecte uniquement par le moyen d'une connexion avec un nom d'utilisateur/mot de passe local.

- *exec-timeout 5 0* : sert à configurer le délai de déconnexion à 5 minute en période d'inactivité.

- *transport input telnet* : permet une connexion uniquement par Telnet

il y a d'autres possibilité de commande pour autoriser d'autres type de connexion par exemple :

- *transport input ssh* : permet de n'autoriser que les connexion SSH

- *transport input telnet ssh* : permet d'autoriser les deux

- *transport input all* : permet tout type de connexion

- *transport input none* : ne permet aucune connexion

- *access-class 1 in* : Applique les ACL sur la ligne VTY,

on peut aussi configurer l'ACL sur la ligne VTY avec la commande : *access-class* qui applique une ACL sur la ligne VTY, la commande : *ip access-group* applique une ACL sur une interface.

```
R2#ping 192.168.1.253
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.253, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/11/16 ms

R2#telnet 192.168.1.253
Trying 192.168.1.253 ...
% Connection refused by remote host
```

Lorsque l'on essaye de ping le SW1 avec R2 on peut voir que le ping fonctionne, cependant lorsque l'on tente de se connecter en utilisant Telnet on obtient un message d'erreur qui indique que la connexion est refusé cela est dû à l'ACL configuré sur la ligne VTY. Avec cette ACL seulement le PC 2 peut se connecter en Telnet au SW1, la connexion fonctionne bien avec PC2 :

```
C:\Users\user>telnet 192.168.1.253
Connecting To 192.168.0.1…

User Access Verification

Username: jeremy
Password:
SW1>
```

La ligne VTY est configuré de la manière suivante sur l'appareil :

```
line vty 0 4
 access-class 1 in
 exec-timeout 5 0
 login local
 transport input telnet
line vty 5 15
 access-class 1 in
 exec-timeout 5 0
 login local
 transport input telnet
```

Ici jusqu'à 5 connexion en simultanés sont permises.

Voyons à présent le fonctionnement de SSH.

SSH (Secure Shell) a été développé en 1995 pour remplacer les protocoles moins sécurisés comme Telnet.

Voici une définition de Shell donnée par Wikipédia :

« L'interface en ligne de commande (CLI, de l'anglais « command line interface ») permet à l'utilisateur d'interagir avec le système à partir de commandes qui sont adaptées au mode texte et qui permettent, entre autres, l'exécution d'applications affichées à l'origine (dans un système moderne, l'environnement graphique est aussi pris en compte) dans un environnement en mode texte (TUI pour Text User Interface) ;

La coque logicielle de type graphique fournit à l'utilisateur un environnement graphique (GUI, pour graphical user interface), généralement un environnement de bureau ou un écran d'accueil.. »

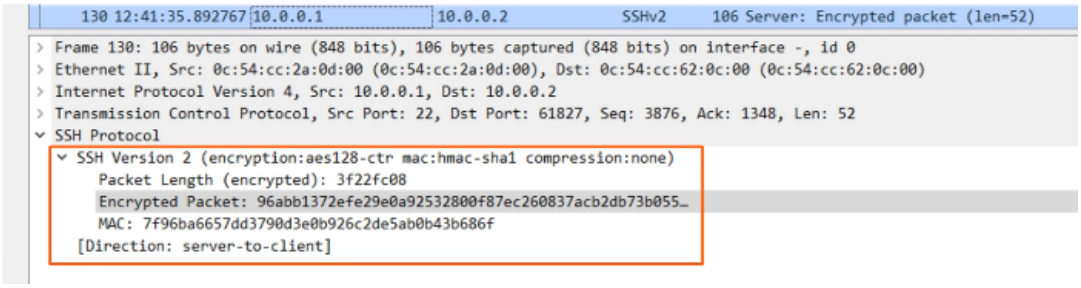
Donc à chaque fois qu'un utilisateur se connecte à une ligne de commande il utilise un Shell.

SSHv2 a la version majeur de révision de SSHv1 et a été publié en 2006.

La version 2 est plus sécurisé et devrait être utilisé le plus souvent possible.

Si un appareil supporte les versions 1 et 2, il dit de lancer la « version 1.99 », ça n'est pas une version de SSH mais cela signifie juste que l'appareil supporte les versions 1 et 2.

SSH fournit des fonctionnalités de sécurité comme le cryptage des données et l'authentification.



Voici l'exemple d'une capture Wireshark par SSH

Le paquet crypté est composé de caractère, seulement le serveur SSH et le client ont la clé pour déchiffrer le paquet. Le serveur SSH écoute le trafique sur le port 22.

Avant de configurer SSH, il faut vérifier si l'OS de l'appareil supporte SSH.

```
SW1#show version
Cisco IOS Software, vios_l2 Software (vios_l2-ADVENTERPRISEK9-M), Version 15.2(4.0.55)E, TEST
ENGINEERING ESTG_WEEKLY BUILD, synced to  END_OF_FLO_ISP
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Tue 28-Jul-15 18:52 by sasyamal

SW1#show ip ssh
SSH Disabled - version 1.99
%Please create RSA keys to enable SSH (and of atleast 768 bits for SSH v2).
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa
Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa
Encryption Algorithms:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
MAC Algorithms:hmac-sha1,hmac-sha1-96
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded): NONE
```

Pour cela on lance les commandes :

- *show version* : pour afficher la version de l'OS, on peut voir K9 à la fin (en bleu), les machines et OS qui supportent SSH ont K9 dans leurs noms. Cisco exporte des NPE (No payload Encryption) des images IOS aux pays qui ont des restrictions sur le chiffrement des technologies.

Les images IOS NPE ne supportent pas les fonctionnalités de chiffrage comme SSH.

- *show ip ssh* : permet aussi d'afficher si SSH est supporté par l'appareil. Dans notre cas on peut voir la version de SSH, mais que SSH est désactivé.

On peut voir ici le message : « please create RSA keys to enable SSH (and of atleast 768 bits for SSH v2). Il s'agit de clés cryptographique qui permettent des fonctions essentiels dans la configuration de SSH.

Une fois qu'à été vérifié si l'appareil supporte SSH, on commence par configurer SSH, pour cela il faut générer une pair de clé publique RSA et une pair de clé privée RSA.

Les clés sont utilisés pour le cryptage et le décryptage des données, l'authentification, etc...

```
SW1(config)#ip domain name jeremysitlab.com

SW1(config)#crypto key generate rsa
The name for the keys will be: SW1.jeremysitlab.com
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

SW1(config)#
*Feb 21 04:22:35.778: %SSH-5-ENABLED: SSH 1.99 has been enabled

SW1(config)#do show ip ssh
SSH Enabled - version 1.99
Authentication methods:publickey,keyboard-interactive,password
Encryption Algorithms:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
MAC Algorithms:hmac-sha1,hmac-sha1-96
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded): SW1.jeremysitlab.com
[output omitted]
```

Voici les commandes à utiliser pour faire cela :

- *ip domain name jeremysitlab.com* : sert à configurer le nom de domaine. Le FQDN de l'appareil est utilisé pour nommer la clé RSA. Le FQDN est l'acronyme de : Fully Qualified Domain Name (Hostname + Nom de domaine)

- *crypto key generate rsa* : sert à générer la clé RSA, ici le nom de la clé est configuré automatiquement c'est SW1.jeremysitlab.com qui est aussi le FQDN de SW1

on choisi ensuite la taille du modulus ou taille de clé en bit. Ici 2048 bits.

On peut aussi utiliser la commande : *crypto key generate rsa modulus* suivie de la longueur de clé pour configurer directement en 1 commande la longueur de clé généré.

Une fois la clé généré un message apparaît indiquant que SSH est à présent activé.

Lorsque l'on vérifie le statut de SSH avec la commande : *show ip ssh* on peut voir à présent qu'il est activé.

A présent que SSH est activé voyons les commandes à utiliser pour le configurer :

```
SW1(config)#enable secret ccna

SW1(config)#username jeremy secret ccna

SW1(config)#access-list 1 permit host 192.168.2.1

SW1(config)#ip ssh version 2

SW1(config)#line vty 0 15

SW1(config-line)#login local

SW1(config-line)#exec-timeout 5 0

SW1(config-line)#transport input ssh

SW1(config-line)#access-class 1 in
```

- *enable secret ccna* : avec cette commande si un mot de passe n'est pas configuré, il ne sera pas possible de se connecter en mode privilèged exec mode en utilisant Telnet.

- *username jeremy secret ccna* : sert à configurer un nom d'utilisateur/mot de passe local, ici jeremy et mot de passe ccna

- *access-list 1 permit host 192.168.2.1* : permet de configurer une ACL pour qu'il autorise seulement l'hôte : 192.168.2.1

- *ip ssh version 2* : est optionnel mais recommandé permet de restreindre SSH seulement à la version 2.

- *line vty 0 15* : permet de configurer toutes les lignes VTY tout comme Telnet.

- login local : permet d'activer l'authentification local, il n'est pas possible d'utiliser *login* pour SSH seulement *login local* fonctionne.

- *exec-timeout 5 0* : permet de configurer le exec timeout

- *transport input ssh* : est la meilleur pratique et permet de limiter les lignes de connexion VTY seulement pour SSH.

- *access-class 1 in* : est optionnel mais recommandé permet d'appliquer l'ACL pour restreindre la ligne de connexion VTY.

Voici les différentes étapes de configuration :

1) configurer le nom d'hôte

2) Configurer le DNS nom de domaine

3) Générer la pair de clé RSA

4) Configurer et activer le mot de passe, et nom utilisateur/motdepasse

5) Activer SSHv2 (seulement) ça n'est pas obligatoire mais recommandé.

6) Configurer lignes VTY

Pour générer une clé il est obligatoire de configurer un hostname et un nom de domaine d'abord

Comme on peut le voir :

```
Router(config)#crypto key generate rsa

% Please define a hostname other than Router.

Router(config)#hostname R2

R2(config)#crypto key generate rsa

% Please define a domain-name first.

R2(config)#ip domain name jeremysitlab.com

R2(config)#crypto key generate rsa
The name for the keys will be: R2.jeremysitlab.com
[output omitted]
```

Depuis un PC on peut utiliser la commande suivante pour se connecter par SSH :

ssh -l username ip-address OU *ssh username@ip-adress*

Cours 43 : FTP & TFTP

Dans ce cours nous verrons le fonctionnement des protocoles FTP (File Transfer Protocol) & TFTP (Trivial File Transfer Protocol). Comme leurs noms le laisse penser ces protocoles sont utilisés pour transférer des fichiers à travers un réseau.

Dans verrons tout d'abord l'intérêt d'utiliser FTP/TFTP, ainsi que leurs fonctionnements et leurs différences. Nous verrons ensuite comment les fichiers du système sont utilisés et stockés dans un IOS Cisco. En dernier temps nous verrons comment utiliser les protocoles FTP/TFTP sur l'IOS Cisco.

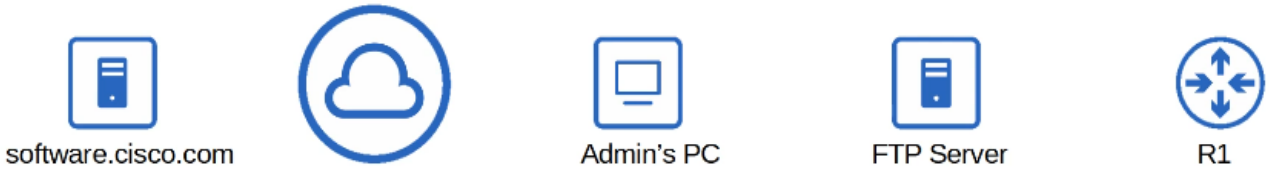
FTP (File Transfer Protocol) et TFTP (Trivial File Transfer Protocol) est un protocole de l'industrie Standard utilisé pour transférer des fichiers à travers le réseau.

Ils utilisent tout deux un modèle de client/serveur. Les clients peuvent utiliser FTP ou TFTP pour copier des fichiers depuis un serveur. Les clients peuvent aussi utiliser FTP ou TFTP pour copier des fichier vers un serveur.

En tant qu'ingénieur réseau, l'utilisation la plus commune de FTP/TFTP est dans la procédure de mettre à jour l'OS d'un appareil du réseau.

On peut utiliser FTP/TFTP pour télécharger une nouvelle version d'un IOS depuis un serveur, et puis redémarrer l'appareil avec la nouvelle image IOS.

Voici un schéma réseau afin de mieux comprendre :



Le serveur Cisco, récupère l'image IOS depuis Cisco, puis il place l'image IOS sur un serveur joignable par l'appareil pour être mis à jour, sur le réseau se sera le serveur FTP. Puis le routeur R1 utilise FTP/TFTP pour copier le fichier dans la mémoire flash de l'appareil.

Le Protocole TFTP (Trivial File Transfer Protocol) a été standardisé en 1981.

Il est appelé « trivial » car il est simple est a des fonctions basiques comparé à FTP.

Il permet seulement à un client de copier un fichier vers ou depuis un serveur.

Ce protocole a été publié après FTP, mais il n'est pas le remplaçant de FTP. C'est un autre outil à utiliser lorsque la simplicité d'utilisation est plus importante que les fonctionnalités.

TFTP n'utilise pas d'authentification (nom d'utilisateur/mot de passe), donc les serveur répondent à toutes les requêtes TFTP. Il n'y a pas de cryptage des données, les données sont transmises en texte clair. C'est la meilleure solution dans un environnement contrôlé pour transférer de petit fichier rapidement. Les serveurs TFTP écoutent sur le port UDP 69.

UDP est une connexion et ne fournit pas de fiabilité avec une retransmission.

TFTP a des fonctions similaires incluses au protocole.

Afin de démontrer la fiabilité de TFTP nous utiliserons le réseau suivant :

Toutes les données de messages TFTP sont confirmés, donc si le client transfère un fichier vers un serveur, le serveur enverra des messages de ACK pour acknowledgment.

Un chronomètre est utilisé et si un message attendu n'est pas reçu à temps, à la fin du temps d'attente de l'appareil celui ci renverra son message précédent.

Par exemple dans le cas du réseau présenté si le client TFTP (à droite) veut télécharger un document depuis le serveur TFTP (à gauche), le client envoie tout d'abord une requête de lecture, le serveur répond avec un

message contenant la donnée et le fichier. Le client répond alors avec un message de confirmation (ACK) mais pour une raison quelconque le message n'atteint pas le serveur, puisque le client a envoyé un ACK, il attend pour le message de données suivant mais celui ci n'arrive pas car le message ACK non plus n'a pas atteint le serveur. Dans ce cas le client va renvoyé un message de confirmation une seconde fois au serveur TFTP pour lui confirmer avoir reçut le message.

Ce processus continue jusqu'à que le client ait reçut le fichier en entier.

TFTP utilise une communication dite de « lock-step ». Le client et le serveur envoient alternativement un message et attend pour une réponse (les retransmission sont envoyés si besoin).

Le transfert de fichier avec TFTP ce passe en trois phases :

1. Connexion : Le client TFTP envoie une requête au serveur, et le serveur lui répond en initialisant la connexion.

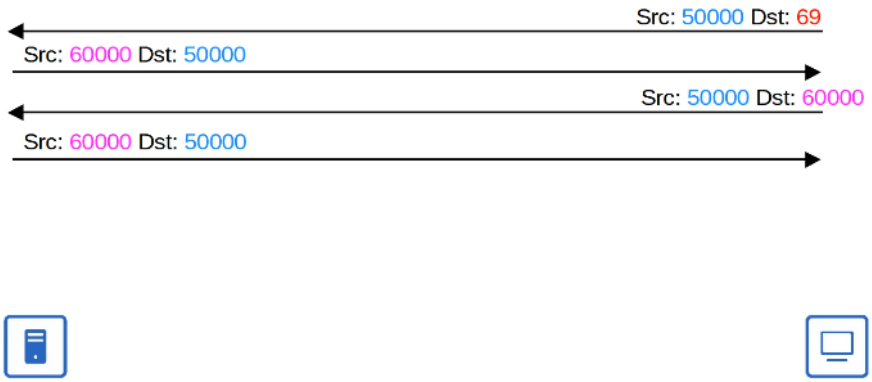
2. Transfert de données : le client et le serveur échangent des messages TFTP. L'un envoie la donnée et l'autre envoie une confirmation.

3. Terminaison de connexion : Après le dernier message de donnée envoyé, une confirmation finale est envoyé pour terminer la connexion.

Lorsque le client envoie le premier message au serveur, le port de destination est UDP 69 et la source est un port aléatoire éphémère.

Ce port aléatoire est appelé « Transfer Identifier » (TID) et permet d'identifier le transfert des données. Le serveur sélectionne un TID aléatoire à utiliser comme port source lorsqu'il répond, il n'utilise plus le port 69. Lorsque le client envoie son message suivant le port de destination sera le TID du serveur et non pas 69.

Voici un schéma pour mieux comprendre cela :



Ici le port est au départ 69, un port aléatoire est utilisé, ici le port 60000 et les deux appareils continuent à échanger avec ces deux ports là.

A présent que nous avons le fonctionnement de TFTP voyons le fonctionnement de FTP.

FTP est un protocole standardisé en 1971, cela est même plus ancien que TCP IP, il s'agit donc d'un très ancien protocole. FTP utilise les port TCP 20 et 21 à la place d'un seul et unique port.

Des noms d'utilisateurs et mots de passes sont utilisés pour l'authentification même s'il n'y a pas de cryptage.

Pour une meilleure sécurité le protocole FTPS (FTP qui utilise SSL/TLS) peut être utilisé.

Une autre option existe avec le protocole STFT (FTP qui utilise SSH) peut aussi être utilisé.

FTP est plus complexe que TFTP et ne permet pas uniquement aux clients le transfert de fichiers, ceux ci peuvent aussi naviguer dans les répertoire, ajouter un répertoire, le supprimer, lister les fichiers etc...

Avec TFTP le client peut seulement informer le serveur de « lui donner un fichier » ou de « prendre tel fichier ».

Le client envoie des commandes FTP au serveur pour faire fonctionner ces fonctions.

Comme explique, FTP utilise deux ports : 20-21

Cela permet deux types de connexions :

Un contrôleur de connexion FTP (TCP 21) est établi et utilisé pour envoyer des commandes FTP et les réponses.

Lorsque les fichier ou données doivent être transférés, séparément par FTP data (TCP 20) les connexions sont établis et terminés si besoin.

Tout d'abord il y a une connexion TCP initié avec le SYN, SYN-ACK, ACK de connexion TCP classique. Maintenant que le contrôle de connexion est établi, le client envoie une commande FTP au serveur, par exemple pour dire que le PC veut avoir un fichier depuis le serveur. Le serveur répond avec une confirmation (ACK). Il y a ensuite deux différents modes qui peuvent être utilisé pour établir la connexion de données FTP.

La méthode par défaut pour établir une connexion de données FTP est avec le mode actif, dans lequel c'est le serveur qui initie la connexion TCP, puis une fois la connexion établie celui procède à un échange de données FTP.

Voyons à présent comment la connexion est établie en mode passif dans ce mode c'est le client qui initie en premier la connexion, cela est souvent nécessaire lorsque le client se trouve derrière un Firewall qui bloque les connexion entrantes depuis le serveur. Pour démontrer cela on utilisera le réseau suivant dans lequel est présent un firewall.

Dans ce cas la connexion est initié par le client qui effectue la requête TCP SYN.

La connexion FTP des données se fait ensuite entre le serveur/client.

Voyons quelques comparaisons des deux protocoles :

FTP : utilise TCP (20 pour les données, 21 pour le contrôle). Les clients peuvent utiliser des commandes FTP pour faire fonctionner différentes action, pas seulement pour copier des fichiers. FTP utilise une authentification par NomUtilisateur/MotDePasse. FTP est plus complexe que TFTP

TFTP : Utilise le port UDP 69, une forme de connexion basique est utilisé. Les clients peuvent uniquement copier des fichiers vers ou depuis un serveur. Il n'y a pas d'authentification requise.

TFTP est plus simple que FTP.

Voyons rapidement la gestion des fichiers systèmes IOS.

Un fichier est une manière de contrôler comment une donnée est stocké et récupérer.

Il est possible de voir le système de fichier d'un appareil sous IOS Cisco avec la commande :

```
show file systems
```

```
Router#show file systems
File Systems:

      Size(b)       Free(b)      Type  Flags  Prefixes
*   2142715904    1994403840      disk     rw   flash0: flash:#
            -             -      disk     rw   flash1:
       966656        962560      disk     rw   flash2:#
            -             -      disk     rw   flash3:
            -             -    opaque     rw   archive:
            -             -    opaque     rw   system:
       262144        256791     nvram     rw   nvram:
            -             -    opaque     rw   tmpsys:
            -             -   network     rw   snmp:
            -             -    opaque     rw   null:
            -             -   network     rw   tftp:
            -             -    opaque     ro   xmodem:
            -             -    opaque     ro   ymodem:
            -             -    opaque     wo   syslog:
            -             -   network     rw   rcp:
            -             -   network     rw   pram:
            -             -   network     rw   ftp:
[output omitted]
```

On peut voir le type de fichiers avec des informations qui sont :

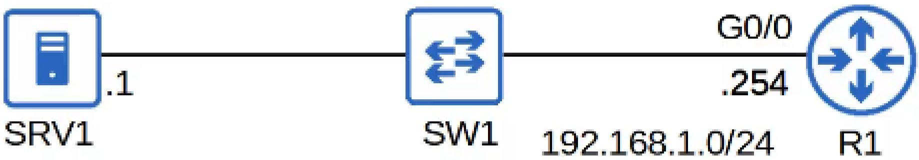
Disk : stockage de l'appareil comme une mémoire flash.

Opaque : utilisé pour des fonctions internes.

Nvram : NVRAM interne. La fichier startup-config est stocké à l'intérieur.

Network : représente un fichier externe du système, par exemple un serveur FTP/TFTP.

Voyons à présent comment utiliser TFTP et FTP pour transférer des fichiers dans la mémoire flash d'un appareil Cisco. Nous utiliserons pour démontrer cela le réseau suivant :



Il est possible d'afficher la version actuelle d'un IOS avec la commande : *show version*

```
R1#show version
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.1(4)M4, RELEASE SOFTWARE
(fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thurs 5-Jan-12 15:41 by pt_team
[output omitted]
```

On peut afficher le contenu de la mémoire flash avec la commande : `show flash`

```
R1#show flash
System flash directory:
File  Length   Name/status
  3   33591768 c2900-universalk9-mz.SPA.151-4.M4.bin
  2   28282    sigdef-category.xml
  1   227537   sigdef-default.xml
[33847587 bytes used, 221896413 available, 255744000 total]
249856K bytes of processor board System flash (Read/Write)
```

Imaginons que l'on souhaite mettre à jour le Routeur R1 avec l'IOS C2900 de la dernière version contenu dans le routeur SRV1.

On utilisera pour cela la commande suivante depuis le le routeur R1 : `copy tftp: flash:`

```
R1#copy tftp: flash:
Address or name of remote host []? 192.168.1.1
Source filename []? c2900-universalk9-mz.SPA.155-3.M4a.bin
Destination filename [c2900-universalk9-mz.SPA.155-3.M4a.bin]?

Accessing tftp://192.168.1.1/c2900-universalk9-mz.SPA.155-3.M4a.bin....
Loading c2900-universalk9-mz.SPA.155-3.M4a.bin from
192.168.1.1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 33591768 bytes]

33591768 bytes copied in 4.01 secs (879550 bytes/sec)
```

Le format de la commande copie depuis la source `tftp` vers la destination `flash`.

Ensuite est demandé d'entrer l'adresse IP du serveur TFTP ici 192.168.1.1

On entre ensuite le nom du fichier du serveur TFTP que l'on veux télécharger et le fichier se télécharge.

On peut à présent lancer la commande `show flash` afin d'afficher les fichiers installés sur le routeur R1, on peut y voir le fichier téléchargé à partir de SRV1.

```
R1#show flash

System flash directory:
File  Length   Name/status
  3   33591768 c2900-universalk9-mz.SPA.151-4.M4.bin
  4   33591768 c2900-universalk9-mz.SPA.155-3.M4a.bin
  2   28282    sigdef-category.xml
  1   227537   sigdef-default.xml
[67439355 bytes used, 188304645 available, 255744000 total]
249856K bytes of processor board System flash (Read/Write)


R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#boot system flash:c2900-universalk9-mz.SPA.155-3.M4a.bin
R1(config)#exit
R1#write memory
Building configuration...
[OK]
R1#reload
Proceed with reload? [confirm]
```

Afin de mettre à jour le système on lance ici les commandes en mode privilégiés :

```
R1(config)#boot system flash:c2900-universalk9-mz.SPA.155-3.M4a.bin
R1(config)#exit
R1#write memory
R1#reload
```

Le système est à présent à jour après redémarrage. On peut à présent y afficher la version utilisé par l'IOS avec la commande *show version* et *show flash*

```
R1#show version
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.5(3)M4a, RELEASE SOFTWARE(fc1)
[output omitted]

R1#delete flash:c2900-universalk9-mz.SPA.151-4.M4.bin
Delete filename [c2900-universalk9-mz.SPA.151-4.M4.bin]?
Delete flash:/c2900-universalk9-mz.SPA.151-4.M4.bin? [confirm]

R1#show flash

System flash directory:
File  Length   Name/status
  4   33591768 c2900-universalk9-mz.SPA.155-3.M4a.bin
  2   28282    sigdef-category.xml
  1   227537   sigdef-default.xml
[33847587 bytes used, 221896413 available, 255744000 total]
249856K bytes of processor board System flash (Read/Write)
```

On peut voir qu'il s'agit bien de la version installé depuis le serveur TFTP.

Si l'on souhaite supprimer l'ancienne version de l'IOS utilisé, on utilise pour cela la commande ***delete*** suivi du chemin du fichier.

```
R1#delete flash :c2900-universalk9-mz.SPA.151-4.M4a.bin
```

Voyons à présent comment faire lorsque l'on souhaite utiliser le protocole FTP pour transférer des fichiers.

```
R1(config)#ip ftp username cisco

R1(config)#ip ftp password cisco

R1(config)#exit

R1#copy ftp: flash:
Address or name of remote host []? 192.168.1.1
Source filename []? c2900-universalk9-mz.SPA.155-3.M4a.bin
Destination filename [c2900-universalk9-mz.SPA.155-3.M4a.bin]?

Accessing ftp://192.168.1.1/c2900-universalk9-mz.SPA.155-3.M4a.bin…
Loading c2900-universalk9-mz.SPA.155-3.M4a.bin from
192.168.1.1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[output omitted]
```

Pour cela on commence tout d'abord par s'authentifier au serveur ftp avec les commandes :

```
R1(config)#ip ftp username cisco
R1(config)#ip ftp password cisco
R1(config)#exit
R1#copy ftp: flash:
```

Les commandes pour copier, supprimer, les fichiers sont identiques à celle utilisés pour TFTP.

La principale différence se situe lors de la connexion, avec FTP il faut tout d'abord se connecter au serveur FTP en utilisant le nom d'utilisateur/MotDePasse du serveur.

Cours 44 : NAT (Partie 1)

Dans ce cours nous verrons le fonctionnement de NAT (Network Address Translation), qui est utilisé pour traduire la source et/ou l'adresse IP de destination d'un paquet vers une adresse IP différente.

Nous verrons tout d'abord les différents adressages d'adresses IP privée, puis nous verrons le fonctionnement de NAT, avec également le fonctionnement du NAT statique et de sa configuration.

IPV4 ne fournit pas assez d'adresses IP privées pour tous les appareils qui en ont besoin dans le monde, car il n'y en a pas suffisamment, la solution à long terme à ce problème est de changer le protocole vers l'IPv6. De changer toutes les adresses IPv4 disponibles en adresses IPv6 est une tâche compliqué, c'est pour cela qu'a été adopté 3 solutions à court terme :

1) CIDR (Classless Inter Domain Routing)

2) Les adresses IP privées

3) NAT

Le Request For Comment (RFC) 1918 spécifie le classement d'adressage IPv4 comme privée :

- Classe A : 10.0.0.0/8 (10.0.0.0 à 10.255.255.255)

- Classe B : 172.16.0.0/12 (172.16.0.0 à 172.31.255.255)

- Classe C : 192.168.0.0/16 (192.168.0.0 à 192.168.255.255)

Il est permis d'utiliser le classement de ces adresses dans un réseau privée. Elles ne peuvent pas être utilisés de manière globale.

Un ordinateur connecté à un réseau utilise très probablement des adresses IP privées, par exemple comme on peut le voir ci dessous.

```
C:\Users\user>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.0.167
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

Les adresses IP privées ne peuvent pas être utilisés à travers Internet. L'ISP ne fournit pas de telles adresses.

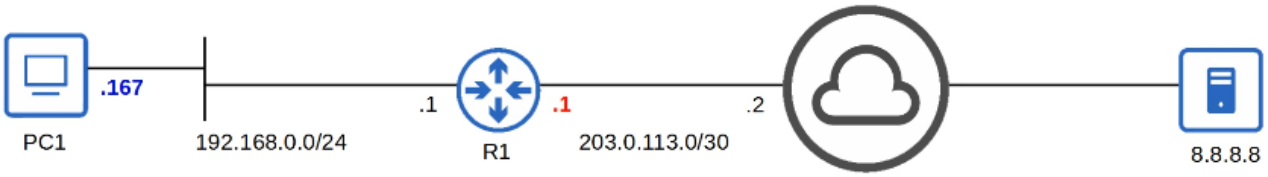
Dans le réseau suivant, deux problèmes sont présent, premièrement les deux PC utilisent tout deux la même adresse IP, et deuxièmement les adresses IP privées ne peuvent pas être utilisés à travers Internet, donc les PC n'accéderont pas à Internet.

NAT permet de résoudre ces deux problèmes, car les routeurs pourront utiliser une adresse IP publique vers Internet. Bien que les adresses IP privées ne peuvent pas être unique, les adresses IP publiques peuvent l'être.

Network Address Translation (NAT) est utilisé pour modifier la source et/ou l'adresse IP de destination d'un paquet.

Il y a nombreuses raisons d'utiliser NAT, mais la raison la plus commune est pour permettre à un hôte avec une adresse IP privée de communiquer avec d'autre hôtes à travers Internet.

Voyons une démonstration rapide de NAT sur le réseau suivant :

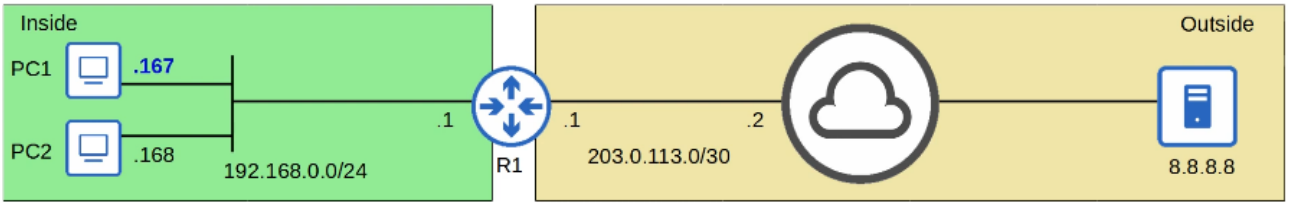


Le PC1 veut communiquer avec le serveur pour cela il crée un paquet avec pour adresse IP source : 192.168.0.167 et pour adresse de destination : 8.8.8.8, il envoie le paquet à sa passerelle par défaut R1. C'est à ce moment que le NAT se passe, R1 traduit 192.168.0.167 en 203.0.113.1, C'est pour cela que ce procédé est appelé source NAT, car il traduit l'adresse IP source, le routeur à ici traduit l'adresse IP privée par une adresse de sa propre interface.

R1 envoie ensuite le paquet vers l'adresse du serveur, ici 8.8.8.8

Le serveur envoie alors une réponse, avec l'adresse IP source : 8.8.8.8 et adresse IP de destination : 203.0.113.1, le paquet est ensuite traduit de la même façon que lors de la première requête.

Voyons à présent le fonctionnement de statique source NAT avec le réseau suivant :



Le NAT statique implique de configurer de manière statique en cartographiant une à une les adresses IP privées vers des adresses IP publiques.

Les adresses peuvent être traduite du publique vers le publique, ou du privée vers le privée, mais voyons comment est traduit une adresse du privée vers le publique.

Une adresse IP local intérieur est cartographié vers une adresse IP intérieur globale.

« Intérieur local » fait références aux adresses IP du réseau local. À l'opposé du « intérieur global » qui fait référence aux adresses IP du réseau externe des hôtes.

Par exemple le PC1 veut communiquer avec le serveur 8.8.8.8, il utilisera tout d'abord l'adresse intérieur local qui sera traduite par le routeur avec l'IP 100.0.0.1, le routeur envoie ensuite le paquet vers le serveur 8.8.8.8 qui lui envoie la réponse vers l'adresse IP publique du routeur R1.

Voyons à présent comment le PC2 communique avec le serveur, pour cela il aura besoin de sa propre adresse IP. Le routeur ne permet pas de pouvoir cartographier les adresses de PC1 et PC2 en utilisant le NAT statique car il lui faudra une seconde fois l'adresse 100.0.0.1, c'est pour cela qu'en configurant une deuxième adresse IP en NAT statique 100.0.0.2, le deuxième PC pourra communiquer avec le réseau extérieur.

Le NAT statique permet aux appareils avec une adresse IP privée de communiquer à travers Internet. Puisque cela requiert une cartographie d'adresse une par une cela ne permet pas de préserver les adresses IP.

Voyons comment configurer le NAT statique :

```
R1(config)#int g0/1
R1(config-if)#ip nat inside

R1(config-if)#int g0/0
R1(config-if)#ip nat outside
R1(config-if)#exit

R1(config)#ip nat inside source static 192.168.0.167 100.0.0.1
R1(config)#ip nat inside source static 192.168.0.168 100.0.0.2
R1(config)#exit

R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 100.0.0.1:56310    192.168.0.167:56310 8.8.8.8:53         8.8.8.8:53
--- 100.0.0.1          192.168.0.167      ---                 ---
udp 100.0.0.2:62321    192.168.0.168:62321 8.8.8.8:53         8.8.8.8:53
--- 100.0.0.2          192.168.0.168      ---                 ---
```

On commence tout d'abord par définir l'interface « intérieur » connecté au réseau interne avec la commande :

```
R1(config)#int g0/1
R1(config)#ip nat inside
```

On définit l'interface « extérieur » connecté au réseau externe avec les commandes :

```
R1(config)#int g0/0
R1(config-if)#ip nat outside
```

On configure ensuite la cartographie des adresses IP avec les commandes :

```
R1(config)#ip nat inside source static 192.168.0.167 100.0.0.1
R1(config)#ip nat inside source static 192.168.0.168 100.0.0.2
```

Le format de la commande est : *ip nat inside* source static suivi de l'adresse ip local intérieur puis de l'adresse ip global intérieur

Pour afficher la configuration du nat on utilise la commande :

```
R1#show ip nat translations
```

On peut voir affiché le protocole utilisé ainsi que les adresses local extérieur et global extérieur, on remarque que les adresses du serveur se terminent toutes deux par :53, il s'agit du protocole DNS qui permet aux PC d'accéder au serveur.

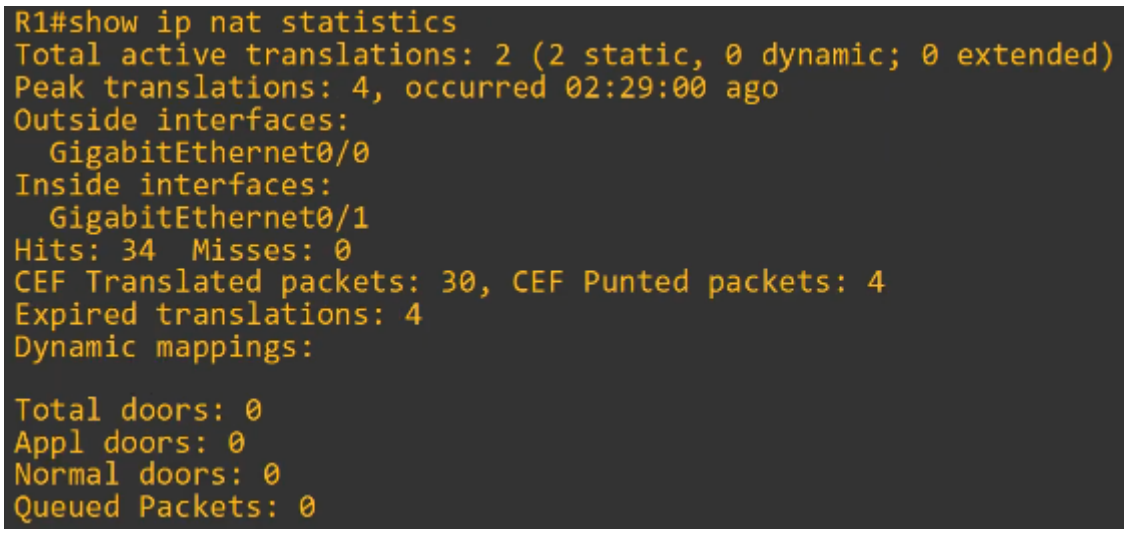
Voyons d'autres commandes qui peuvent être utiles lorsque l'on utilise NAT :

Il est possible de retirer les adresses IP de traduction avec la commande :

```
R1#clear ip nat translations *
```

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 100.0.0.1:56310    192.168.0.167:56310 8.8.8.8:53         8.8.8.8:53
--- 100.0.0.1          192.168.0.167      ---                 ---
udp 100.0.0.2:62321    192.168.0.168:62321 8.8.8.8:53         8.8.8.8:53
--- 100.0.0.2          192.168.0.168      ---                 ---

R1#clear ip nat translation *

R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 100.0.0.1          192.168.0.167      ---                 ---
--- 100.0.0.2          192.168.0.168      ---                 ---
```

La commande suivante permet d'afficher de manière plus détaillé les tables NAT de chaque interface



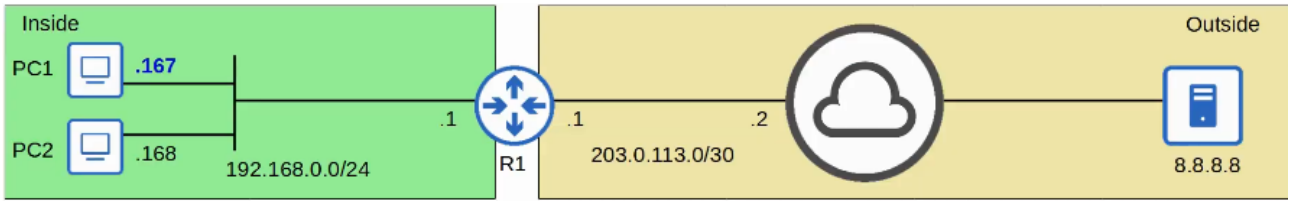

```
R1#show ip nat statistics
```

Cours 45 : NAT (Partie 2)

Dans ce cours nous continuerons la seconde partie du sujet NAT (Network Address Translation).

Nous reverrons une nouvelle chose à propos du NAT statique, nous verrons ensuite le NAT dynamique qui permet de cartographier les adresses automatiquement au lieu de le faire manuellement pour chaque adresses. Puis un autre type de NAT qui est le dynamic PAT qui permet de traduire non pas seulement l'adresse IP mais aussi le numéro de port.

Nous verrons tout d'abord dans le réseau suivant le fonctionnement du NAT statique :



Le NAT statique permet de configurer statiquement en cartographiant chaque adresses IP privée une à une par des adresses IP publiques.

Lorsque le trafique depuis l'hôte interne est envoyé vers le réseau externe, le routeur va traduire l'adresse IP source.

Par exemple avec le NAT statique :

192.168.0.167 sera traduit en 100.0.0.1

192.168.0.168 sera traduit en 100.0.0.2

Cette cartographie permet à un hôte externe d'accéder aux hôtes internes par l'adresse IP interne global.

Dans le NAT dynamique, le routeur cartographie dynamiquement des adresses IP interne local vers des adresses interne globale selon le nombre d'adresses nécessaires.

Une ACL est utilisé pour identifier quelle trafique doit être traduit.

Si l'adresse IP source est permise par l'ACL, l'adresse IP source sera traduite.

Si l'adresse IP source est bloqué par l'ACL, l'adresse IP source ne sera pas traduite.

Un pool NAT est utilisé pour définir les adresses IP interne global.

Par exemple sur le réseau précédent, sur R1,

l'ACL 1 :

permit 192.168.0.0/24

deny any

POOL1 : 100.0.0.1 à 100.0.0.10

Si un paquet avec une adresse IP source permise par l'ACL 1 arrive, l'adresse IP traduira l'adresse IP source vers une adresse du POOL1.

Si une adresse est « denied » par l'ACL cela ne signifie pas que l'adresse IP sera bloqué mais simplement que l'adresse ne sera pas traduite.

Bien que les adresses soient assignés dynamiquement la cartographie est toujours une à une (une adresse ip local interne par adresse ip global interne)

S'il n'y as pas suffisamment d'adresses IP interne global disponible (par exemple que celles disponibles sont utilisés), cela sera appelé « NAT pool exhaustion ».

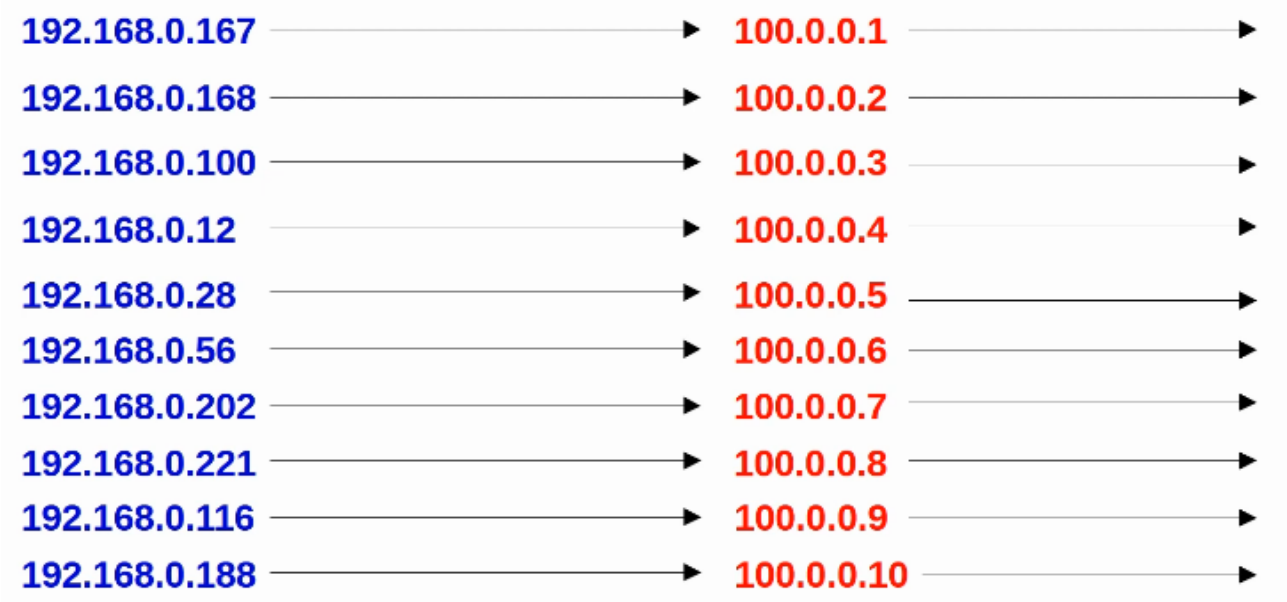
Si un paquet d'un autre hôte interne arrive et à besoin d'un NAT mais qu'il n'y a pas d'adresses disponibles, le routeur ne gardera pas le paquet.

L'hôte ne sera pas possible d'accès en dehors du réseau jusqu'à ce que l'une des adresses IP global internes ne deviennent disponibles.

Les entrées NAT dynamiques seront expirés automatiquement si non utilisés, il est aussi possible de les supprimer automatiquement.

Voyons comment fonctionne le NAT Pool Exhaustion, l'adresse IP source 192.168.0.167 est traduite en 100.0.0.1, l'adresse IP 192.168.0.168 est aussi traduite en 100.0.0.2, etc...

même chose pour toutes les adresses suivantes :



Si une nouvelle adresse veut être traduite par exemple 192.168.0.98, puisque plus aucune adresse n'est disponible le routeur va bloquer le paquet. Pour que cette nouvelle adresse soit joignable, l'adresse IP 192.168.0.167 est supprimé car expiré après un certain temps, l'adresse 192.168.0.98 pourra joindre le trafique en utilisant l'adresse traduite de l'ancienne adresse : 100.0.0.1

Il reste tout de même possible pour les hôtes d'utiliser plusieurs fois la même adresse IP publique par le moyen de PAT (Port Address Translation)

Pour configurer le NAT dynamique on utilise les commandes suivantes :

On définit l'interface interne connecté au réseau interne avec les commandes :

```
R1(config)#int g0/1
R1(config)#ip nat inside
```

Pour définir l'interface externe connecté au réseau externe on lance les commandes

```
R1(config-if)#int g0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```

Pour définir le trafique qui devrait être traduit on lance les commandes :

```
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
R1(config)#ip nat pool POOL1 100.0.0.0 100.0.0.255 prefic-length 24
R1(config)#ip nat inside source list 1 pool POOL1
```

Pour définir le pool des adresses IP interne global on utilise la commande :

```
R1(config)#ip nat pool POOL1 100.0.0.0 100.0.0.255 prefix-length 24
```

Pour configurer un NAT dynamique avec une cartographie de l'ACL au pool on utilise la commande :

```
R1(config)#ip nat inside source list 1 pool POOL1
```

Pour afficher la table NAT on lance la commande : *show ip nat translations*

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 100.0.0.1:3       192.168.0.167:3    8.8.8.8:3          8.8.8.8:3
udp 100.0.0.1:58685    192.168.0.167:58685 8.8.8.8:53        8.8.8.8:53
--- 100.0.0.1          192.168.0.167      ---                ---
icmp 100.0.0.2:3       192.168.0.168:3    8.8.8.8:3          8.8.8.8:3
udp 100.0.0.2:49536    192.168.0.168:49536 8.8.8.8:53        8.8.8.8:53
--- 100.0.0.2          192.168.0.168      ---                ---
```

Comme on peut le voir les adresses IP sont retirés lorsque l'on relance la commande après une minute :

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 100.0.0.1          192.168.0.167      ---                ---
--- 100.0.0.2          192.168.0.168      ---                ---
```

Pour afficher les statistiques de NAT on lance la commande : *show ip nat statistics*

```
R1#show ip nat statistics
Total active translations: 6 (0 static, 6 dynamic; 4 extended)
Peak translations: 6, occurred 00:00:30 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 32  Misses: 0
CEF Translated packets: 20, CEF Punted packets: 12
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool POOL1 refcount 6
 pool POOL1: netmask 255.255.255.0
        start 100.0.0.0 end 100.0.0.255
        type generic, total addresses 256, allocated 2 (0%), misses 0
[output omitted]
```

Avec cette commande on peut afficher l'ACL correspondante à la cartographie du Pool.

Voyons à présent le fonctionnement de PAT (NAT Overload).

PAT permet de traduire les adresses IP et le numéro de port d'un paquet si nécessaire.

En utilisant un port unique pour chaque flux de communication, une seule adresse IP publique peut être utilisé par plusieurs différentes adresses internes des hôtes. (Le numéro de port est 16bits ce qui fait un total de 65 000 numéro de port disponible).

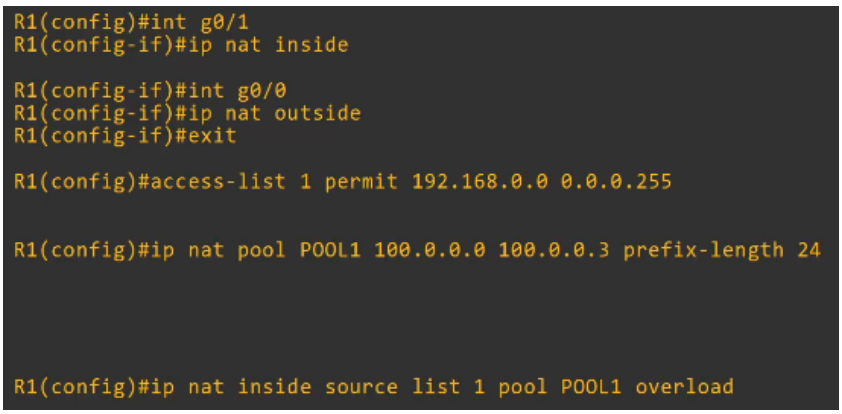
Le routeur va suivre quelle adresse interne local est utilisé par quelle adresse interne global et le port.

Par exemple dans le schéma de réseau précédent le PC1 avec l'adresse IP source 192.168.0.167 :54321 veut joindre le serveur 8.8.8.8 :53, le PC2 avec l'adresse IP source 192.168.0.168 :54321 veut lui aussi joindre le même serveur de l'adresse 8.8.8.8 :53, pour joindre le serveur R1 va traduire les deux adresses en 100.0.0.1 mais avec des numéros de ports différents qui seront : 100.0.0.1 :54321 et 100.0.0.1 :54322

Pour répondre le serveur envoie lui aussi pour adresses de destination la même adresse mais avec des numéros de ports différents : 54321 et 54322.

Puisque plusieurs hôtes peuvent partager une seule et même adresse IP publique, PAT est très utile pour préserver les adresses IP publique et est utilisé dans tous les réseau dans le monde.

Voici les commandes qui permettent de configurer PAT :



On commence par définir l'interface interne connecté au réseau interne avec les commandes :

```
R1(config)#int g0/1
R1(config-if)#ip nat inside
```

Pour définir l'interface externe connecté au réseau externe on utilise les commandes :

```
R1(config-if)#int g0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```

Pour définir le trafique qui doit être traduit on utilise la commande :

```
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
```

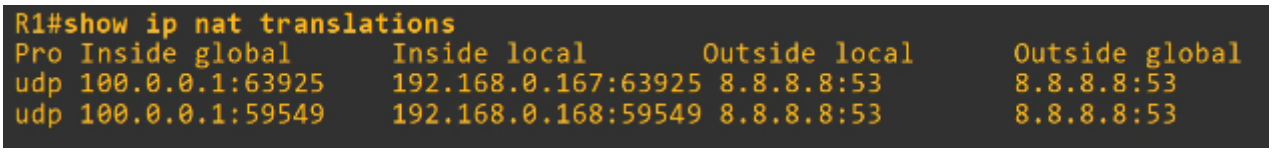
Pour définir le pool des adresses IP interne global on lance la commande :

```
R1(config)#ip nat pool POOL1 100.0.0.0 100.0.0.3 prefix-length 24
```

Pour configurer le PAT en cartographiant l'ACL au pool et en utilisant l'overload on lance la commande :

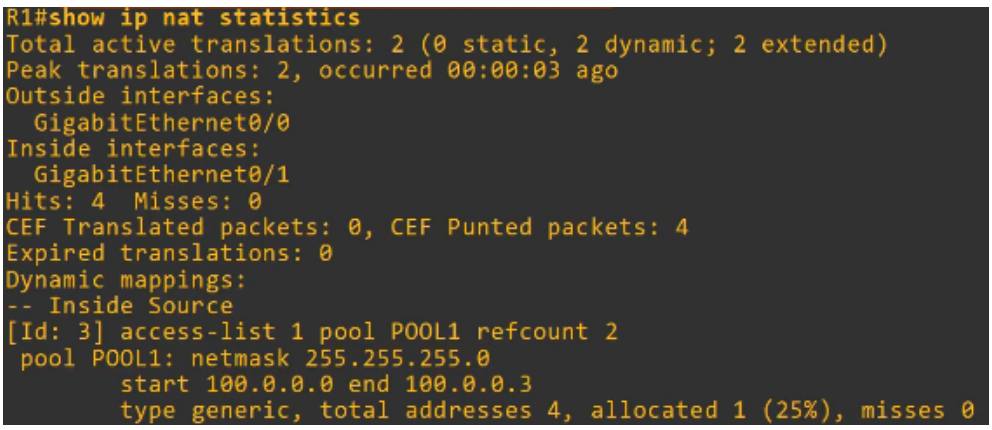
```
R1(config)#ip nat inside source list 1 pool POOL1 overload
```

Voyons la configuration utilisé par le routeur R1 :



Comme on peut le voir ci dessus l'adresse IP interne global est identique sur les deux ligne, c'est le numéro de port qui change.

On peut afficher plus de détail avec la commande : *show ip nat statistics*



Une autre manière de configurer PAT, il s'agit même peut être de la manière la plus commune, qui est de configurer le routeur afin qu'il utilise sa propre adresse IP lorsqu'il traduit l'adresse IP source d'autre paquet, pour cela on lance les commandes suivantes :

```
R1(config)#int g0/1
R1(config-if)#ip nat inside

R1(config-if)#int g0/0
R1(config-if)#ip nat outside
R1(config-if)#exit


R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255



R1(config)#ip nat inside source list 1 interface g0/0 overload
```

Pour de définir une interface interne connecté au réseau interne on utilise les commandes :

```
R1(config)#int g0/1
R1(config-if)#ip nat inside
```

Pour définir une interface externe connecté au réseau externe, on utilise les commandes :

```
R1(config-if)#int g0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```

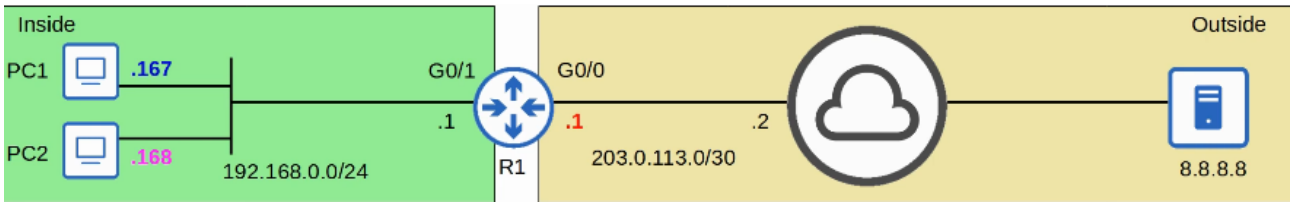
On définit ensuite le trafique qui devra être traduit avec la commande :

```
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
```

Pour configurer PAT en cartographiant l'ACL vers l'interface et activer overload on lance la commande :

```
R1(config)#ip nat inside source list 1 interface g0/0 overload
```

On peut démontrer cela avec le réseau suivant :



Lorsque les PC1 et PC2 envoient un paquet au serveur, l'adresse IP traduite par le routeur est la même que celle des PC mais le numéro de leurs ports est différent.

Voici l'affichage de la configuration avec la commande : *show ip nat translations*

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 203.0.113.1:65205  192.168.0.167:65205 8.8.8.8:53         8.8.8.8:53
udp 203.0.113.1:59641  192.168.0.168:59641 8.8.8.8:53         8.8.8.8:53
```

```
R1#show ip nat statistics
Total active translations: 2 (0 static, 2 dynamic; 2 extended)
Peak translations: 2, occurred 00:36:30 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 12  Misses: 0
CEF Translated packets: 0, CEF Punted packets: 12
Expired translations: 4
Dynamic mappings:
-- Inside Source
[Id: 4] access-list 1 interface GigabitEthernet0/0 refcount 2
```

On affiche plus de détails avec la commande : *show ip nat statistics*

Cours 46 : Quality of Service (Partie 1)

Dans ce cours nous verrons ce qu'est la qualité de service ou en Anglais QoS (Quality of Service).

La qualité de service est utilisé pour prioriser certain type de trafique réseau pour minimiser des choses comme les délais et la perte de paquets.

QoS est souvent utilisé pour prioriser la Voix à travers le trafique de l'IP depuis un téléphone IP pour s'assurer que la qualité de l'audio est correct.

Nous verrons donc tout d'abord une introduction sur les téléphones IP et sur les la voix à travers les VLANs, puis nous verrons ce qu'est le Power over Ethernet (PoE) et nous donnerons une introduction à la qualité de service (QoS) pour mieux comprendre comment cela fonctionne.

Lest téléphones IP sont des téléphones standards qui fonctionnent à travers le réseau des téléphones publique ou en anglais : public switched telephone network (PSTN).

Certaines fois cela est appelé POTS (Plain Old Telephone Service)

Les téléphones IP utilisent VoIP (Voice over IP) qui est une technologie qui permet au téléphones d'appeler à travers un réseau IP comme Internet.

Les téléphones IP sont connectés à un Switch comme n'importe quelle hôte. Il existe cependant une autre option pour connecter ces appareils à un Switch.

Un téléphone IP Cisco ressemble à cela :

Les téléphones IP on 3 ports Switch Internes :

- 1 port est le uplink qui connecte au switch externe.

- 1 port est le downlink vers le PC

- 1 port est connecté en interne au téléphone lui même

Il y a donc à l'intérieur du téléphone IP une sorte de petit Switch interne avec 3 ports.

L'un est utilisé pour se connecter le Switch en Ethernet, un autre pour se connecter au PC avec un câble Ethernet. Et le dernier port sert à se connecter en interne au téléphone lui même.

Cela permet au PC et au téléphone IP de partager un seul port Switch. Le trafique depuis le PC passe par l'IP du téléphone vers le Switch.

Il est recommandé de séparer le trafique de la « voix » (depuis le téléphone IP) et le trafique de donnée (depuis le PC) en les plaçant dans des Vlans séparés. Cela peut être fais en utilisant un voice VLAN.

Le trafique depuis le PC sera non balisé, mais le trafique depuis le téléphone sera balisé avec un ID de VLAN.

Nous utiliserons ce réseau pour configurer le Switch :

Voici les commandes à exécuter pour configurer le Switch :

```
SW1(config)#interface gigabitethernet0/0
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
SW1(config-if)#switchport voice vlan 11

SW1#show interfaces g0/0 switchport
Name: Gi0/0
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 10 (VLAN0010)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 11 (VLAN0011)
![output omitted]
```

```
SW1(config)#interface gigabitethernet0/0
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
SW1(config-if)#switchport voice vlan 11
```

Le PC1 enverra ici le trafique sans balise. Le SW1 utilisera CDP pour avertir PH1 de baliser le trafique PH1 dans le Vlan 11 On affiche ensuite la configuration avec la commande :

```
SW1#show interfaces G0/0 switchport
```

On peut voir affiché la vlan utilisé pour le port 0/0 est la Vlan 10

la Vlan utilisé pour la voix est la Vlan 11

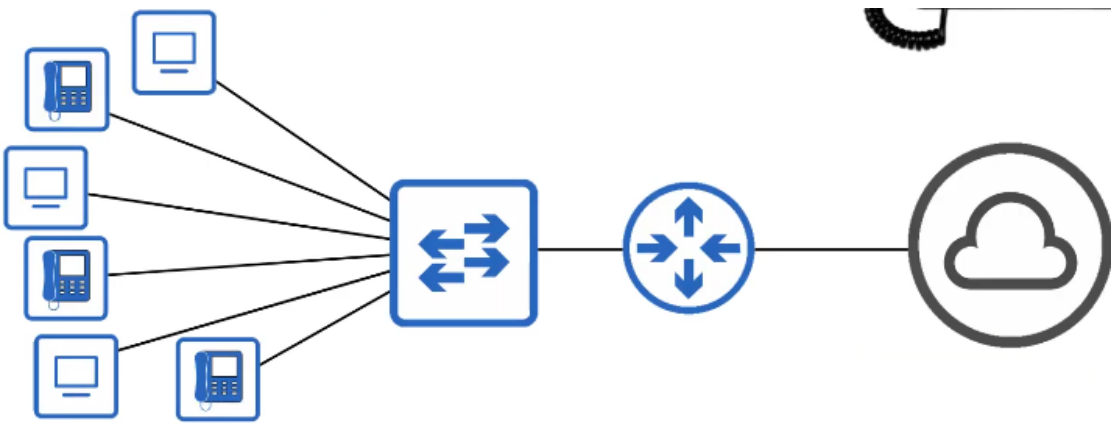
Les mode administratif et opérationnel sont en statique.

Il est aussi possible d'afficher la configuration trunk avec la commande :

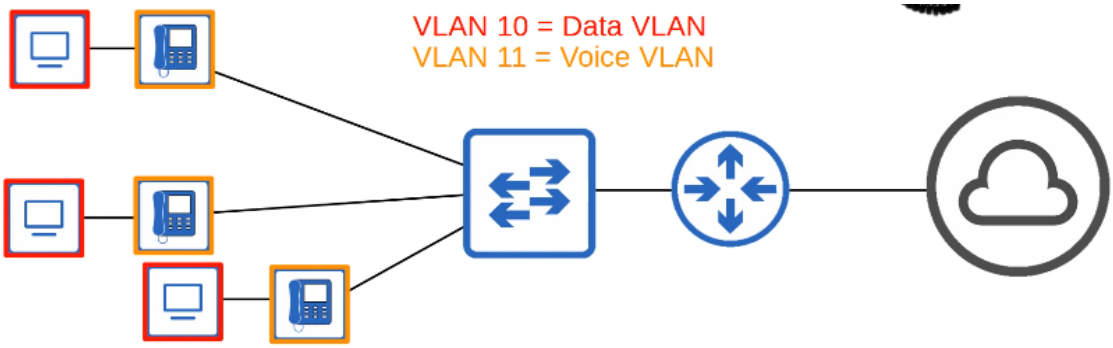
```
SW1#show interfaces trunk
SW1#show interfaces g0/0 trunk
```

```
SW1#show interfaces trunk
SW1#
SW1#show interfaces g0/0 trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       off              negotiate      not-trunking  1

Port        Vlans allowed on trunk
Gi0/0       10-11

Port        Vlans allowed and active in management domain
Gi0/0       10-11

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10-11
```

Donc au lieu d'utiliser un câble pour chacun des PC et téléphones IP comme ceci :



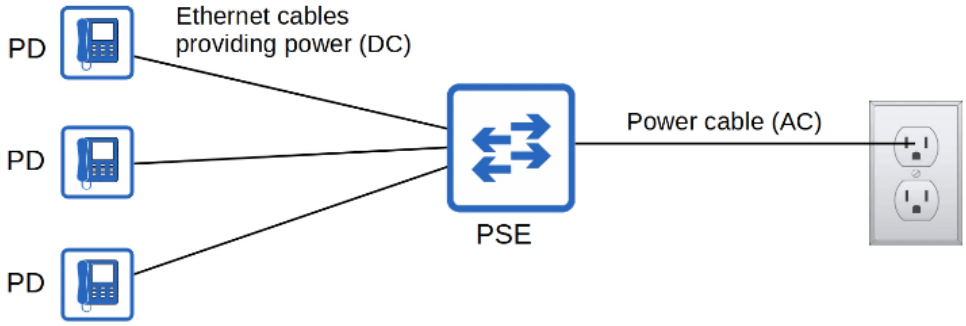
Il est possible d'utiliser moins de ports Ethernet sur le Switch en connectant les téléphones IP au PC et d'utiliser une Vlan pour séparer les trafique par exemple VLAN 10 pour les données et le VLAN 11 pour la VLAN de la voix comme sur le réseau suivant :



PoE permet à des équipements avec sources d'énergie, en Anglais (Power Sourcing Equipment (PSE) de fournir de l'énergie aux appareils à alimenter (Powered Devices ou PD en Anglais) à travers un câble Ethernet.

Le PSE est un Switch et les appareils à alimenter sont des téléphones IP, des caméras IP, des points d'accès, etc...

Le PSE reçoit l'énergie depuis une sortie AC, il le convertit en énergie DC et le fournit cette énergie DC vers les appareils à alimenter.



Il faut tout de même faire attention lorsque l'on utilise cette solution, parfois trop d'électricité peut endommager l'appareil électrique.

PoE possède donc une procédure déterminer si l'appareil connecté à besoin d'énergie et combien d'énergie il en aurait besoin.

Lorsqu'un appareil est connecté à un port PoE, le PSE (Switch) envoie de très faible signaux d'énergie, et gère la réponse afin de déterminer combien de puissance l'appareil aurait besoin.

Si l'appareil à besoin d'énergie, le PSE envoie l'énergie et permet à l'appareil de démarrer.

Le PSE continue de gérer les appareils et le montant requis de puissance énergique.

Il y a une fonction appelé Power policing qui peut être configuré pour empêcher un appareil de prendre trop d'énergie.

```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/0
SW1(config-if)# power inline police
SW1(config-if)# end
SW1# show power inline police g0/0
Available:800(w) Used:32(w) Remaining:768(w)
Interface Admin  Oper       Admin      Oper       Cutoff Oper
          State  State      Police     Police     Power  Power
--------- ------ ---------- ---------- ---------- ------ -----
Gi2/1     auto   on         errdisable ok          17.2  16.7
```

Voici quelques commandes qui permettent la gestion de l'énergie avec la politique d'énergie :

power inline police permet de configurer l'énergie avec des paramètres par défaut : en désactivant le port et en envoyant un message Syslog si l'appareil reçoit trop d'énergie.

Ceci est équivalent à la commande *power inline police action err-disable*

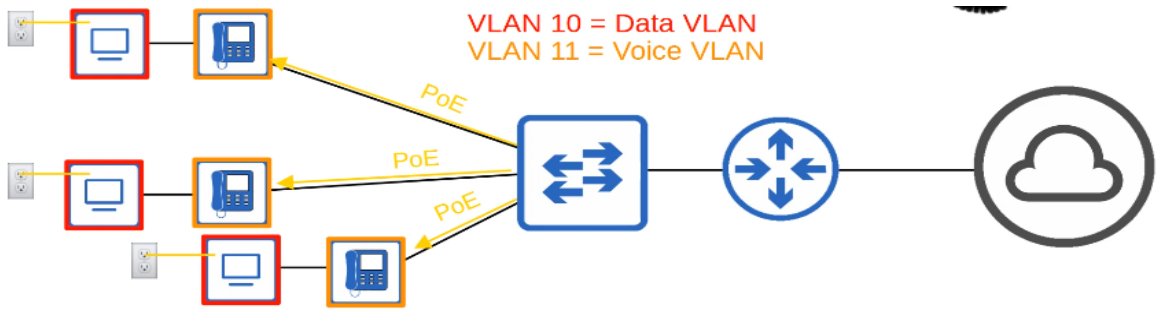
L'interface sera mise en état « error-disabled » et pourra être réactivé avec un *shutdown* suivi d'un *no shutdown*.

power inline police action log n'éteint pas l'interface si l'appareil reçoit trop d'énergie. Cela redémarre l'interface et envoie un message Syslog.

Voici quelques standard de PoE :

Nom	Standard	Watts	Puissance par pair
Cisco Inline Power (ILP)	Crée par Cisco, n'est pas un Standard	7	2
PoE (Type 1)	802.3af	15	2
PoE+(Type 2)	802.3at	30	2
UpoE (Type 3)	802.3bt	60	4
UpoE+ (Type 4)	802.3bt	100	4

Sur le réseau précédent on peut utiliser le PoE pour les téléphones IP et une prise mural pour les PC.

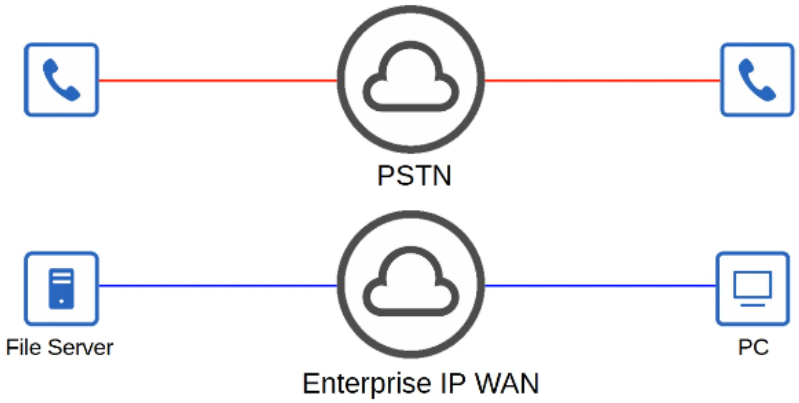


Voyons à présent ce qu'est QoS. Auparavant :

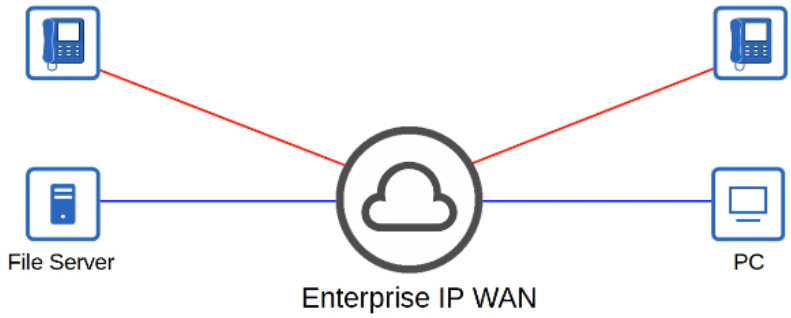
Le trafique de Voix et le trafique des données étaient utilisés par des réseaux totalement séparés.

Le trafique réseau utilisait le PSTN (Public Switch Telephone Network)

Le trafique des données utilisait le réseau IP (WAN entreprise, Internet, etc.)



QoS n'était pas nécessaire car les différents types de trafiques ne se partageaient pas la ban,de passante. Comme dans les réseaux modernes les réseaux partagent le même réseau dans lequel passe le téléphone IP, le trafique vidéo, le trafique de données, etc...



La qualité de service permet donc de gagner de l'argent pour des fonctionnalités avancés pour la voix et le trafique vidéo, par exemple l'intégration avec la collaboration de logiciels (Cisco WebEx, Microsoft Teams, etc...)

Les différents types de trafique doivent maintenant se partager la bande passante.

QoS est un ensemble d'outils utilisés par les appareils réseau pour appliquer différents traitements à différents paquets. Par exemple ajouter une priorité à certain type de trafique et réduire la priorité sur un autre.

QoS est utilisé pour gérer les caractéristiques suivantes du trafique réseau :

1. La bande passante : la capacité de la ligne est mesuré en bits par secondes (Kbps, Mbps, Gbps, etc.) Les outils QoS permettent de réserver un certain montant de bande passante pour un certain type de trafique. Par exemple 20% de trafique de voix, 30% de trafique pour un certain type de données, et laisser 50% pour tout les autres types de trafique.

2. Le délai : Il y a plusieurs façons de mesurer le délai, avec le temps que le trafique a pris pour aller depuis la source vers la destination est appelé le « one-way delay »

Le temps que le trafique prend pour aller depuis la source vers la destination et retourner est appelé le « two-way delay »

3. Jitter : est la variation du one-way delay entre le paquet envoyé par la même application. Les téléphones IP ont un « jitter buffer » pour fournir un délai fixe aux paquets audio.

4. Perte : est le pourcentage de paquets envoyés qui n'atteignent pas leurs destinations, cela peut être dû à des câbles endommagés.

Peut aussi être causé lorsque la queue de paquets d'un appareil est pleine et que l'appareil commence à bloquer la réception des paquets.

Il y a plusieurs standard recommandés pour une qualité audio correcte (par exemple les appels audio) :

Le One-way delay : doit être de 150ms ou moins

Le Jitter : de 30ms ou moins

La perte : de 1% ou moins

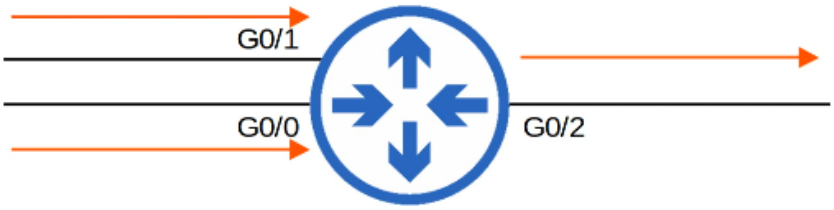
Si ces standard ne sont pas remplis, il peut y avoir une réduction de la qualité de l'appel audio.

Si le réseau reçoit des messages plus rapidement qu'il ne les repartage à l'interface approprié, les messages sont alors placés dans une queue.

Par exemple sur ce routeur, le trafique est réceptionné plus rapidement depuis les interface G0/1 et G0/0 pour être redirigé vers l'interface G0/2

Par défaut, un message en queue est redirigé dans un First In First Out (FIFO)

Les messages sont envoyés dans l'ordre duquel ils sont reçus. Il n'y a pas de traitement spécial, pour le type de trafique, la queue peut alors devenir pleine et les nouveaux paquets peuvent être perdus, ceci est appelé le tail drop.



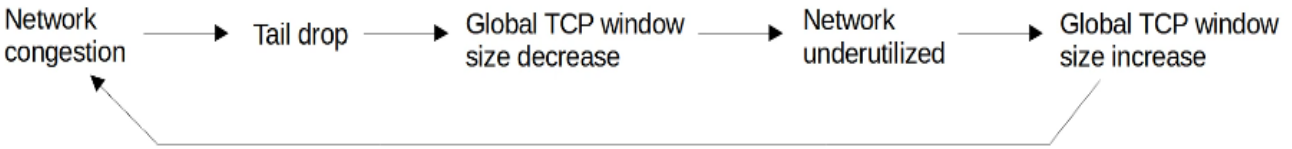
Tail drop peut être problématique car cela peut mener à TCP global synchronisation.

Les hôtes qui utilisent le « sliding window » augmentent ou diminuent le taux duquel ils envoient le trafique en fonction du besoin.

Lorsqu'un paquet est perdu il sera alors retransmit. Lorsqu'une perte se passe, l'envoyeur réduira le taux de bande passante à l'interface auquel il envoie le trafique, il augmentera ensuite petit à petit le taux de bande passante à nouveau.

Lorsque la queue est pleine et les tails drop se passent, tous les hôtes TCP qui envoient un trafique vont ralentir le taux de bande passante auquel ils envoient le trafique.

Ils vont ensuite augmenter le taux duquel ils envoient le trafique qui conduira rapidement à plus de congestion, plus de perte de paquets, et ce même processus recommencera à nouveau.



Voici un schéma qui résume le processus :

Une solution qui permet d'empêcher le tail drop et le TCP global synchronisation est le Random Early Detection (RED), lorsque le montant de trafique dans la queue atteint un certain seuil, l'appareil commence à envoyer aléatoirement des paquets perdus depuis le flux TCP.

Ces flux TCP qui perdent des paquets vont réduire le taux auquel le trafique est envoyé, mais cela provoquera la TCP synchronisation, dans lesquelles tous les flux TCP se réduisent et puis augmentent le taux de transmissions au même moment dans des ondes.

Dans le standard RED, tous les types de trafiques sont traités de la même manière.

Dans une version amélioré appelé le Weighted Random Early Detection (WRED) permet de contrôler quelles paquets sont perdus en dépendant de la classe de trafique.

Cours 47 : QoS (Partie 2)

Dans ce cours nous verrons la seconde partie de l'étude de la qualité de service, en Anglais Quality Of Service (QoS), dans la première partie nous avons vu la voix à travers les VLANs, le PoE, ainsi que l'intérêt d'utilisation du QoS qui est pour prioriser le trafique de la voix et la vidéo afin de réduire les délais, le jitter et les pertes. Dans ce cours détaillerons plus le fonctionnement de QoS.

Nous verrons tout d'abord la classification et les classification et le marquage, nous verrons ensuite la gestion des queues et des congestions puis les politique de conception pour savoir quelle taux du trafique faire entrer ou sortir d'une interface.

L'intérêt de QoS est avant tout de pouvoir donner une priorité à certains types de trafique par rapport à d'autres lors des congestions. La classification organise le trafique réseau (les paquets) dans des classes de trafiques (avec des catégories).

La classification est fondamental à QoS, car pour donner une priorité à certains trafiques il faut identifier à quelle type de trafique donner la priorité.

Il existe plusieurs méthode de classification du trafique, en voici plusieurs exemples :

- Une ACL permet à ce que le trafique soit permis par l'ACL pour lui donner un certain traitement par rapport à d'autre trafiques.

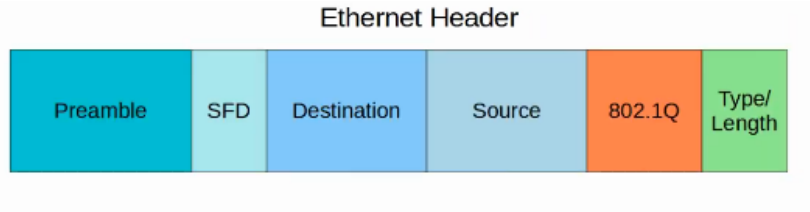
- NBAR (Network Based Application Recognition) fait fonctionner une inspection profonde des paquets, en regardant à travers les couches 3 et 4 jusqu'à la couche 7 pour identifier le type spécifique de trafique.

- Dans les couches 2 et 3 les entêtes ont une partie spécifique utilisé pour cela.

Le PCP (Priority Code Point) est une partie de la balise 802.1Q (dans l'entête Ethernet) et peut être utilisé pour identifier la priorité du trafique si haute ou basse. Cela ne fonctionne seulement lorsqu'il y a une balise dot1q.

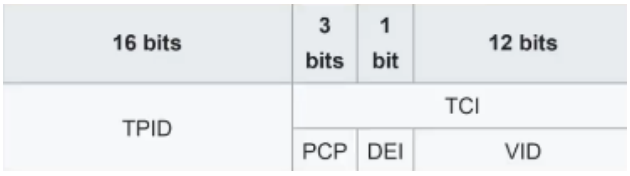
Le DSCP (Differentated Services Code Point) est une partie de l'entête IP qui peut aussi être utilisé pour identifier la priorité du trafique si elle est haute ou basse.

Voyons plus en détail ces méthodes de classifications :



Ci dessus l'entête Ethernet, pour PCP on peut voir qu'il y a la balise dot1q et une balise VLAN.

Le PCP est contenu dans cette partie 802.1Q, on peut voir ici les parties de la balise dot1q :



PCP est aussi connu comme CoS (Class of Service), son utilisation est défini par le standard IEEE 802.1p, on peut voir ci dessus qu'il y a 3 bits qui donnent 8 valeurs possibles (car $2^3 = 8$)

Ces valeurs sont définies en fonction du tableau suivant :

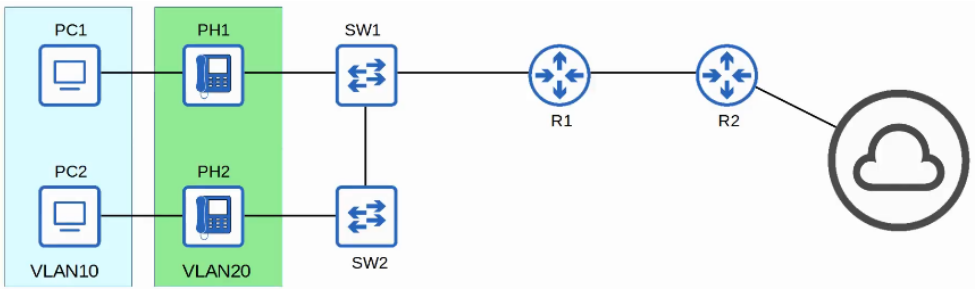
Valeur PCP	Type de trafique
0	Best Effort
1	Background
2	Excellent Effort
3	Critical Applications
4	Vidéo
5	Voice

7	Contrôle Réseau

« Best effort » signifie qu'il n'y a pas de garantie que la donnée est bien acheminé ou que cela correspond à un standard QoS. C'est un trafique régulier et non pas une haute priorité.

Les téléphones IP marque les trafiques appels signalés (utilisés pour établir des appels) comme PCP3. Ils marquent le trafique de voix actuel avec PCP5.

Voici un réseau dans lequel est réparti deux VLAN, une pour le trafique des données PC (VLAN 10) et une autre VLAN pour la voix (VLAN20)



Puisque le PCP est trouvé dans l'entête dot1q, il peut uniquement être utilisé avec les connexions suivantes :

- Les liens Trunk

- Les liens d'accès avec un VLAN de voix

Dans le diagramme ci dessus, le trafique entre R1 et R2 ou entre R2 et une destination externe ne sera pas balisé avec dot1q. Le trafique à travers ces liens PCP ne peuvent pas être marqués avec une valeur PCP.

Voyons à présent comment le marquage et la classification est faite dans la couche 3 :

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

Le marquage est contenu dans les parties DSCP et ECN. Auparavant ces bits étaient organisés différemment avec ToS, 3 bits étaient utilisé pour le IPP (IP préférence) et 5 bits pour différents usages sans intérêt défini à présent c'est 5 bits pour DSCP et 2 bits pour ECN.

Voyons plus en détail IPP :

Le marquage avec le standard IPP est similaire à PCP :

- les bits 6 et 7 sont réservés pour le « contrôle trafique du réseau » (par exemple des messages OSPF entre des routeurs).

- Le bit 5 est pour la voix

- le bit 4 pour la video

- le bit 3 est pour le voice signaling

- le bit 0 est pour le best effort

Avec les bits 6 et 7 réservés, 6 valeurs possibles sont affichés.

Bien que 6 valeurs est suffisant pour plusieurs réseaux, les prérequis pour le QoS de certains réseaux demandent plus de flexibilité.

Voyons plus en détail DSCP :

Le RFC 2474 (1998) définit la partie du DSCP, et d'autres différents services que le RFC à élaborés sur son usage. Avec la mis à jour de IPP vers DSCP le nouveau marquage du standard à dû être implémenté, le type de marquage a été changé car en ayant des marquages de différents standards pour différents types de trafique, l'implémentation QoS est simplifié, QoS fonctionne mieux entre les ISP et les entreprises avec d'autres bénéfices.

Différents marquages ont donc été crées et standardisés, en voici quelques uns :

- le Default Forwarding (DF) – Best effort trafic

- le Expedited Forwarding (EF) – bas trafique perte/latence/jitter (la voix)

- Assured Forwarding (AF) – un ensemble de 12 valeurs de standard

- Class Selector (CS) – un ensemble de 8 valeurs de standard, qui fournit une compatibilité avec IPP

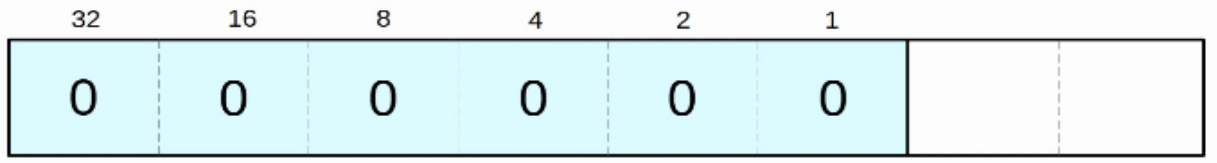
On configure un class-map appelé TEST avec la commande :

```
R1(config)#class-map TEST
```

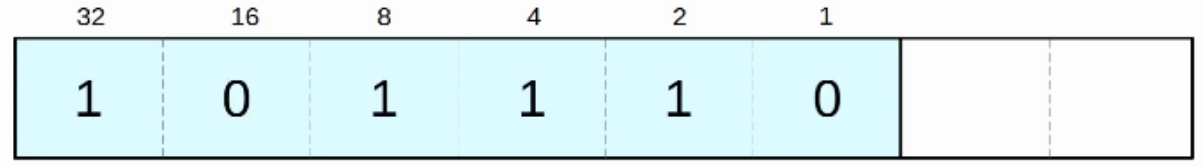
```
R1(config)#class-map TEST
R1(config-cmap)#match dscp ?
  <0-63>   Differentiated services codepoint value
  af11     Match packets with AF11 dscp (001010)
  af12     Match packets with AF12 dscp (001100)
  af13     Match packets with AF13 dscp (001110)
  af21     Match packets with AF21 dscp (010010)
  af22     Match packets with AF22 dscp (010100)
  af23     Match packets with AF23 dscp (010110)
  af31     Match packets with AF31 dscp (011010)
  af32     Match packets with AF32 dscp (011100)
  af33     Match packets with AF33 dscp (011110)
  af41     Match packets with AF41 dscp (100010)
  af42     Match packets with AF42 dscp (100100)
  af43     Match packets with AF43 dscp (100110)
  cs1      Match packets with CS1(precedence 1) dscp (001000)
  cs2      Match packets with CS2(precedence 2) dscp (010000)
  cs3      Match packets with CS3(precedence 3) dscp (011000)
  cs4      Match packets with CS4(precedence 4) dscp (100000)
  cs5      Match packets with CS5(precedence 5) dscp (101000)
  cs6      Match packets with CS6(precedence 6) dscp (110000)
  cs7      Match packets with CS7(precedence 7) dscp (111000)
  default  Match packets with default dscp (000000)
  ef       Match packets with EF dscp (101110)
```

Voyons plus en détail chacunes de ces parties.

DF (Default Forwarding) est utilisé pour le trafique best-effort, le marquage DSCP pour DF est 0

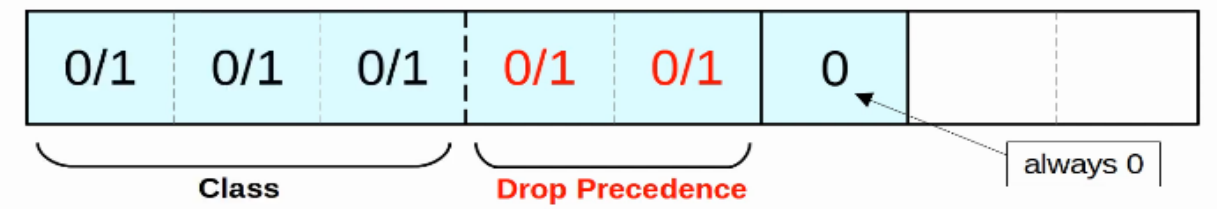


EF (Expedited Forwarding) est utilisé pour le trafique qui requière de basse perte/latence/jitter, le marquage DSCP pour EF est 46 comme suit :



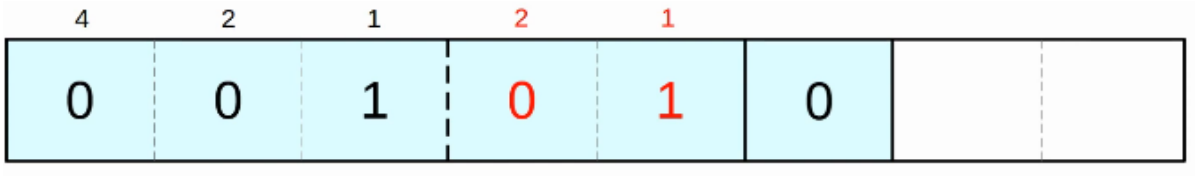
AF (Assured Forwarding) définit 4 classes de trafique. Tous les paquets dans une classe ont la même priorité. Pour chaque classe, il y a 3 niveaux de perte précédente :

- Haute perte précédente = plus de chance de perte de paquet durant la congestion



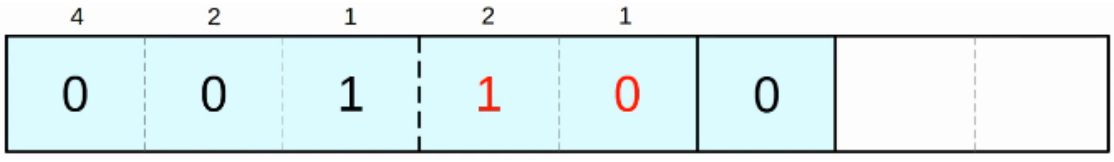
Lorsque l'on écrit la valeur de AF on note AF suivi du numéro décimal de la classe suivi du numéro décimal de la perte précédente.

Par exemple pour noter l'AF du nombre suivant :



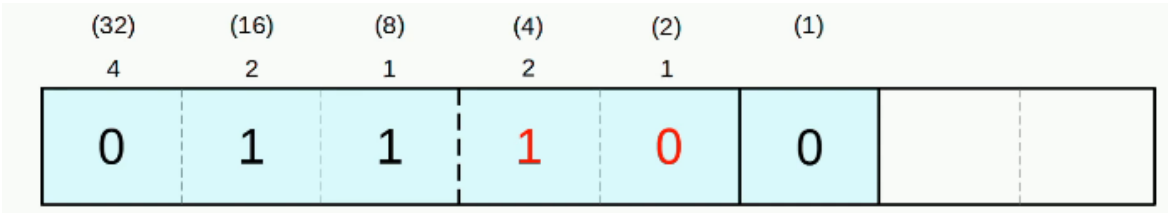
On écrira « AF11 », AF11 est aussi la même valeur de DSCP 10, pour calculer le DSCP on additionne les valeurs qui contiennent un bit à 1

Pour noter la valeur AF du nombre suivant :



On écrira « AF12 », AF12 est aussi la même valeur que DSCP 12

Pour noter la valeur AF du nombre suivant :



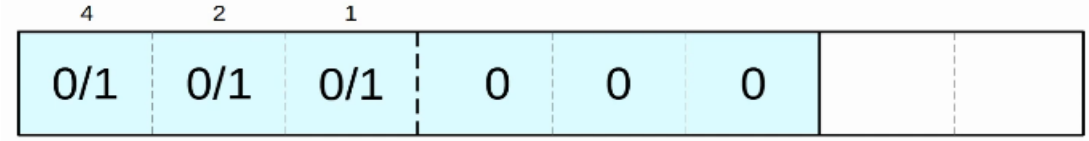
On écrira « AF32 », AF32 est aussi la même valeur de DSCP 28, pour calculer le DSCP on additionne les valeurs qui contiennent un bit à 1

La formule pour convertir de AF vers DSCP est 8X + 2Y où X est la valeur de la classe et Y la valeur de la perte précédente.

Voici un résumé des différentes valeurs à attribuer en fonction de la priorité, la valeur AF41 est la valeur la plus haute avec la plus grande priorité, AF13 est la valeur la plus basse avec la plus basse priorité

CS (Class Selector) définit 8 valeurs DSCP pour la compatibilité arrière avec IPP

Les 3 bits ajoutés pour DSCP sont définis à 0, et le IPP originel est utilisé pour faire 8 valeurs.



Donc CS est similaire à IPP, la différence est dans le nom des valeurs, pour IPP il y a les valeurs 0, 1, 2, 3 ,4, 5, 6, 7 pour CS il y a les valeurs : CS0, CS1, CS2, CS3, CS4, CS5, CS6, CS7

Le RFC 4954 a été développé avec l'aide de Cisco pour associer ces valeurs afin de standardiser leur utilisations.

Le RFC donne plusieurs recommandations, les plus importantes sont les suivantes :

- trafic de voix : EF

- Video Interactive : AF4X

- Streaming Vidéo : AF3X

- Données de haute priorité : AF2X

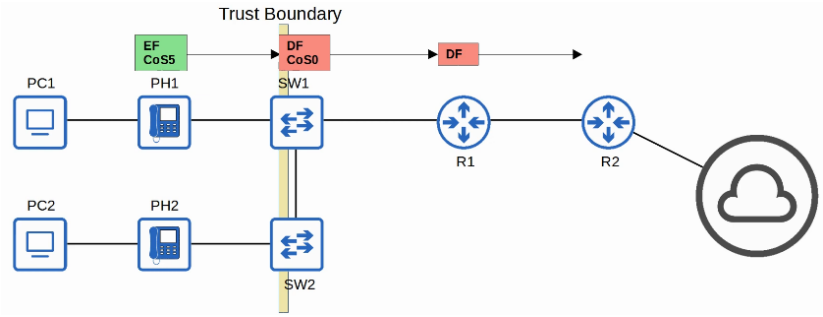
- Best effort : DF

Le trust boundary d'un réseau définit où les appareils font confiance ou non au marquage QoS pour recevoir des messages.

Si le marquage est de confiance, l'appareil va transférer le message sans changer son marquage.

Si le marquage n'est pas de confiance, l'appareil va changer le marquage en accordement avec la politique de configuration.

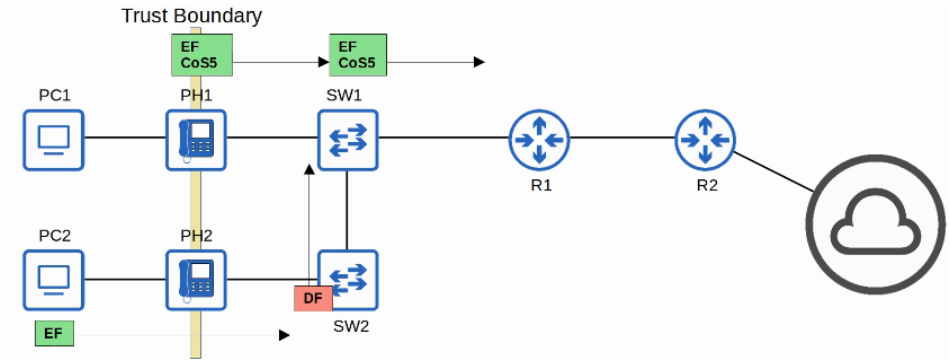
Sur le réseau suivant le trust boundary est placé au niveau du switch, les messages avant le Switch sont de confiance donc le marquage est EF mais après le Switch le marquage n'est plus de confiance donc les messages sont DF



Si un téléphone IP est connecté au port Switch il est recommandé de bouger le trust boudary vers les téléphones IP. Cela ce fait avec la configuration.

Si un utilisateur marque son PC avec une priorité haute, le marquage changera (en pas confiance)

Donc sur le réseau suivant, le trust boundary est placé au niveau des téléphones et les messages des téléphones sont toujours de type EF et ne changent pas, par contre les messages des PC sont de type EF au départ et changent lorsqu'ils passent le trust boundary vers des messages de type DF.



Voyons le concept de gestion des congestions et des fil d'attente.

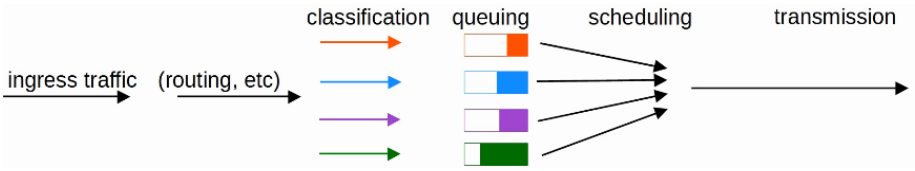
Lorsque l'appareil d'un réseau reçoit un trafique plus rapidement qu'il ne peut le repartager vers l'interface approprié, les paquets sont placés dans une fil d'attente de l'interface et attendent d'être repartagés. Lorsque la fil d'attente est remplie, les paquets qui ne peuvent plus entrer dans la fil d'attente dont bloqués (tail drop). RED et WRED lache les paquets plus tôt afin d'éviter le tail drop.

Une partie essentiel de QoS est qu'il peut utiliser plusieurs fil d'attente et non pas une seule. C'est à ce moment que la classification prend son sens. L'appareil qui correspond à certains facteurs (par exemple le marquage

DSCP de l'entête IP) et le place dans la fil d'attente approprié. L'appareil, est le seule capable de partager la trame en dehors d'une interface donc un planificateur est utilisé pour décider quelle fil d'attente est partagé depuis le suivant.

La priorisation permet aux planificateurs de donner à certaines files d'attentes une plus grande priorité par rapport à d'autres.

Voici un schéma pour simplifier cela :

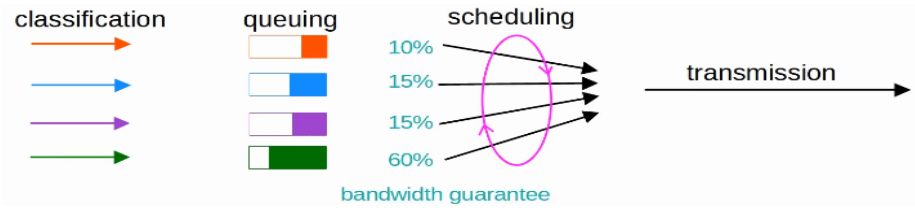


Une méthode de planification commune est le weighted round-robin

- le round robin les paquets sont pris depuis chaque fil d'attente dans l'ordre et de manière cyclique.

- weighted signifie que plus de données sont pris à partir de la plus haute priorité de fil d'attente chaque fois que le planificateur atteint la fil d'attente.

CBWFQ (Class-Based Weighted Fair Queuing) est une méthode populaire de planification qui utilise le planificateur weighted round-robin tant que la garantie de chaque fil d'attente à un certain pourcentage de bande passante pour l'interface durant la congestion.



Le schéma est à présent le suivant :

Le roud robin schedulling n'est pas idéal pour le trafique de voix et vidéo. Même si le trafique de voix et vidéo reçoit une garantie minimal du montant de la bande passante, round robin peut prendre un certain délai ainsi que du jitter puisque même avec une haute priorité la fil d'attente doit attendre leur tour dans le planificateur.

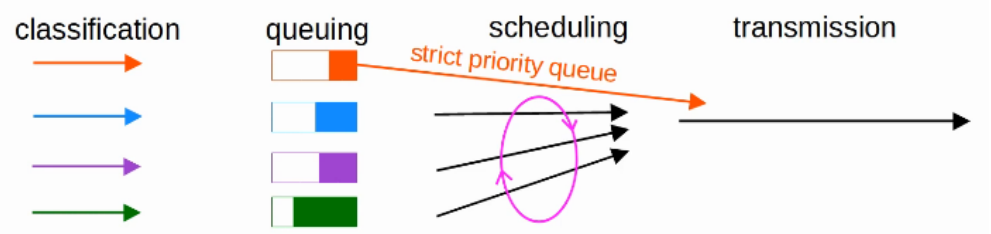
LLQ (Low Latency Queuing) désigne une (ou plus) de fil d'attente comme fil d'attente de priorité strict. Cela signifie que s'il y a un trafique dans la fil d'attente, le planificateur attendra toujours de prendre le paquet suivant depuis la fil d'attente jusqu'à ce qu'elle soit vide.

Cela est très efficace pour réduire le délai et jitter de la voix et du trafique vidéo.

Il y a tout de même l'inconvénient de laisser les autres fil d'attentes si le trafique est toujours désigné comme fil d'attente de strict priorité.

La politique peut contrôler le montant du trafique permis dans la fil d'attente de priorité strict donc cela peut prendre tous les liens de la bande passante.

Voici un schéma qui résume le fonctionnement :



Le shaping et policing (façonnage et politique en français) sont utilisé pour contrôler le taux du trafique.

Le shaping ajoute en mémoire de la fil d'attente le trafique si le taux de trafique va au dessus du trafique configuré.

Le policing bloque le trafique si le taux de trafique va au dessus du taux configuré.

Le trafique « rafale » est permis pour une courte période de temps si le taux est au dessus de celui configuré.

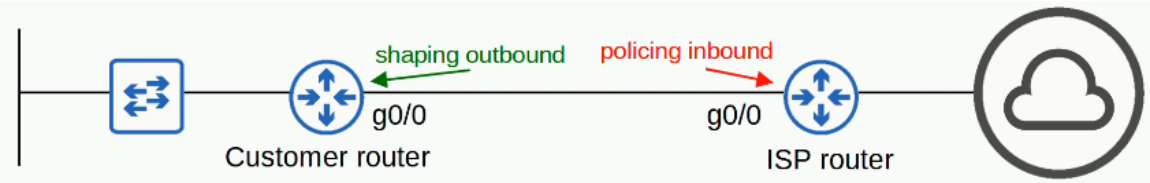
Cela est pratique aux données des application qui sont en « rafale » de nature. Au lieu d'un direct constant de données, elles envoient des données en rafales.

Le montant du trafique en rafale permis est aussi configurable.

Dans les deux cas, la classification peut être utilisé pour permettre différents taux pour différents type de trafique.

Pourquoi limiter le trafique en fonction de l'envoi/Réception ?

Pour comprendre on peut utiliser le réseau suivant :



Si le routeur ISP réceptionne et est limité à 300Mbps par politique de ISP, le routeur du client va envoyer lui en rafale 300Mbps pour éviter que ses paquets ne soient perdus par le routeur de l'ISP.

Cours 48 : Security Fundamentals

Dans ce cours nous verrons les fondamentaux de la sécurité informatique.

Nous verrons tout d'abord les concepts clés de la sécurité puis des types d'attaques communes qui visent les entreprises, nous verrons ensuite le concept des mot de passes et de l'authentification multifacteur ou Multi-facteur Authentication (MFA) ensuite nous verrons le concept de Authentification, Authorization, Accounting (AAA). Nous verrons également les éléments d'un programme de sécurité.

Répondons tout d'abord à la question : Quelle est le but de la sécurité dans une entreprise ?

Le principe de la triade CIA forme les fondations de la sécurité :

- Confidentiality (Confidentialité) : seulement les utilisateurs autorisés peuvent avoir accès à la donnée, certaines informations sont publiques et peuvent être accédés par n'importe qui, certaines sont secrètes et doivent uniquement être accessible par des personnes spécifiques.

- Intégrity (Intégrité) : Les données ne peuvent pas être altérés (modifiés) par des utilisateurs non autorisés. Les données doivent rester correct et authentique.

- Availability (Disponibilité) : Le réseau/système doit être opérationnel et accessible aux utilisateurs autorisés. Les attaquants peuvent menacer la confidentialité, l'intégrité et la disponibilité des informations du système d'une entreprise.

Il y a aussi certains concept fondamentaux qu'il faut comprendre :

- Vulnérabilité : il s'agit d'une faiblesse potentiel qui peut compromettre le CIA d'un système/info

Une faiblesse potentiel n'est pas un problème en tant que tel, les fenêtres d'une maison sont par exemple des faiblesses potentiel et peuvent être utilisés et exploités des voleurs.

- Exploit : il s'agit d'une chose qui peut potentiellement être utilisé pour exploiter une vulnérabilité.

Une chose qui peut potentiellement être utilisé comme exploit n'est pas un problème en tant que tel.

Par exemple une pierre peut exploiter la faiblesse d'une fenêtre et peut être utilisé pour entrer dans une maison, mais les pierres ne sont pas des problèmes en soit.

- Threat (Menace) : est le potentiel pour être vulnérabilité d'être exploité, par exemple le voleur est la menace qui pourrait utiliser la pierre pour casser la fenêtre et entrer dans la maison.

Un Hacker qui exploite une vulnérabilité du système est un Threat ou menace.

- Mitigation technique : est quelque chose qui peut protèger des menaces Peut être implémenté de partout où une vulnérabilité peut être exploité : les appareils clients, les serveurs, les Switchs, Les routeurs, Murs de feu, etc...

Un système n'est jamais parfaitement protégé, il existe toujours des menaces.

Voyons quelques menaces qui peuvent potentiellement exploiter des vulnérabilités pour compromettre la confidentialité, l'intégrité ou la disponibilité, CIA du système d'information d'une entreprise :

- Attaques DoS (Denial of Service)

- Attaques par Spoofing

- Attaque par Reflection/Amplification

- Attaque de Man in the middle

- Attaque de Reconnaissance

- Malware

- Attaque par Social Engineering

- Attaque de Mot de passe

Il existe bien plus d'attaques potentiels que celles ci mais les principales sont celles ci, voyons plus en détail chacune d'elles :

- Denial of Service (DoS) :

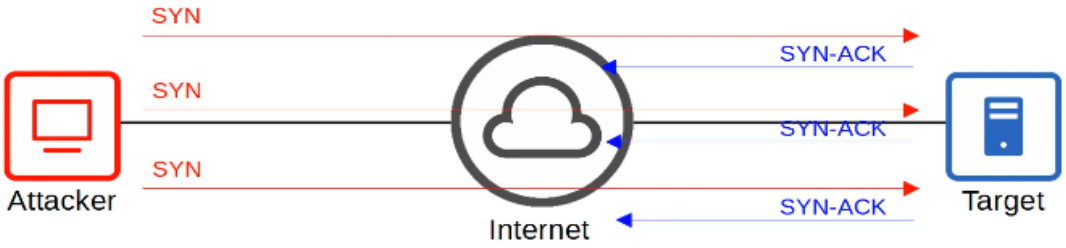
les attaque DoS menacent la disponibilité d'un système. Une attaque commune DoS est l'inondation TCP SYN flood qui exploite le three way handshake : SYN | SYN-ACK | ACK

Dans une inondation SYN, l'attaquant envoie un certains nombre de messages TCP SYN vers la cibles. La cible envoie des messages SYN-ACK en réponse pour chacun des SYN qu'il reçoit.

L'attaquant ne répond jamais avec le Ack final du TCP three way handshake.

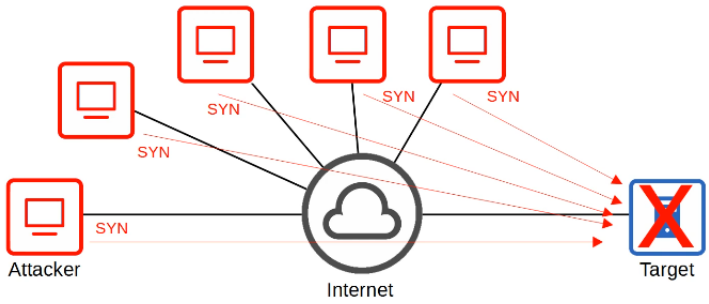
La connexion incomplète remplie la table de connexion TCP de la cible, celle ci sera alors en timeout et sera supprimé de la table après une certaine période de temps, mais l'attaquant continuera d'envoyer des messages SYN pour remplir la table. La cible ne sera plus capable de rendre la connexion TCP légitime.

Le schéma suivant résume la situation :



Ce type d'attaque n'est généralement pas réalisé par un seule attaquant, un type d'attaque beaucoup plus puissant est l'attaque DDoS.

Dans une attaque DDoS (Distributed Denial Of Service) l'attaquant infecte plusieurs ordinateurs cibles avec des malware et les utilise pour initier des attaques par déni de service par exemple des attaques TCP SYN flood. Ce groupe d'ordinateurs infecté est appelé botnet.



Le schéma suivant résume la situation :

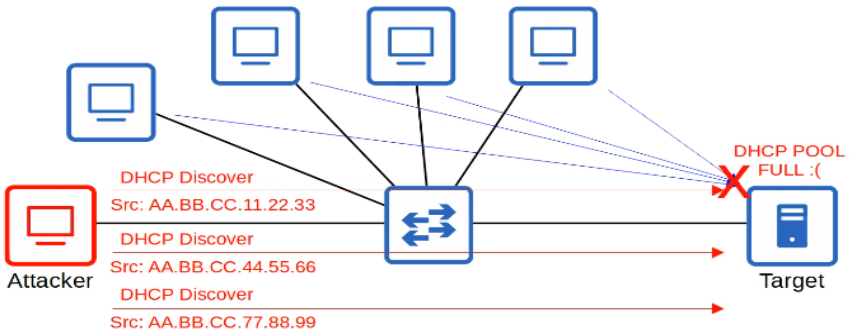
- Attaque Spoofing :

Le spoof d'une adresse est pour utiliser une fausse adresse source (IP ou Adresse MAC)

De nombreuses attaques implique le spoofing ça n'est pas un seul type d'attaque.

L'exemple d'un attaque Spoofing est le DHCP exhaustion. Un attaquant utilise l'usurpation d'adresse MAC pour inonder des messages DHCP Discover, le serveur cible qui est le POOL DHCP devient complet ce qui résulte en un DoS mais vers d'autres appareils.

Ce schéma résume la situation :



- Attaque Reflection/Amplification :

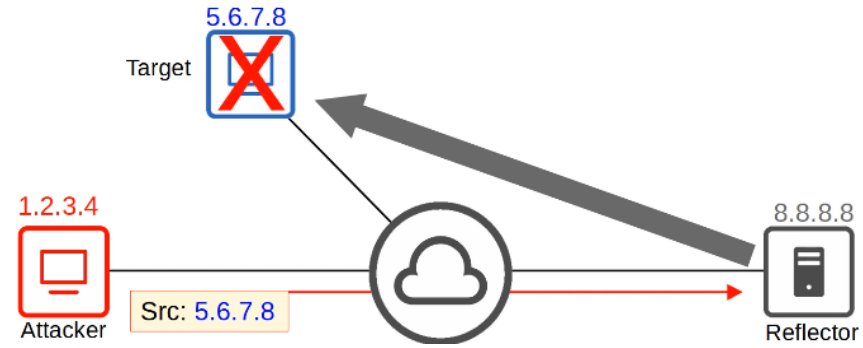
Dans une attaque par réflexion, l'attaquant envoie le trafique vers un réflecteur, et usurpe l'adresse source du paquet en utilisant l'adresse IP d'une cible.

Le réflecteur (Par exemple le serveur DNS) envoie la réponse à l'adresse IP cible.

Si le montant du trafique envoyé à la cible est suffisamment large, cela peut résulter en une attaque DoS. Il existe une forme d'attaque plus puissante appelé attaque par amplification.

Une attaque par réflexion devient une attaque par amplification lorsque le montant du trafique envoyé par l'attaquant est petit mais qu'il déclenche en un large montant de trafique pour être envoyé depuis le réflecteur vers la cible.

Le schéma suivant résume la situation :



- Attaque Man in the Middle :

Dans une attaque Man in the Middle, l'attaquant se place entre la source et la destination pour espionner les communication, ou pour modifier le trafique avant qu'il n'atteigne la destination.

Un exemple commun est le ARP spoofing aussi connu comme ARP poisoning.

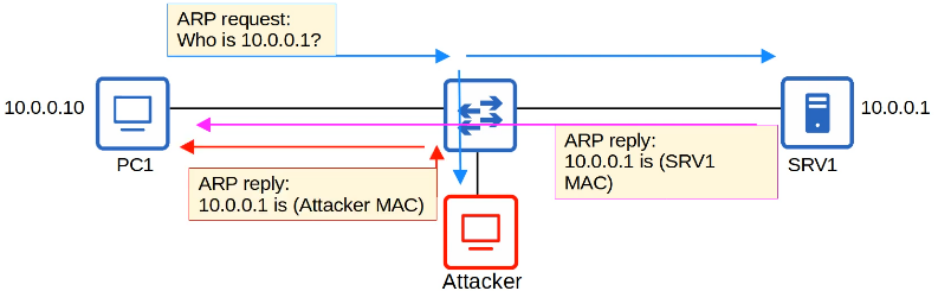
L'hôte envoie une requête ARP, pour demander une adresse MAC d'un autre appareil.

La cible de la requête envoie la réponse ARP pour informer la requête de sa propre adresse MAC.

L'attaquant attend et envoie une autre réponse ARP pour répondre après la réponse légitime.

Si l'attaquant la réponse ARP de l'attaquant arrive en dernier, il écrira par dessus l'entrée ARP dans la table ARP du PC1.

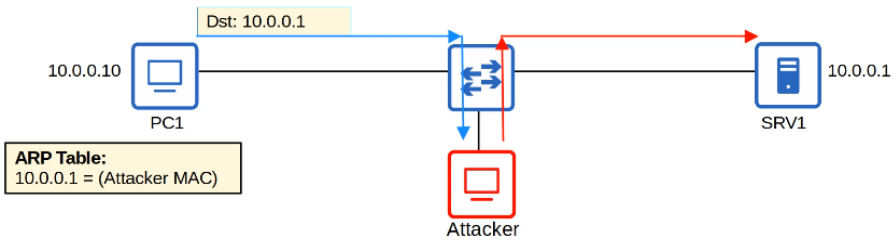
Le schéma suivant résume la situation :



Dans la table ARP du PC1 l'entrée pour 10.0.0.1 aura l'adresse MAC de l'attaquant.

Lorsque le PC1 essayera d'envoyer le trafique vers le SRV1, il le transmettre à l'attaquant, l'attaquant pourra inspecter les messages et les repartager au SRV1. L'attaquant peut aussi modifier les messages avant de les retransmettre au SRV1.

Cela compromet la confidentialité et l'intégrité des communication entre PC1 et SRV1

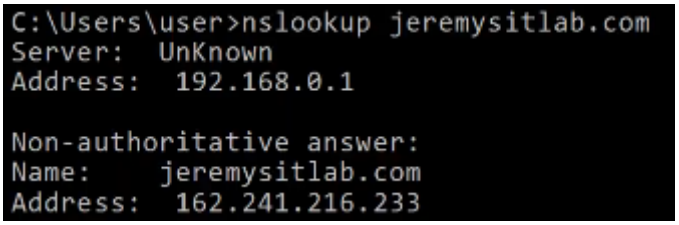


- Attaque par reconnaissance :

Les attaques par reconnaissance ne sont pas des attaques en elles même mais elles sont utilisés pour récupérer des informations à propose de la cible et peuvent être utilisés pour de future attaques.

C'est souvent des information publiquement disponibles.

Par exemple on peut lancer un nslookup pour apprendre l'adresse IP d'un site web :



A partir de là il est possible de rechercher des ports ouverts qui peuvent être des vulnérabilités potentiels. Une requête whois permet d'apprendre l'adresse mail, le téléphone, l'adresse physique, etc...

- Malware : (Logiciel malicieux) se réfère à une variété de programmes qui peuvent infecter un ordinateur. Les virus infectent d'autre logiciel (ou programme). Le virus se propage puisque le logiciel est partagé par les utilisateurs. Cela corrompe ou modifie les fichier de l'ordinateur cible.

Les worms ne requière pas de programme hôte, ce sont des malware qui sont « standalone » et qui sont capables de se propager par eux même sans l'interaction d'un utilisateur. La propagation peut congestionner le réseau, mais le payload d'un vers peut causer des préjudices supplémentaires aux appareils de la cible.

Les cheval de Troie sont des logiciels nuisible qui se distinguent des logiciels légitime. Ils se propagent par l'interaction de l'utilisateur comme avec l'ouverture d'une pièce jointe, ou un téléchargement depuis Internet.

Les types de Malwares précédents peuvent exploiter des vulnérabilités variés pour menacer n'importe quelle CIA de l'appareil cible.

- Attaque par Social Engineering :

Les attaques par Social Engineering ciblent la partie la plus vulnérable d'un système qui est : les utilisateurs. Ils impliquent la manipulation psychologique pour faire que la cible révèle des informations confidentiel ou fasse certaines actions.

Phishing implique des mails frauduleux qui apparaissent comme étant d'un business légitime (Amazon, Banque, Compagnie de carte de crédit, etc...) et contient des liens vers des sites frauduleux qui semblent légitime. Les utilisateurs doivent se connecter au site frauduleux et donnent leurs identifiants à l'attaquant.

Spear phishing est une forme de phishing plus ciblé par exemple qui cible les employés de certaines compagnies.

Le Whaling est un phishing qui cible des individus avec un haut statut. Par exemple le président d'une compagnie.

Le Vishing (Voice Phishing) est un phishing qui se passe à travers le téléphone. Par exemple un individu vous appelle et se dis être l'ingénieur informatique de l'entreprise dans laquelle vous travaillez et vous demande de réinitialiser votre mot de passe.

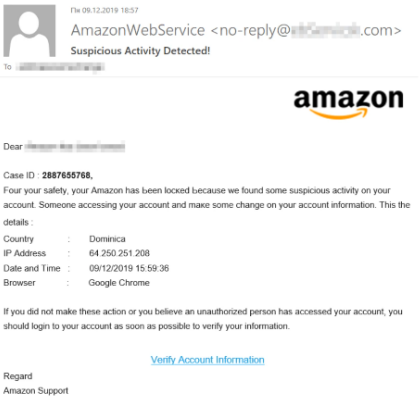
Le Smishing (SMS phishing) est un phishing qui utilise les messages texte SMS.

Les attaques watering hole compromettent les sites que la victime visite fréquemment. Si un lien malicieux est placé sur un site web, la victime lui fais confiance et n'hésite pas à cliquer dessus.

Les attaques Tailgating impliquent une entrée restreint, une zone sécurisé en simplement marchant par derrière une personne autorisé lorsqu'elle entre.

Souvent la cible laisse la porte ouverte à l'attaquant pour rester poli et assume que l'attaquant est aussi une personne autorisé à entrer.

Voici un exemple de mail frauduleux qui permettent à l'attaquant de récupérer les identifiants d'un utilisateur :



AmazonWebService <no-reply@[redacted].com>

Suspicious Activity Detected!

To [redacted]

amazon

Dear [redacted]

Case ID : **2887655768,**

Four your safety, your Amazon has been locked because we found some suspicious activity on your account. Someone accessing your account and make some change on your account information. This the details :

Country	:	Dominica
IP Address	:	64.250.251.208
Date and Time	:	09/12/2019 15:59:36
Browser	:	Google Chrome

If you did not make these action or you believe an unauthorized person has accessed your account, you should login to your account as soon as possible to verify your information.

Verify Account Information

Regard

Amazon Support

- Attaque de Mot de passe :

La plupart des systèmes utilisent un identifiant et mot de passe pour authentifier un utilisateur.

Le nom d'utilisateur est souvent simple et facile à deviner (par exemple le nom d'utilisateur d'une adresse mail) et la résistance et le secret d'un mot de passe est utilisé pour fournir assez de sécurité.

Les attaquants peuvent apprendre le mot de passe utilisateur par plusieurs méthodes :

En le devinant, avec une attaque par dictionnaire qui est un programme qui lance un dictionnaire ou liste des mot pour trouver le mot de passe de la cible.

L'attaque par Brute force est un programme qui essaye toutes les possibilité de combinaison de lettre, de nombres et de caractère spéciaux pour trouver le mot de passe de la cible.

Un mot de passe puissant doit contenir : au moins 8 caractères (De préférence plus) un mélange de lettres majuscules et minuscules, au moins un caractère spécial.

Il doit être changé régulièrement.

Voyons le concept de l'authentification multi-facteur, ou Multifactor Authentication (MFA) en Anglais. Le MFA implique plus que juste un nom d'utilisateur/mot de passe pour prouver son identité. Cela implique de fournir 2 des choses suivantes :

- Une chose que l'on sais : par exemple un nom utilisateur, un mot de passe, un PIN, etc...

- Une chose que l'on possède : par exemple en confirmant une notification qui apparaît sur téléphone, un badge devant être scanné, etc...

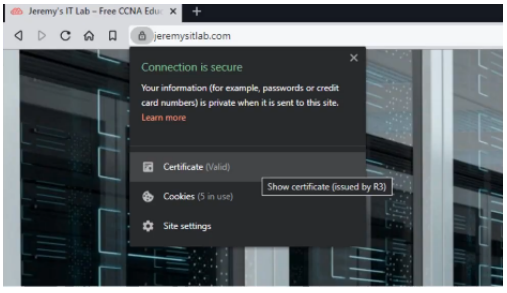
- Une chose que l'on est : un accès biométrique comme le scan du visage, le scan de la main, le scan de l'empreinte digitale, le scan de la rétine, etc...

Lorsque l'authentification multi-facteur est requise pour connexion la sécurité est grandement amélioré. Même si l'attaquant apprend le mot de passe (une chose que l'on sais) il ne pourra pas se connecter au compte.

Une autre forme d'authentification implique des certificats digitales qui sont utilisés pour prouver l'identité du porteur du certificat.

Ils sont utilisés pour les sites web pour vérifier que le site web est légitime. Les entités qui veulent un certificat pour prouver leurs identité envoient un CSR (Certificate Signing Request) au CA (Certificate Authority) qui va regénérer et signer le certificat.

On peut voir le certificat avec le symbole du cadenas qui indique que le site est sécurisé et légitime.



Le AAA (triple A) est l'acronyme de Authentication, Autorization et Accounting

C'est un cadre pour contrôler et gérer les utilisateurs de l'ordinateur du système (par exemple un réseau)

- Authentication (Authentification) est la procédure de vérification de l'identité utilisateur par le moyen de l'enregistrement.

- Authorization est la procédure qui accorde les permissions d'accès à un utilisateur.

Il accorde l'accès utilisateur à certains fichiers/services la restriction d'accès pour entrer dans un fichier/services est l'autorisation

- Accounting est la procédure d'enregistrement de l'activité de l'utilisateur dans le système.

Par exemple lorqu'un utilisateur fais le changement d'un fichier

Les entreprises utilisent des serveurs AAA pour fournir des services AAA

ISE (Identity Services Engine) est un serveur Cisco AAA

Les serveurs AAA supportent les protocoles AAA suivants :

- RADIUS : un protocole standard ouvert. Il utilise les ports UDP 1812 et 1813

- TACACS+ : un protocole Cisco propriétaire. Utilise le port TCP 49

Les éléments de sécurité d'un programme consistent à sensibiliser les employés à propos des des menaces de sécurité potentiel et des risques.

Par exemple une compagnie envoie de faux emails de phishing pour que les employés cliquent et un lien pour qu'ils signent avec leur identifiants.

Bien que les mails ne sont pas nuisibles les employés qui tombent dans le piège du faux mail seront informés par un programme qui leur avertit qu'ils doivent faire plus attention aus mails de phishing.

Les programmes d'entraînement des utilisateurs sont plus formateurs que les programmes de sensibilisation. Par exemple, un entraînement dédié qui éduque l'utilisateur sur la politique de sécurité et comment créer un mot de passe solide pour éviter les menaces potentiels.

Les contrôle d'accès physique protègent les équipement et les données d'attaques potentiels en autorisant seulement des utilisateurs permis dans une zone protégé comme un réseau privée ou le un data center.

Les verrous multi-facteurs peuvent protéger l'accès à des zones restreintes, par exemple une porte qui requière un badge et un contrôle de l'empreinte digitale pour entrer.

Les permissions d'un badge peuvent facilement être changés par exemple les permissions peuvent être supprimés lorsque l'employé quitte la compagnie.

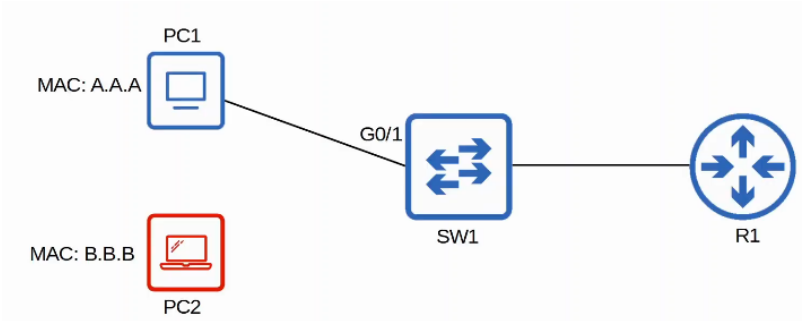
Cours 49 : Port Security

Dans ce cours nous verrons le fonctionnement de Port Security qui est une fonctionnalité des Switch Cisco qui permet de contrôler les adresses MAC source qui sont autorisé dans un port Switch.

Nous ferons d'abord une introduction de ce qu'est port security, nous verrons ensuite pourquoi utiliser port security quelle est l'intérêt de l'utiliser, puis nous verrons comment configurer port security.

Port security est une fonctionnalité intégré dans les switch Cisco qui permet de contrôler quelle adresse MAC source est permise pour entrer dans un port switch. Par exemple si une adresse MAC source non autorisé entre par le port, une action sera prise par le switch.

L'action par défaut est de placer l'interface en état : « err-disabled »



Par exemple sur le réseau suivant :

Le switch est configuré pour qu'il n'accepte que les adresses MAC du PC1 : A.A.A qui entrent dans l'interface G0/1

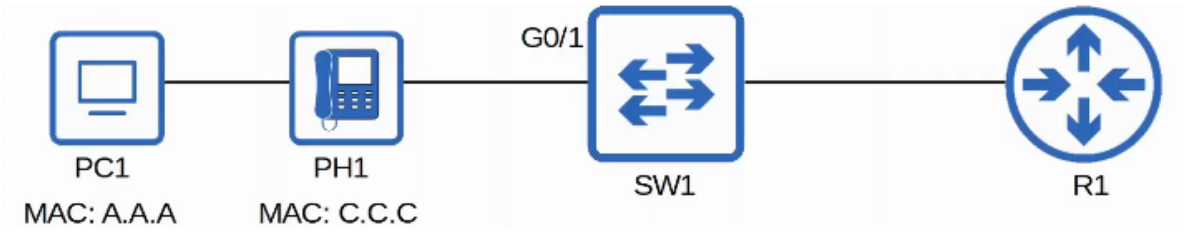
A présent imaginons que le câble du PC1 soit déconnecté et que ce soit le PC2 qui soit connecté à cette même interface, le SW1 va bloquer l'interface en « err-disabled » puisqu'il se rend compte que l'adresse MAC n'est pas celle du PC1.

Il faudra réactiver l'interface pour que celle ci soit à nouveau active.

Lorsque l'on active port security sur une interface avec les paramètres par défaut, une adresse MAC est autorisé. Il est possible de configurer l'adresse MAC autorisé manuellement.

Si l'on ne configure pas l'interface manuellement le switch va permettre la première source adresse MAC qui entre sur l'interface. Les adresses MAC peuvent être apprises de manière dynamique ou bien manuel.

Il est possible de changer le nombre maximal d'adresses MAC autorisés.



Dans le réseau suivant par exemple, il faut autoriser le plus d'adresses MAC puisque les deux appareils (le téléphone et le PC) ne pourront pas communiquer sinon car ils sont connectés à une seule interface.

Port security permet au réseau admin de contrôler quelles appareils sont autorisés à accéder le réseau.

Le changement d'adresse MAC est une chose simple, il est facile de configurer un appareil pour envoyer une trame avec une adresse MAC source différente.

Au lieu de spécifier manuellement l'adresse MAC à autoriser sur chaque port, port security a l'habilité de limiter le nombre d'adresses MAC autorisés sur une interface.

Par exemple dans le cas d'une attaque DHCP starvation, l'attaquant va faire du spoofing sur de fausses adresses MAC. Le serveur DHCP va assigner des adresses IP à ces fausses adresses MAC ce qui va limiter le pool DHCP.

Les tables d'adresses MAC des switch peuvent aussi devenir pleines à cause de ce genre d'attaques.

En limitant le nombre d'adresses MAC sur une interface peut protéger de ce genre d'attaques.

Voici à présent les commandes à utiliser pour activer port security :

```
SW1(config)#interface g0/1
SW1(config-if)#switchport port-security
Command rejected: GigabitEthernet0/1 is a dynamic port.

SW1(config-if)#do show int g0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
![output omitted]

SW1(config-if)#switchport mode access

SW1(config-if)#do show int g0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access


SW1(config-if)#switchport port-security
SW1(config-if)#
```

```
SW1(config)#interface g0/1
SW1(config-if)#switchport port-security
```

Pour afficher la configuration de l'interface on lance la commande :

```
SW1(config-if)#do show int g0/1 switchport
```

Port security peut être activé uniquement en mode access port ou en port trunk il doit être configuré de manière statique comme access ou trunk avec les statut :

switchport mode access ou *switchport mode trunk*

Les statut : *switchport mode dynamic auto* et *switchport dynamic desirable* ne permettent pas la configuration statique avec port security.

Pour activer le mode statique on lance la commande :

```
SW1(config-if)#switchport mode access
```

Le statut de l'interface a à présent changé en statique, il est donc possible d'activer port security.

On lance donc la commande :

```
SW1(config-if)#switchport port-security
```

La commande *show port-security interface* suivi du nom d'interface, est très utile on peut voir qu'est indiqué si port security est activé sur l'interface le violation mode permet de dire si l'interface doit être désactivé si une adresse MAC externe entre.

On peut aussi voir affiché le nombre maximal d'adresses MAC.

```
SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0
```

Lorsque l'on fait un ping entre deux appareils le résultat de la commande est différent :

```
SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 000a.000a.000a:1
Security Violation Count   : 0
```

A présent lorsque l'on tente de connecter un PC avec une autre adresse MAC à l'interface, le statut de l'interface est à présent différent en secure-shutdown :

```
SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 000b.000b.000b:1
Security Violation Count   : 1
```

```
SW1#show interfaces status

Port      Name               Status       Vlan       Duplex  Speed Type
Gi0/0                        connected    1            auto   auto unknown
Gi0/1                        err-disabled 1            auto   auto unknown
```

L'interface est en statut « err-disabled », pour réactiver cette interface il faudra retirer l'appareil qui cause cela puis lancer les commandes :

```
SW1(config)#interface g0/1
SW1(config-if)#shutdown
SW1(config-if)#no shutdown
```

On peut voir que l'interface est à nouveau activé :

```
SW1#show port-security interface g0/1
Port Security               : Enabled
Port Status                 : Secure-up
Violation Mode              : Shutdown
Aging Time                  : 0 mins
Aging Type                  : Absolute
SecureStatic Address Aging  : Disabled
Maximum MAC Addresses       : 1
Total MAC Addresses         : 0
Configured MAC Addresses    : 0
Sticky MAC Addresses        : 0
Last Source Address:Vlan    : 0000.0000.0000:0
Security Violation Count    : 0
```

Il existe une autre manière de réactiver une interface, c'est avec « errDisable Recovery »

Cela permet à une interface de se réactiver automatiquement après un certain temps.

```
SW1#show errdisable recovery
ErrDisable Reason            Timer Status
-----------------            --------------
arp-inspection               Disabled
bpduguard                    Disabled
channel-misconfig (STP)      Disabled
dhcp-rate-limit              Disabled
dtp-flap                     Disabled
![output omitted due to length]
psecure-violation            Disabled
security-violation           Disabled
sfp-config-mismatch          Disabled
storm-control                Disabled
udld                         Disabled
unicast-flood                Disabled
vmps                         Disabled
psp                          Disabled
dual-active-recovery         Disabled
evc-lite input mapping fa    Disabled
Recovery command: "clear     Disabled

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:
```

Ici toute les 5 minutes (ou 300 secondes) toutes les interfaces err-disabled sont réactivés automatiquement, mais pour activer cela il faut que la fonction psecure-violation soit activé on lance pour cela la commande :

```
SW1(config)#errdisable recovery cause psecure-violation
```

Pour modifier le temps d'intervalle d'activation entre chaque interface on lance la commande :

```
SW1(config)#errdisable recovery interval 180
```

Lorsque l'on lance show errdisable recovery on peut voir à présent que la fonctionnalité psecure-violation est à présent bien activé :

```
SW1#show errdisable recovery
ErrDisable Reason            Timer Status
-----------------            --------------
![output omitted due to length]
psecure-violation            Enabled
![output omitted due to length]

Timer interval: 180 seconds

Interfaces that will be enabled at the next timeout:

Interface       Errdisable reason       Time left(sec)
---------       -----------------       --------------
Gi0/1          psecure-violation          149
```

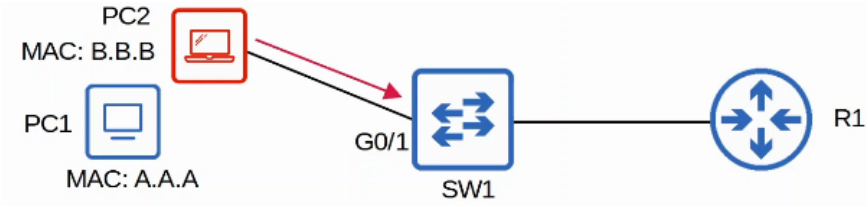
Il y a trois différents modes qui déterminent ce que le switch fais si une trame non autorisé entre dans l'interface configuré avec port security :

- Shutdown : cela éteint l'interface en le plaçant en statut « err-disabled », cela génère aussi un message SNMP lorsque l'interface est désactivé. Le compteur de violation est placé à 1 lorsque l'interface est désactivé.

- Restrict : Le switch va bloquer le trafique des adresses MAC non autorisés, l'interface n'est pas désactivé. Le switch génère un message Syslog ou SNMP chaque fois qu'une adresse MAC non autorisé est détecté. Le compteur de violation s'incrémente à 1 pour chaque trame non autorisé.

- Protect : Le switch bloque le trafique des adresses MAC non autorisés, l'interface n'est pas désactivé dans ce mode. Cela ne génère pas de messages Syslog ou SNMP et cela n'incrémente pas le compteur de violation.

On utilisera le réseau suivant :



Voici les commandes pour configurer le Violation mode restrict :

```
SW1(config-if)#switchport port-security
SW1(config-if)#switchport port-security mac-address 000a.000a.000a
SW1(config-if)#switchport port-security violation restrict

*May 23 22:54:09.951: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC
address 000b.000b.000b on port GigabitEthernet0/1.

SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Restrict
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 000b.000b.000b:1
Security Violation Count   : 12
```

Voici les commandes pour configurer le Violation mode protect :

```
SW1(config-if)#switchport port-security
SW1(config-if)#switchport port-security mac-address 000a.000a.000a
SW1(config-if)#switchport port-security violation protect


SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Protect
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 000b.000b.000b:1
Security Violation Count   : 0
```

Le mode Shutdown est le mode par défaut.

Voyons plus en détail la configuration de port security avec le MAC Adress Aging.

```
SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 000a.000a.000a:1
Security Violation Count   : 0
```

Par défaut l'adresse MAC n'expire pas et est permanente (Aging Time : 0 mins), il est possible de changer le temps avec la commande : *switchport port-security aging time* suivi du temps en minutes. Le aging time par défaut est « absolute » c'est à dire que après que l'adresse MAC est apprise, le aging time commence et l'adresse MAC est supprimé après que le temps expire, même si le switch continue de recevoir des trames de cette adresse MAC source.

Le aging time est en inactivity après que l'adresse MAC sécurisé est apprise, le aging time commence mais est réinitialisé chaque fois qu'une trame de cette adresse MAC source est réceptionné sur cette interface.

Il est possible de changer le type d'aging time avec la commande :

switchport port-security aging time {absolute | inactivity}

l'aging Secure Static MAC (les adresses configurés avec *switchport port-security mac-address x.x.x*) est désactivé par défaut.

Cela peut être activé avec *switchport port-security aging static*

Par exemple on configure l'aging de Secure MAC address avec les commandes suivantes :

```
SW1(config-if)#switchport port-security aging time 30
SW1(config-if)#switchport port-security aging type inactivity
SW1(config-if)#switchport port-security aging static


SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 30 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Enabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 000a.000a.000a:1
Security Violation Count   : 0
```

La commande : *show port-security* affiche quelles interfaces ont port-security activé et plusieurs autres informations :

```
SW1#show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                (Count)       (Count)         (Count)
---------------------------------------------------------------------------
      Gi0/1              1           1                  0         Shutdown
---------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 4096
```

L'apprentissage des « sticky » secure Mac address peut être activé avec les commandes suivantes :

```
SW1(config-if)#switchport port-security mac-address sticky
```

Lorsque activé les adresses MAC dynamiquement apprises sont ajoutés à la running-config comme ceci :

switchport port-security mac-address sticky suivi de l'adresse MAC

Ces adresses MAC « sticky » n'expire jamais.

Il est nécessaire de lancer la running-config vers le startup-config pour les rendre totalement permanent sinon elles ne seront pas gardés lorsque le switch redémarre.

Lorsque l'on lance la commande : *switchport port-security mac-address sticky*

toutes les adresses MAC apprises dynamiquement sont convertis en secure MAC Address sticky.

Lorsque l'on lance la commande : *no switchport port-security mac-address sticky* toutes les adresses MAC sticky apprises sont convertis en secure MAC apprises dynamiquement.

Voici les commandes que l'on lance pour la configuration en mode sticky :

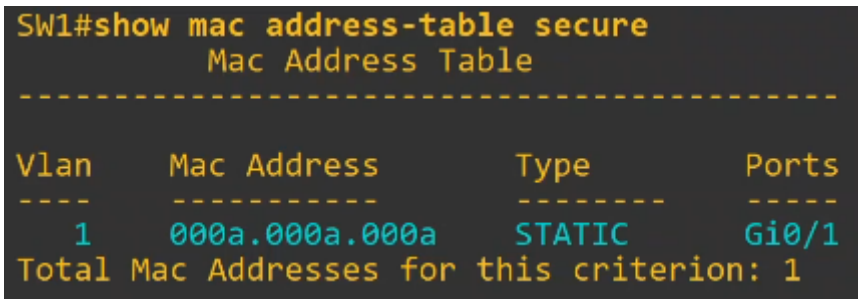
```
SW1(config-if)#switchport port-security
SW1(config-if)#switchport port-security mac-address sticky
SW1(config-if)#do show running-config interface g0/1
!
interface GigabitEthernet0/1
 switchport mode access
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 000a.000a.000a
 switchport port-security
 negotiation auto
```

Les adresses MAC secure sont ajoutés à la table d'adresse MAC comme toutes les autres adresses MAC. Les adresses Sticky et statique secure MAC address ont pour statut STATIC

Les adresses MAC secure dynamiquement apprises ont pour statut DYNAMIC

Il est possible de voir toutes les adresses MAC secure avec la commande :

show mac address-table secure



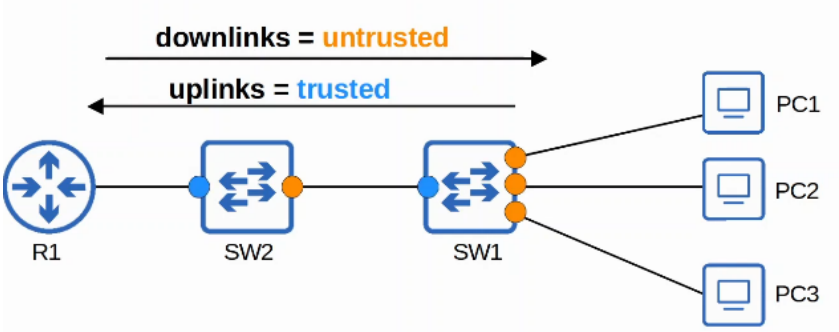
Cours 50 : DHCP Snooping

Dans ce cours nous verrons ce qu'est DHCP Snooping qui est une fonctionnalité disponible sur les Switch Cisco qui aide à protéger des attaques qui prennent avantages du DHCP.

Nous ferons d'abord une introduction de ce qu'est le DHCP Snooping et comment il fonctionne, quelle attaque il permet de contrer et comment le configurer sur un Switch Cisco.

DHCP Snooping est une fonctionnalité de sécurité d'un switch qui est utilisé pour filtrer des messages DHCP sur des ports qui ne sont pas de confiance.

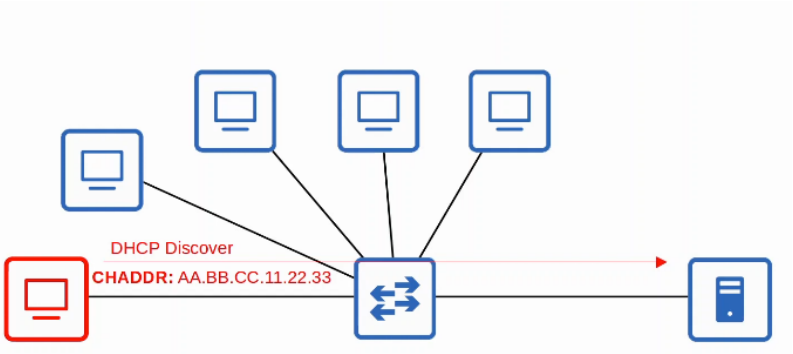
Le Snooping DHCP filtre uniquement des messages DHCP. Les messages qui ne sont pas DHCP ne sont pas affectés. Tous les ports ne sont pas de confiance par défaut, il faut ensuite configurer quelles ports faire confiance. Les ports « uplink » sont configurés comme port de confiance et les ports « downlink » ne sont pas de confiance. « downlink » fait référence aux ports d'une interface qui vont dans la direction de l'hôte. « uplink » fait référence aux ports ports qui partent à partir de l'hôte. Par exemple sur le réseau suivant, en bleu les ports uplink et en orange les ports downlink :



Par exemple dans le cas ou le PC1 envoie un message DHCP, le port étant de confiance est bien réceptionné sur le SW1, le SW2 réceptionne ensuite le message sans le vérifier car il est réceptionné par un port de confiance.

A présent si dans le cas ou le port du PC2 n'est pas de confiance, si le PC2 envoie un message DHCP celui ci ne sera pas repartagé par le Switch.

Un exemple d'une attaque par DHCP est appelé « DHCP starvation » ou aussi appelé « DHCP exhaustion » dans ces attaques un attaquant utilise une fausse adresse MAC pour inonder le Switch de messages DHCP discover. Le serveur DHCP pool ciblé devient rempli ce qui résulte d'un denial of service sur d'autres appareils.



Comme dans l'exemple du réseau suivant :

Dans les messages DHCP envoyé il y a aussi une entête dans laquelle est contenu le CHADDR (Client Hardware Address) qui permet d'indiquer l'adresse MAC du client.

Cette entête est nécessaire dans le cas ou un message est envoyé vers une destination externe au réseau local, par exemple dans le cas ou le serveur DHCP est dans un réseau externe et que le message DHCP est partagé par un relais DHCP lorsque le serveur reçoit cette trame par le client, l'adresse MAC source de la trame ne sera pas l'adresse MAC du client.

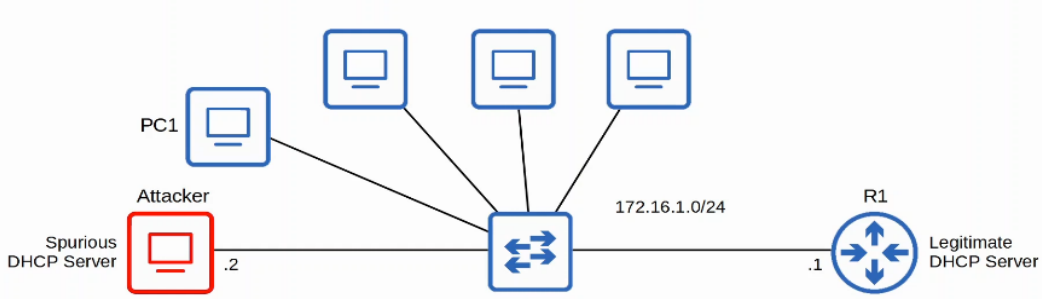
Un autre type d'attaque possible avec DHCP est appelé le DHCP Poisoning (Man in the Middle)

De même type que le ARP poisoning, le DHCP poisoning peut être utilisé pour faire fonctionner une attaque Man in the Middle. Un serveur DHCP parasite répond aux messages clients DHCP Discover et leurs assigne une adresse IP mais fais que les clients utilisent le serveur DHCP parasite comme passerelle par défaut.

Les clients acceptent le premier message client qu'ils reçoivent.

Cela cause que le client envoie le trafique à l'attaquant au lieu de la passerelle par défaut.

L'attaquant peut ensuite examiner/modifier le trafique avant de le partager à la passerelle par défaut.



Dans le réseau précédent par exemple le PC1 envoie un message Discover au Switch qui envoie en Broadcast à tous les appareils du réseau local en incluant l'attaquant. Le serveur DHCP légitime envoie une DHCP OFFER et le serveur DHCP parasite envoie lui aussi un DHCP OFFER.

C'est le message DHCP OFFER du serveur parasite qui est réceptionné en premier par le PC1.

Le PC1 va donc envoyer une réponse DECLINE au routeur R1, le DHCP légitime.

Donc si le PC1 veut envoyer un message il passera par le serveur parasite qui est le PC de l'attaquant qui lui même repartagera le message au serveur DHCP légitime qui est le routeur R1.

Lorsque DHCP snooping filtre les messages il les différencie entre les messages du serveur DHCP et le message du client DHCP.

Les messages envoyés par le serveur DHCP sont :

- OFFER

- ACK

- NAK est l'opposé de ACK est est utilisé pour décliner les messages REQUEST des clients

Les messages DHCP envoyés par les clients sont :

- DISCOVER

- REQUEST

- RELEASE est utilisé pour dire au serveur que le client n'a plus besoin d'adresse IP

- DECLINE est utilisé pour décliner une offre d'adresse IP d'un serveur DHCP

Si un message est reçu sur un port qui est de confiance il sera repartagé sans inspection.

Si un message DHCP est reçu sur un port qui n'est pas de confiance il sera inspecté comme suit :

- Si c'est un serveur DHCP il sera bloqué

- Si c'est un message client DHCP il aura le fonctionnement suivant :

Les requêtes de messages DISCOVER/REQUEST vérifie si la trame de l'adresse MAC source et les messages DHCP CHADDR correspondent si c'est le cas le message est partagé si ça ne l'est pas le message n'est pas partagé.

Les messages RELEASE/DECLINE est vérifié si l'adresse IP du paquet source et de l'interface de réception correspondent avec l'entrée de la table DHCP Snooping. Si elles correspondent le paquet est repartagé si elles ne correspondent pas le paquet est bloqué.

Lorsque le client prend une adresse IP d'un serveur cela crée une nouvelle entrée dans la table DHCP snooping.

Voyons à présent comment faire la configuration basique de DHCP snooping, pour cela on utilise les commandes suivantes :

On commence par configurer le SW2 avec les commandes :

```
SW2(config)#ip dhcp snooping
SW2(config)#ip dhcp snooping vlan 1
SW2(config)#no ip dhcp snooping information option
SW2(config)#interface g0/0
SW2(config-if)#ip dhcp snooping trust
```

On configure ensuite le SW1 avec les commandes suivante :

```
SW1(config)#ip dhcp snooping
SW1(config)#ip dhcp snooping vlan 1
SW1(config)#no ip dhcp snooping information option
SW1(config)#interface g0/0
SW1(config-if)#ip dhcp snooping trust

SW1#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
0C:29:2F:18:79:00   192.168.100.10   86294       dhcp-snooping   1     GigabitEthernet0/3
0C:29:2F:90:91:00   192.168.100.11   86302       dhcp-snooping   1     GigabitEthernet0/1
0C:29:2F:67:E9:00   192.168.100.12   86314       dhcp-snooping   1     GigabitEthernet0/2
Total number of bindings: 3
```

Une autre fonctionnalité de DHCP Snooping est le Rate-Limiting

Le DHCP Snooping peut limiter le taux auquel les messages DHCP sont permis pour entrer une interface. Si le taux de messages DHCP croise la limite configuré l'interface est err-disabled.

Tout comme port security l'interface est manuellement réactivé ou automatiquement réactivé avec errdisable recovery.

Voici comment configurer errdisable recovery :

```
SW1(config)#interface range g0/1 - 3
SW1(config-if-range)#ip dhcp snooping limit rate 1

*Jun  5 13:15:14.180: %DHCP_SNOOPING-4-DHCP_SNOOPING_ERRDISABLE_WARNING: DHCP Snooping received 1 DHCP packets on
interface Gi0/1
*Jun  5 13:15:14.181: %DHCP_SNOOPING-4-DHCP_SNOOPING_RATE_LIMIT_EXCEEDED: The interface Gi0/1 is receiving more
than the threshold set
*Jun  5 13:15:14.182: %PM-4-ERR_DISABLE: dhcp-rate-limit error detected on Gi0/1, putting Gi0/1 in err-disable
state
*Jun  5 13:15:15.185: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Jun  5 13:15:16.190: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to down
```

Voici comment réactiver err-disable recovery :

```
SW1(config)#errdisable recovery cause dhcp-rate-limit
```

On vérifie ensuite la configuration errdisable avec la commande :

```
SW1#show errdisable recovery
ErrDisable Reason            Timer Status
-----------------            --------------
arp-inspection               Disabled
bpduguard                    Disabled
channel-misconfig (STP)      Disabled
dhcp-rate-limit              Enabled
dtp-flap                     Disabled
gbic-invalid                 Disabled
inline-power                 Disabled
![output omitted due to length]



Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

Interface       Errdisable reason       Time left(sec)
---------       -----------------       --------------
Gi0/1            dhcp-rate-limit          293
```

La limite de taux peut être très utile pour protéger contre des attaques DHCP exhaustion.

L'option 82 aussi connu comme DHCP agent de relaie option d'information ou en Anglais « DHCP relay agent information option » est l'une des nombreuses options DHCP.

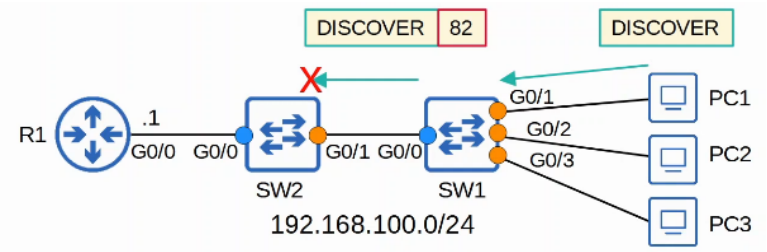
L'option 82 fournit des informations additionnels à propos des agent de relais DHCP qui reçoivent les messages clients sur quelle interface, quelle VLAN, etc...

L'agent de relais DHCP peut ajouter l'option 82 aux messages qu'ils repartagent vers le serveur DHCP.

Avec le DHCP snooping activé, par défaut les switchs Cisco ajoutent l'option 82 aux messages DHCP qu'ils reçoivent du client, même si le switch ne fonctionne pas comme agent relais DHCP.

Par défaut les switchs Cisco bloquent les messages DHCP avec l'option 82 qui sont reçu sur un port qui n'est pas de confiance.

Par exemple si un message contenant l'option 82 est envoyé par l'agent relais celui ci sera bloqué par le Switch si celui ci n'est pas de confiance :



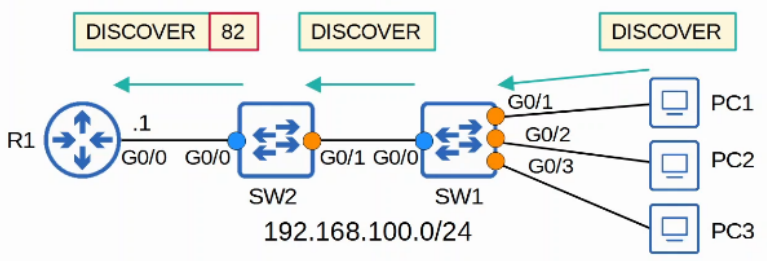
On peut voir ici que le message est bloqué par le Switch

```
SW2#
*Jun  6 01:36:15.298: %DHCP_SNOOPING-5-DHCP_SNOOPING_NONZERO_GIADDR: DHCP_SNOOPING drop message with non-
zero giaddr or option82 value on untrusted port, message type: DHCPDISCOVER, MAC sa: 0c29.2f67.e900
```

C'est pour cela qu'il peut être utilisé la commande suivante :

```
SW1(config)#no ip dhcp snooping information option
```

Ici la commande est lancé sur le Switch 1 ce qui cause que l'option 82 est supprimé, mais qu'ensuite l'option 82 est ajouté par le Switch 2 :



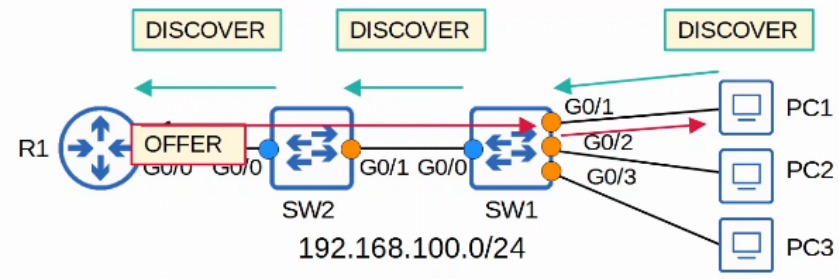
Le Routeur R1 va lui bloquer le message car l'option 82 y est présente dans l'entête.

```
R1#
*Jun  6 01:46:46.763: DHCPD: inconsistent relay information.
*Jun  6 01:46:46.763: DHCPD: relay information option exists, but giaddr is zero
```

On ajoute donc la commande sur le Switch 2 :

```
SW2(config)#no ip dhcp snooping information option
```

Cette fois le Switch va repartager les messages DHCP Discover et répond normalement :

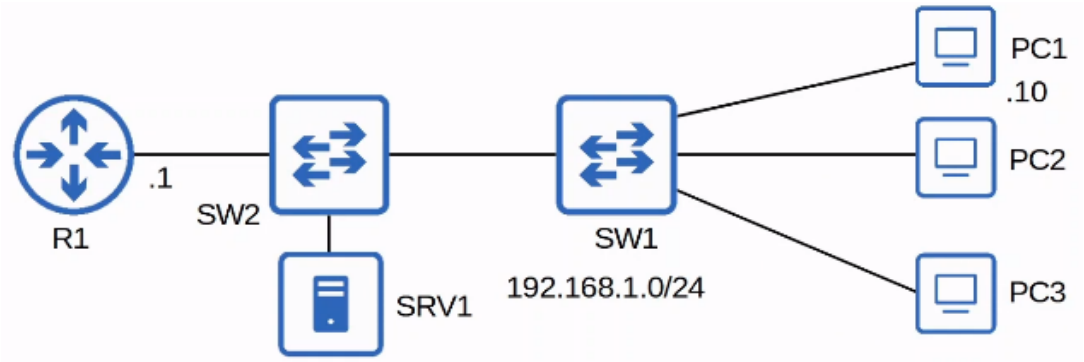


Cours 51 : ARP Inspection

Dans ce cours nous verrons ce qu'est le ARP Inspection, il s'agit d'une fonctionnalité de sécurité des Switch, qui inspecte les messages ARP de la même manière que DHCP Snooping inspecte les messages DHCP.

Nous verrons tout d'abord ce qu'est Dynamic ARP Inspection et comment il fonctionne, nous verrons quelles attaques il permet de bloquer et comment le configurer sur un Switch.

Commençons par revoir le fonctionnement du protocole ARP sur le réseau suivant :



ARP est utilisé pour apprendre l'adresse MAC d'un autre appareil avec une adresse IP connu.

Par exemple un PC utilise ARP pour apprendre l'adresse MAC de sa passerelle par défaut pour communiquer avec le réseau externe.

Cela consiste en l'échange de deux messages : la requête et la réponse.

Par exemple PC1 envoie une requête DNS avec pour adresse IP source 192.168.1.10 et adresse IP de destination : 8.8.8.8 mais l'adresse MAC de destination lui est inconnu.

Le PC va envoyer la trame à sa passerelle par défaut car l'adresse 8.8.8.8 est en dehors de son réseau local. Il va donc envoyer en Broadcast la requête ARP suivante avec les informations suivantes :

Adresse IP source : 192.168.1.10, Adresse IP destination : 192.168.1.1, Adresse MAC source : son adresse MAC, Adresse MAC de destination : F.F.F

Tous les appareils vont recevoir ce message puisque l'adresse de destination est l'adresse de Broadcast F.F.F

On peut voir le message suivant sur Wireshark :

```
> Frame 99: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
v Ethernet II, Src: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
    Sender IP address: 192.168.1.10
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.1
```

Puisque le routeur R1 est bien l'adresse de Destination de la requête du PC1, R1 va envoyer une réponse ARP afin d'informer le PC1 qu'il s'agit bien de son adresse MAC et qu'il puisse l'enregistrer dans sa table ARP. Le routeur R1 ajoute lui aussi une entrée dans sa table ARP pour le PC1 lorsqu'il reçoit la requête ARP.

Voici la réponse ARP envoyé par le routeur R1 :

```
> Frame 224: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
v Ethernet II, Src: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00), Dst: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
  > Destination: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
  > Source: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00)
    Sender IP address: 192.168.1.1
    Target MAC address: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
    Target IP address: 192.168.1.10
```

Le PC1 peut à présent ajouter l'adresse MAC du routeur R1 lorsqu'il envoie des requêtes DNS, le routeur R1 recevra la requête puis la retransmettra au réseau externe.

Il existe un autre type de message ARP qui sont appelés les messages « Gratuitous ARP » qui sont des réponses ARP envoyés sans réception de requête ARP.

Ces messages sont envoyés à l'adresse MAC de Broadcast (F.F.F), ces messages permettent aux appareils du réseau d'apprendre les appareils enregistrés sans avoir à envoyer de requêtes ARP.

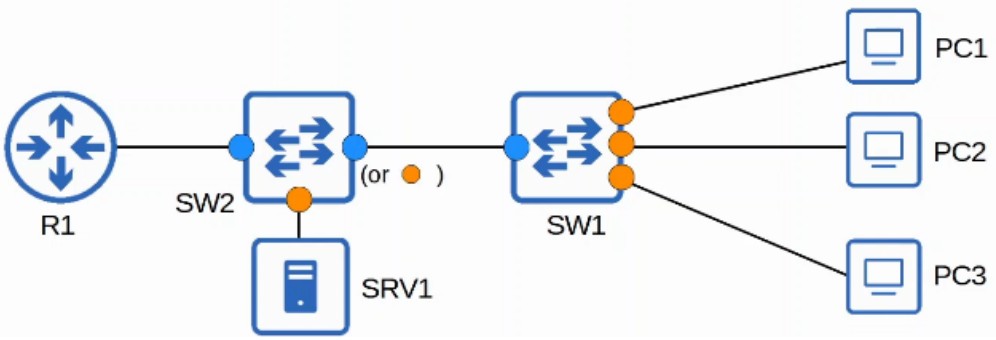
Certains appareils envoient automatiquement des messages « GARP » lorsque l'interface est activé, que l'adresse IP change, que l'adresse MAC change, etc...

Dans le réseau précédent par exemple la réponse ARP est envoyé vers tous les appareils du réseau local, les PC vont enregistrer le PC2 dans leur table ARP ansi que les Switchs.

DAI est une fonctionnalité de sécurité des switchs qui est utilisé pour filtrer les messages ARP reçus sur des ports qui ne sont pas de confiance. DAI filtre uniquement les messages ARP. Les messages non ARP ne sont pas affectés. Tous les ports ne sont pas de confiance par défaut.

Tous les ports connectés à d'autres appareils du réseau (Switch, routeur) doivent être configurés comme de confiance, les interfaces connectés à des hôtes finaux ne sont pas confiance.

Sur le réseau précédent par exemple les interfaces de confiance sont en bleus celle qui ne le sont pas sont en orange :



Le fonctionnement de Dynamic ARP Inspection est similaire à DHCP, par exemple sur le réseau, le PC1 envoie une requête ARP au SW1, puisque le message arrive sur un port qui n'est pas de confiance le Switch va utiliser DAI pour vérifier que le message est normal, si c'est le cas il va le distribuer au Switch 2, le SW2 ne va pas le vérifier car il reçoit la requête sur une interface de confiance. Le SW2 redistribue le paquet au routeur R1 qui envoie ensuite une réponse ARP.

Si à présent le PC2 envoie le message et que le SW2 décide de bloquer le message car il contredit les règle de DAI (nous verrons comment est ce possible de contredire les règle de DAI), le message ne sera pas redistribué.

Voyons le fonctionnement de ARP poisoning (Man in the Middle)

De manière similaire à DHCP Poisoning, ARP Poisoning implique que l'attaquant manipule la table de la cible ARP donc le trafique est envoyé à l'attaquant.

Pour faire cela l'attaquant envoie des messages ARP Gratuitous en utilisant l'adresse IP d'un autre appareil. Les autres appareils du réseau reçoivent le GARP et mettent à jour leurs table ARP ce qui cause que le trafique

est envoyé à l'attaquant au lieu de l'envoyer à la destination légitime.

A présent lorsque par exemple le PC1 veut envoyer un paquet au réseau externe, il envoie le message au PC2 en premier qui l'enregistre et qui peut le sauvegarder ou même le modifier puis le redistribuer à la destination légitime, dans le réseau précédent le routeur R1.

Voyons comment DAI peut protéger de ce type d'attaque. DAI inspecte l'adresse MAC de l'envoyeur ainsi que son adresse IP de la partie des messages ARP reçu sur des ports qui ne sont pas de confiance et vérifie s'il y a une entrée qui correspond dans la table DHCP snooping binding.

```
SW1#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
0C:29:2F:18:79:00   192.168.100.10   86294       dhcp-snooping   1     GigabitEthernet0/3
0C:29:2F:90:91:00   192.168.100.11   86302       dhcp-snooping   1     GigabitEthernet0/1
0C:29:2F:67:E9:00   192.168.100.12   86314       dhcp-snooping   1     GigabitEthernet0/2
Total number of bindings: 3
```

S'il y a une entrée qui correspond le message ARP est redistribué normalement.

S'il n'y a pas d'entrée qui correspond le message ARP est bloqué.

DAI n'inspecte pas les messages reçus sur des ports de confiance et sont redistribués normalement.

Les ACLs ARP peuvent être configurés manulement pour cartographier des adresses MAC/IP pour vérifier le DAI. Cela peut être utile pour les hôtes qui n'utilisent pas DHCP.

DAI peut être utilisé pour faire des vérifications de paquet plus approfondie mais cela est optionnel.

Tout comme DHCP Snooping, DAI supporte aussi la limité des taux (rate limiting) pour empêcher l'attaquant de surcharger le switch de messages ARP.

DHCP Snooping et DAI requière des ressources du CPU depuis le Switch, donc si même si les messages de l'attaquant sont bloqués cela peut surcharger le CPU avec des messages ARP, si l'attaquant tente de faire cela, la limite des taux va désactiver l'interface.

Voici les commandes à utiliser pour configurer DAI, on commence par configurer le SW2 du réseau précédent :

```
SW2(config)#ip arp inspection vlan 1
SW2(config)#interface range g0/0 - 1
SW2(config-if-range)#ip arp inspection trust
```

SW2(config)#ip arp inspection vlan 1

Permet d'activer DAI sur un vlan (ici le vlan 1) il faut ajouter tous les vlan du réseau sinon seulement les vlan spécifiés seront inspectés

`SW2(config)#interface g0/0 -- 1`

`SW2(config-if-range)#ip arp inspection trust`

On lance ensuite ces commandes afin d'activer les interface G0/0 et G0/1 comme interfaces de confiance.

On lance les même commandes sur le SW1 sauf que l'on fais cette fois confiance seulement à l'interface G0/0

```
SW1(config)#ip arp inspection vlan 1
SW1(config)#interface g0/0
SW1(config-if)#ip arp inspection trust
```

On peut noter une différence entre la configuration DHCP Snooping et la configuration de DAI :

DHCP snooping requière deux commande pour être activé :

`ip dhcp snooping`

`ip dhcp snooping vlan` *`numéro de vlan`*

DAI requière uniquement une seule commande :

`ip arp inspection vlan` *`numéro de vlan`*

Il est possible d'afficher les interfaces avec l'inspection arp avec la commande :

`show ip arp inspection interfaces`

```
SW1#show ip arp inspection interfaces

 Interface        Trust State     Rate (pps)    Burst Interval
 --------------   -----------     ----------    --------------
 Gi0/0            Trusted               None               N/A
 Gi0/1            Untrusted               15                 1
 Gi0/2            Untrusted               15                 1
 Gi0/3            Untrusted               15                 1
 Gi1/0            Untrusted               15                 1
 Gi1/1            Untrusted               15                 1
 Gi1/2            Untrusted               15                 1
 Gi1/3            Untrusted               15                 1
 Gi2/0            Untrusted               15                 1
 Gi2/1            Untrusted               15                 1
 Gi2/2            Untrusted               15                 1
 Gi2/3            Untrusted               15                 1
 Gi3/0            Untrusted               15                 1
 Gi3/1            Untrusted               15                 1
 Gi3/2            Untrusted               15                 1
 Gi3/3            Untrusted               15                 1
```

On peut voir différentes informations comme l'interface, si l'interface est de confiance ou non, le taux de limiting (rate). Il y a une différence entre DAI rate limiting et DHCP snooping rate limiting, DAI rate limiting est activé sur les ports qui ne sont pas de confiance par défaut avec un taux de 15 paquets par secondes. Cela est désactivé sur des ports de confiance par défaut.

Le DHCP snooping est configuré en terme de X paquet par seconde, tandis que DAI interval par rafal (Burst interval) permet de configurer la limite des taux en terme X paquets par Y secondes.

Pour configurer DAI rate limiting on lance les commandes suivantes :

```
SW1(config)#interface range g0/1 - 2
SW1(config-if-range)#ip arp inspection limit rate 25 burst interval 2
```

`ip arp inspection limit rate 25 burst interval 2`

Ici les interface G0/1 et G0/2 ont été configurés pour avoir un taux de limite de 25 paquet, le burst interval a été changé pour passer de 1 à 2 par secondes. La configuration du Burst Interval est optionnel si elle n'est pas configuré, par défaut ce sera 1 par seconde.

On configure ensuite l'interface G0/3 avec les commandes :

```
SW1(config-if-range)#interface range g0/3
SW1(config-if)#ip arp inspection limit rate 10
```

Ici le Burst interval n'est pas configuré.

On peut ensuite afficher la configuration :

```
SW1(config-if)#do show ip arp inspection interfaces

 Interface        Trust State     Rate (pps)    Burst Interval
 --------------   -----------     ----------    --------------
 Gi0/0            Trusted               None               N/A
 Gi0/1            Untrusted               25                 2
 Gi0/2            Untrusted               25                 2
 Gi0/3            Untrusted               10                 1
![output omitted]
```

Si les messages ARP sont reçus plus vite que le taux spécifié, l'interface passera en err-disabled. Pour la réactiver il faudra soit :

- éteindre (*shutdown*) et ralumer l'interface (*no shutdown*)

- utiliser *errdisable recovery cause arp-inspection*

Par défaut DAI vérifie l'adresse MAC de l'envoyeur et son adresse IP, il est cependant possible de modifier cela avec la commande :

```
SW1(config)#ip arp inspection validate src-mac
```

```
SW1(config)#ip arp inspection validate ?
  dst-mac  Validate destination MAC address
  ip       Validate IP addresses
  src-mac  Validate source MAC address
```

Voici la définition donnée par Cisco pour les options de la commande :

dst-mac :Compare l'adresse MAC de destination dans l'en-tête Ethernet avec l'adresse MAC cible dans le corps ARP. Cette vérification est effectuée pour les réponses ARP. Lorsqu'ils sont activés, les paquets avec différentes adresses MAC sont classés comme non valides et sont supprimés.

IP : Compare le corps ARP pour les adresses IP non valides et inattendues. Les adresses comprennent 0.0.0.0, 255.255.255.255, et toutes les adresses de multidiffusion IP. Les adresses IP de l'expéditeur sont comparées dans toutes les demandes et réponses ARP. Les adresses IP cibles sont comparées uniquement dans les réponses ARP.

Src-mac : Compare l'adresse MAC source dans l'en-tête Ethernet avec l'adresse MAC de l'expéditeur dans le corps ARP. Cette vérification est effectuée sur les demandes et les réponses

ARP. Lorsqu'ils sont activés, les paquets avec différentes adresses MAC sont classés comme non valides et sont supprimés.

Voici un message Wireshark qui contient une réponse ARP :

```
> Frame 224: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
v Ethernet II, Src: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00), Dst: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
  > Destination: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
  > Source: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00)
    Sender IP address: 192.168.1.1
    Target MAC address: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
    Target IP address: 192.168.1.10
```

Ici on peut voir que l'adresse de destination correspond à celle contenu dans la réponse ARP donc le message est accepté.

Ces vérifiations sont faites en plus de la vérification standard DAI.

Si configuré les messages ARP devront tous passer les vérifications pour être validés.

DAI peut aussi être configuré pour faire d'autres vérifications :

Ici on lance les trois commandes suivante :

```
SW1(config)#ip arp inspection validate dst-mac
SW1(config)#ip arp inspection validate ip
SW1(config)#ip arp inspection validate src-mac
```

On vérifie ensuite laquelle est retenue dans la configuration, on peut voir que seulement la dernière commande est correctement enregistré.

```
SW1(config)#do show running-config | include validate
ip arp inspection validate src-mac
```

Par contre si l'on lance la configuration pour inspecter les 3 en une commande, cette fois les trois paramètres sont enregistrés :

```
SW1(config)#ip arp inspection validate ip src-mac dst-mac

SW1(config)#do show running-config | include validate
ip arp inspection validate src-mac dst-mac ip
```

Il faut donc lancer la vérification de validation en une seule commande.

```
SW2#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
0C:29:2F:18:79:00   192.168.1.12     79226       dhcp-snooping   1     GigabitEthernet0/1
0C:29:2F:90:91:00   192.168.1.10     79188       dhcp-snooping   1     GigabitEthernet0/1
0C:29:2F:67:E9:00   192.168.1.11     79210       dhcp-snooping   1     GigabitEthernet0/1
Total number of bindings: 3
```

Voyons rapidement le fonctionnement de ARP ACLs :

On affiche d'abord la table DHCP snooping binding du SW2

Le SRV1 a une adresse IP statique, lorsqu'il essaie d'envoyer une requête ARP, le message sera bloqué avec le message d'erreur suivant :

```
!SRV1 has a static IP address of 192.168.1.100, so it does not have an entry in SW2's DHCP
!snooping binding table.

*Jun 19 05:56:15.538: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Gi0/2, vlan 1.
([0c29.2f1e.7700/192.168.1.100/0000.0000.0000/192.168.1.1/05:56:14 UTC Sat Jun 19 2021])
```

Cela est dû au fais que le SRV1 n'est pas enregistré dans la table DHCP snooping Binding.

Pour configurer l'interface on lance les commandes suivante :

```
SW2(config)#arp access-list ARP-ACL-1
SW2(config-arp-nacl)#permit ip host 192.168.1.100 mac host 0c29.2f1e.7700

SW2(config)#ip arp inspection filter ARP-ACL-1 vlan 1
```

```
SW2(config)#arp access-list ARP-ACL-1
```

```
SW2(config-arp-nacl)#permit ip host 192.168.1.100 mac host 0c29.2f1e.7700
```

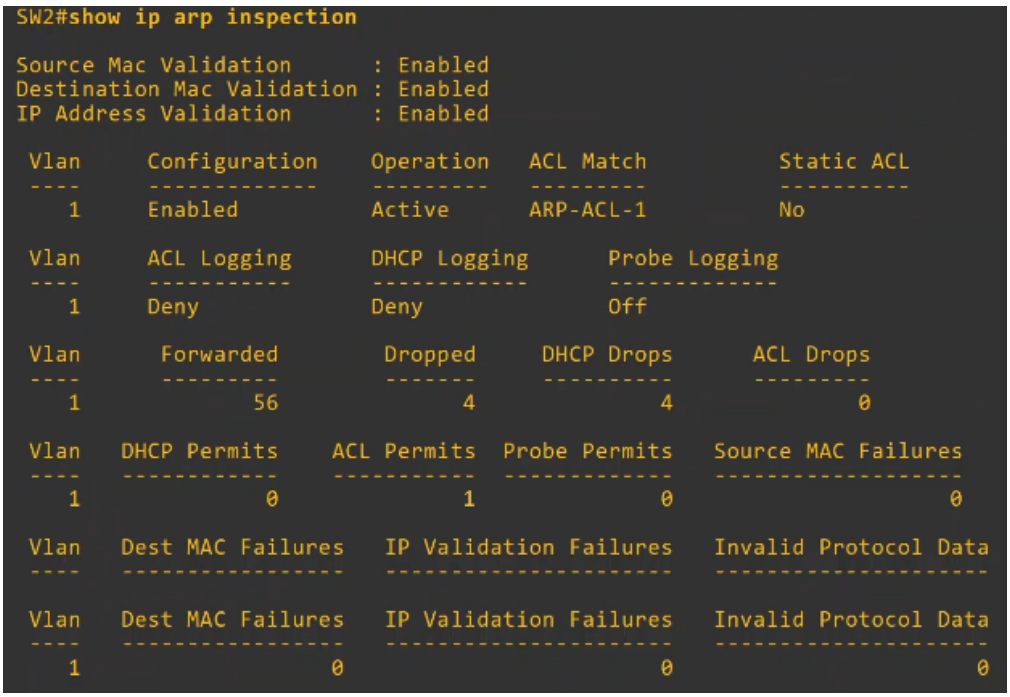
```
SW2(config)#ip arp inspection filter ARP-ACL-1 vlan 1
```

Ici une ACL appelé ARP-ACL-1 été crée avec la première commande, puis l'hôte avec l'adresse IP et l'adresse MAC spécifié ont été autorisés. On applique ensuite la configuration en lançant la commande arp inspection.

Cela permet à ce que le SRV1 puisse à présent envoyer une requête ARP et que le SW1 le retransmette bien qu'il n'y ait pas d'entrée dans la table DHCP snooping binding.

Une commande utile pour afficher la configuration arp inspection est :

```
SW2#show ip arp inspection
```



On peut voir que les validation source mac, dest mac et ip son activés, on voit aussi que DAI est activé sur la Vlan 1 et que l'ARP-ACL est fonctionnel. Le statique AC est configuré comme « no » si le static ACL est en « yes » le déni implicite à la fin de l'ARP-ACL prendra effet, cela cause que les messages ARP non permis par l'ARP ACL d'être bloqués. Cela signifie que seulement l'ARP-ACL peut être vérifié, la table DHCP snooping ne sera pas vérifié.

Cours 52 : Architecture LAN

Dans ce cours nous verrons les architectures LAN, avec tout d'abord les 2-Tier et 3-Tier des Architecture LAN qui sont des conseptions utilisé dans les réseaux d'entreprise.

Nous verrons ensuite Spine-Leaf Architecture qui est une conception commune utilisé dans des environnements de centre de données, puis les SOHO (Small Office/Home Office).

Nous avons étudié dans les cours précédents des réseaux de technologies variés avec le routage, les switch, STP, Etherchannel, OSPF, etc...

Nous allons voir quelques conceptions basique d'architectures Réseaux.

Il y a des standard de bonne pratique dans la conception d'un réseau.

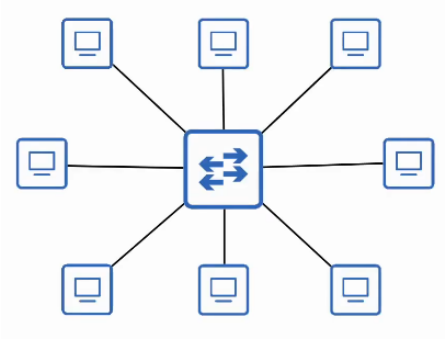
Il y a tout de même peu de réponses qui sont toujours correct dans tous les cas.

La réponse à la plupart des questions général à propos de la conception d'un réseau est que cela dépend puisque les prérequis de chaque réseau est différent.

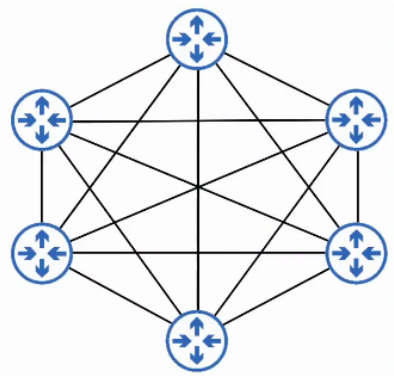
Lorsque l'on débute dans une carrière en réseau, il ne nous sera probablement pas demandé de faire une conception d'un réseau, il est tout de même important de comprendre les réseaux que l'on va configurer et réparer c'est pour cela qu'il faut connaître les bases dans la conception d'un réseau.

Nous allons tout d'abord faire une introduction sur les différentes topologies communes qui sont possibles.

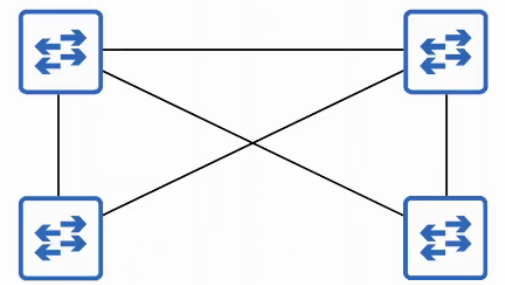
- Étoile : Lorsque plusieurs appareils sont connectés à un appareil central il est possible de les dessiner en « étoile », c'est pour cela que cette topologie est appelé topologie en étoile.



- Réseau totalement maillé : Lorsque chaque appareil est connecté à chacun des autre appareil du réseau.



- Réseau partiellement maillé : Lorsque les appareils sont connectés entre eux mais pas tous.



Les conception de LAN Two-Tier Campus sont des architectures réalisés dans un bâtiment ou plusieurs bâtiments. Les conception de LAN deux tiers consistent en deux couches hiérarchiques :

- Couche d'accès (Access Layer)

- Couche Distribution (Distribution Layer)

Aussi appelé le « collapdsed core » ou coeur effondré en Français car il ommet la couche qui se trouve dans les conception Trois Tier : La couche Coeur.

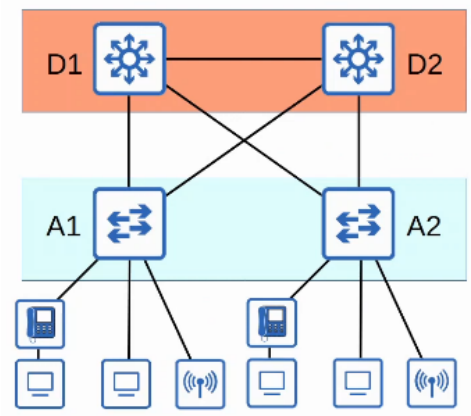
La couche accès est la couche à laquelle les hôtes se connectent (Les PC, les Imprimantes, etc...)

Les Switch qui sont en couche Accès ont donc beaucoup de ports auquel des hôtes sont connectés.

Le QoS est réalisé dans cette couche d'accès, mais aussi les services de sécurité comme port security, DAI, etc.. les ports Switch doivent avoir le PoE activé pour les point d'accès sans fil, les téléphones IP, etc...

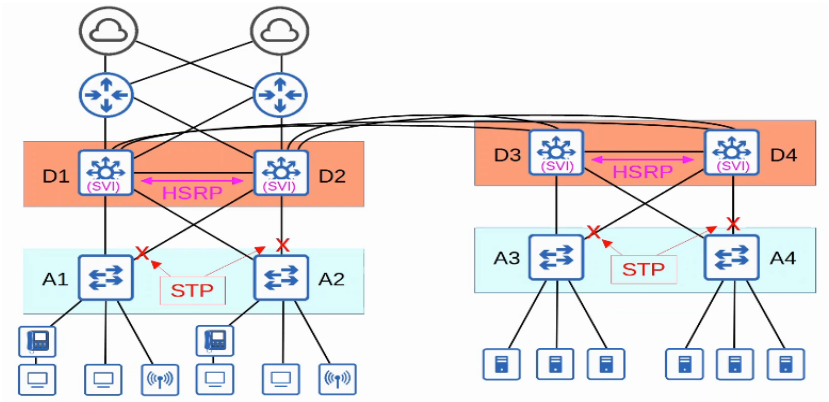
La couche Distribution agrège les connexion faites par la couche d'accès avec les Switch. Cette couche est la limite entre la couche 2 et 3. Cette couche permet la connexion aux services comme Internet, WAN, etc...

Voyons un exemple :

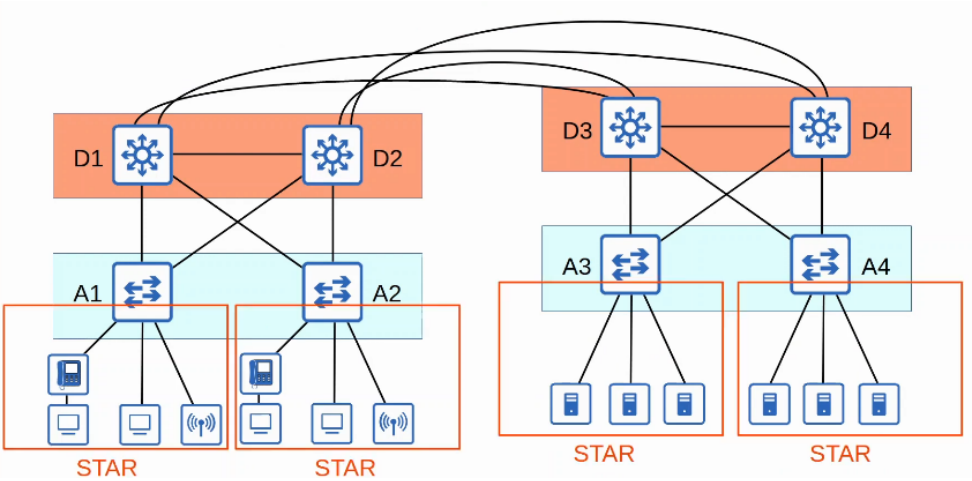


Ici la couche Distribution est en orange, la couche Accès est en bleu, le protocole STP va ici être activé au niveau de la couche d'accès et désactiver les interfaces correspondante à ce niveau.

Le protocole HSRP qui est un protocole de couche 3 sera quant à lui activé au niveau de la couche Distribution. Le réseau peut être étendu de la manière suivante :

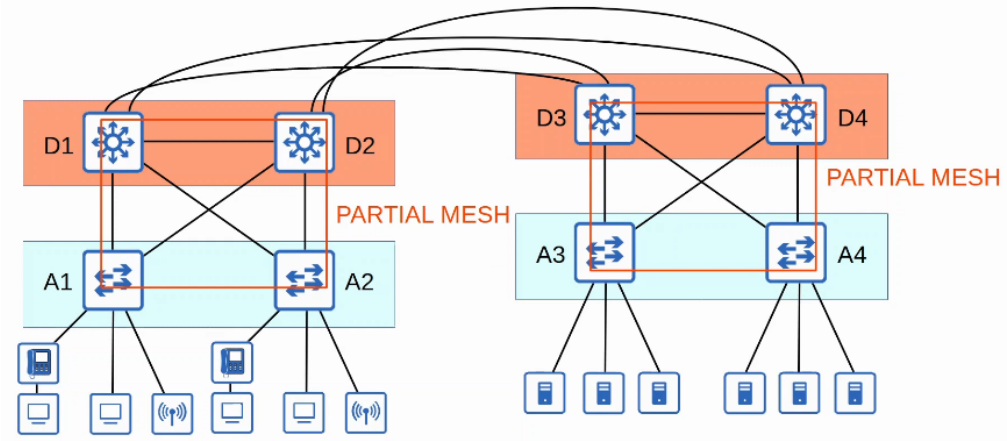


Dans une conception collapsed core, la couche distribution est souvent appelé la couche Core distribution, car elle a pour rôle de fonctionner sur les deux couches.

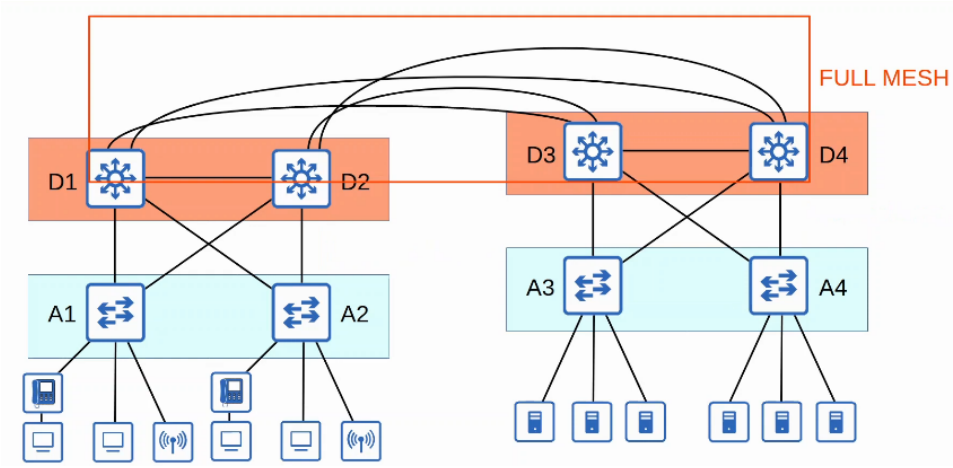


Dans le réseau suivant on peut voir qu'est présent 4 réseau avec une topologie en étoile :

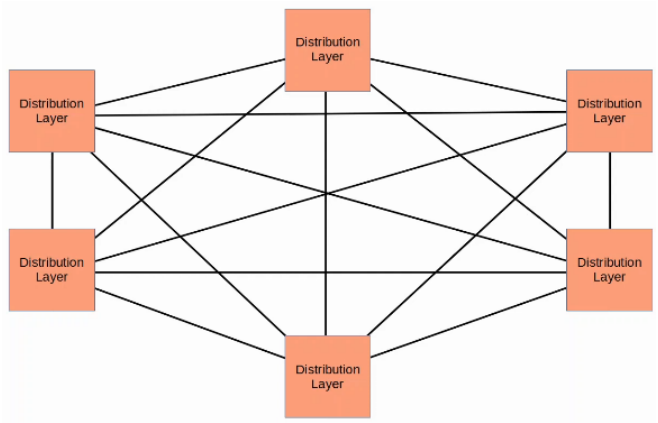
Est aussi présent 2 réseaux avec une topologie partiellement maillé, puisque quelques appareils mais pas tous sont connectés entre eux :



Est présent 2 réseaux avec une topologie totalement maillé, puisque tous les appareils (Switchs) sont connectés entre eux :



Si le réseau est large il peut y avoir plusieurs couches réseaux connectés à différentes partie des LAN :

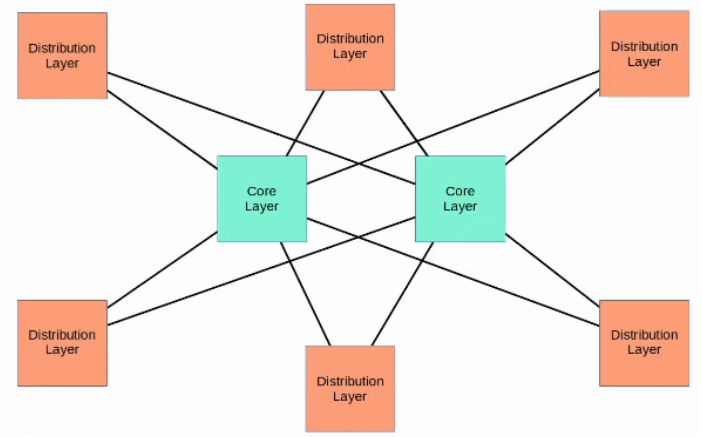


Le nombre de connexion requises entre les couches Distribution des Switch augmente rapidement.

Pour aider l'évolution de large réseaux LAN il est possible d'ajouter une couche coeur (Core Layer)

s'il y a plus de 3 couches distribution pour une seule localisation.

Voici à quoi devrait ressembler le réseau avec une couche coeur (Core Layer) :



Une conception d'un réseau LAN consiste en 3 couches hiérarchiques :

- Couche Accès

- Couche Distribution

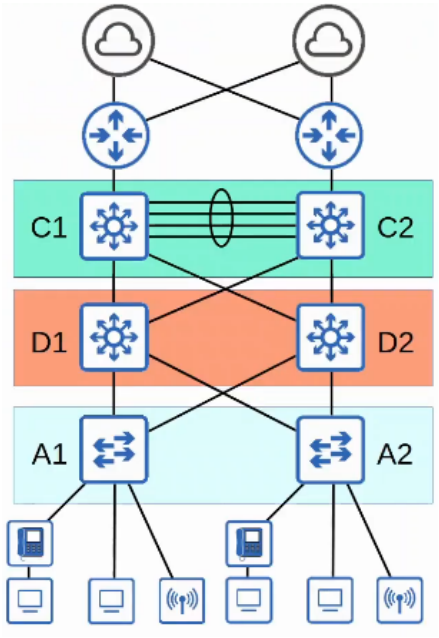
- Couche Coeur (Core Layer)

La couche coeur permet à connecter plusieurs couches distribution dans de large réseaux LAN. La concentration est donnée sur la rapidité (avec une rapidité de transport)

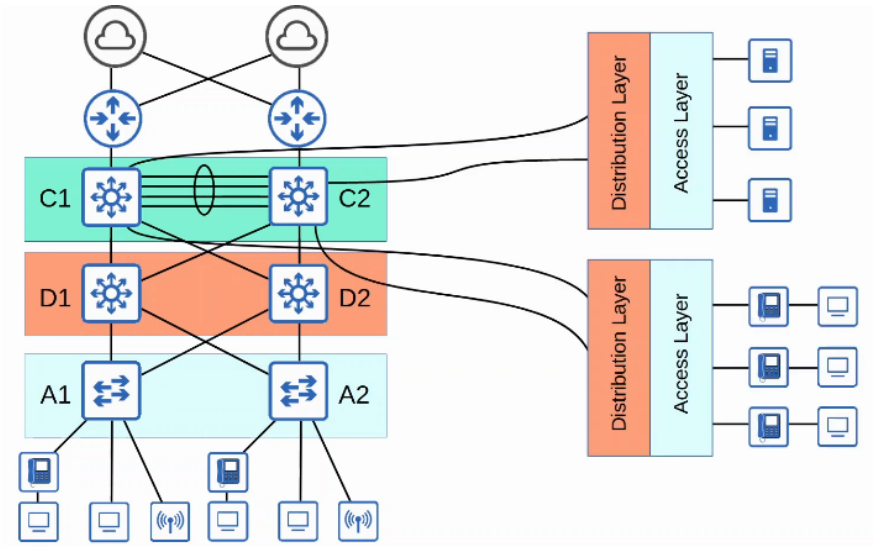
Les opération CPU intensive comme la sécurité, Marquage QoS/Classification, etc... doit être activé sur cette couche. Les connexion sont uniquement de couche 3. Cette couche doit pouvoir maintenir la connectivité à travers les LAN même si les appareils dysfonctionnent.

Voyons les réseaux précédemment vu mais avec une couche Coeur.

Sur le réseau suivant, les Switch multicouche sont configurés en Etherchannel et constituent la couche coeur :



On peut aussi ajouter des couches de Distribution connectés à cette couche coeur :

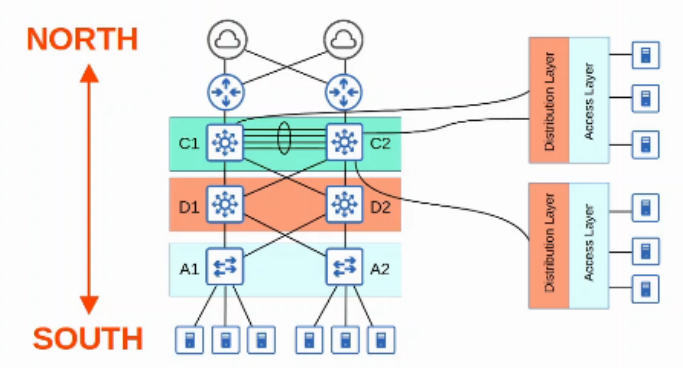


Les architecture Spine Leaf sont conçu pour fonctionner dans des centres de données.

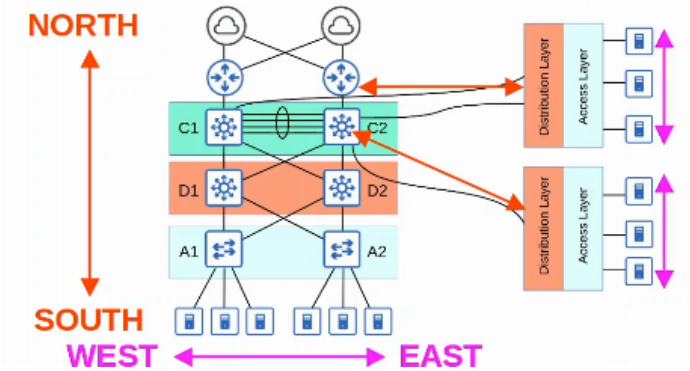
Les centres de données sont des espaces ou bâtiments conçus pour être utilisé afin de stocker des système d'ordinateur comme des serveurs et des appareils réseau.

Les centre de donnée traditionnel utilisent des architecture Trois-Tier (Avec les couches Accès, Distribution et Coeur). Cela fonctionne bien lorsque la plupart du trafique du réseau est Nord-Sud

L'information Nord Sud (North-South) fais référence aux couches qui sont placés de manière horizontal du nord et sud comme sur le réseau précédemment vu :



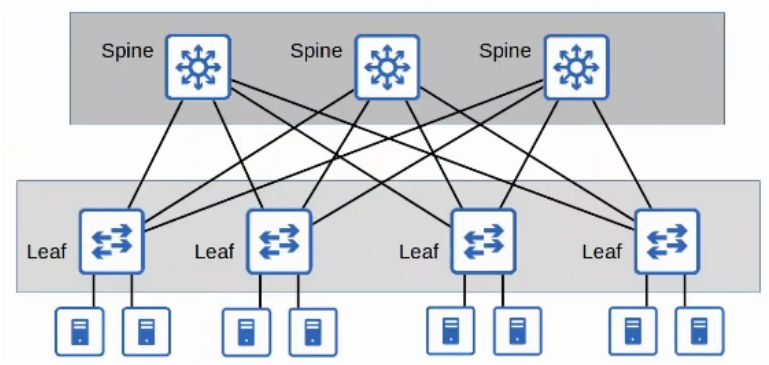
Contrairement aux trafique de type Est-Ouest (East-West) :



Avec la présence de serveurs virtuels, les applications sont souvent déployé de manière distribués (à travers plusieurs serveurs physique), ce qui augmente le montant de trafique Est-Ouest du data center. Les architectures Trois-Tier conduisent en un goulot d'étranglement de la bande passante en fonction de la variabilité de la latence server-to-server qui dépende du chemin que le trafique prend.

Pour résoudre cela, les architecture Spine-Lead (aussi appelé Architecture Clos) deviennent importante dans les data center.

Voici à quoi cela ressemble :



Il y a quelques règles à propos des architectures Spine Leaf :

- Chaque Switch Leaf est connecté à chacun des Switch Spine

- Chaque Switch Spine est connecté à chacun des Swicth Leaf

- Les Switch Leaf ne se connectent pas à d'autres Switch Leaf

- Les Switch Spine ne se connectent pas à d'autres Switch Spine

- Les hôtes finaux (serveurs, etc...) se connectent uniquement à des Switch Leaf

Le chemin pris par le trafique est aléatoirement choisi pour balancer le trafique sur les Switch Spine

Chaque serveur est séparé par le même nombre de « bond » (excepté ceux connectés au même Leaf) se qui fournit une latence consistent pour le trafique East West. (Sur le réseau 3 bond à chaque fois)

Les Small Office/Home Office (SOHO) se réfèrent aux office de petites compagnies, ou de petites maison avec peu d'appareils.

Il n'est pas obligatoire qu'il s'agisse d'une « office » de maison, si le réseau de la maison est connecté à Internet ce réseau est considéré comme un réseau SOHO.

Les réseaux SOHO n'ont pas de nécessité à être complexe, donc toutes les fonctions du réseau sont fournis par un seule appareil, souvent appelé un routeur maison ou routeur sans fil.

Ce seul appareil peut servir comme :

- Routeur

- Switch

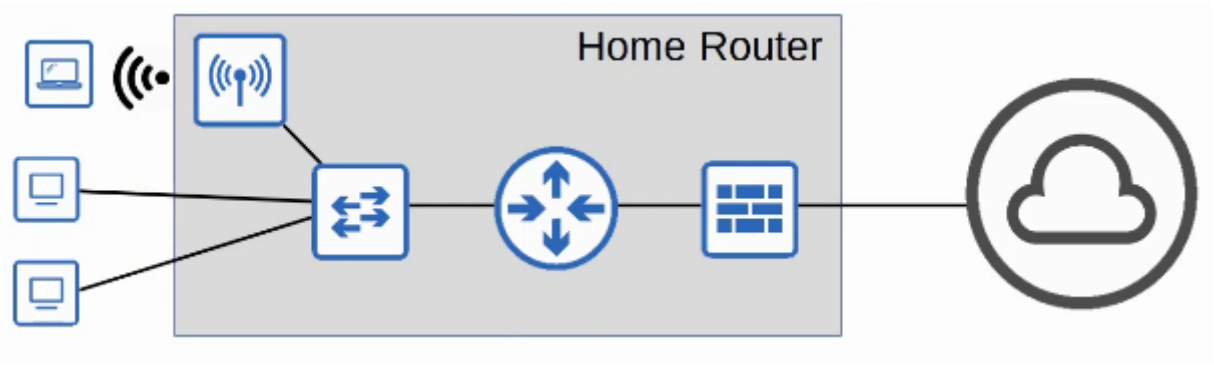
- Firewall

- Point d'accès sans fil

- Modem

Voici un exemple de ce type de routeur qui sont utilisés :

De petits réseaux pour de petites entreprises n'ont pas forcément les ressources pour obtenir de grosses infrastructures c'est pour cela qu'ils utilisent des routeurs de ce type comme sur la topologie suivante :



Cours 53 : Architectures WAN

Dans ce cours nous ferons une introduction de WAN, puis sur différents types de connexions utilisé par WAN : leased lines, ainsi qu'une autre technologie WAN appelé MPLS (Multi Protocole Label Switching) qui permet de fournir une sorte de VPN. Nous verrons également quelques options pour la connectivité internet, ainsi que les connexion Internet par VPN (Virtual Private Networks).

WAN est l'acronyme de Wide Area Network, un WAN est un réseau qui s'étend à travers une large zone géographique, par exemple entre des villes, des régions, etc...

Les WANs sont donc utilisés pour connecter géographiquement des LANs séparés.

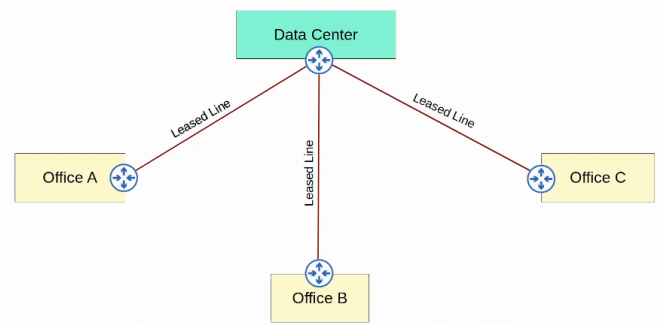
Internet lui même peut être considéré comme un WAN, le terme WAN est de manière général utilisé pour se référer à des connexions en entreprise privée qui connecte ses sites et centre de données entre eux.

Il y a également les réseaux comme Internet, VPN (Virtual Private Networks) qui peuvent être utilisés pour créer des connexions privée par WAN.

Il y a eu différentes technologies WAN durant plusieurs années. En fonction de la localisation, certaines seront disponible d'autres ne le seront pas.

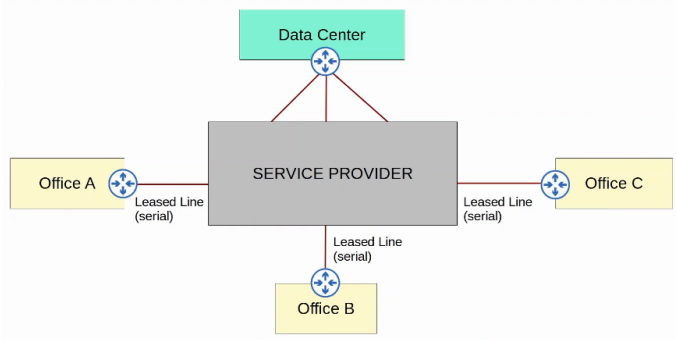
Les technologies qui sont considérés comme ancienne dans un pays peuvent continuer à être utilisé dans un autre pays.

Par exemple sur le réseau suivant :



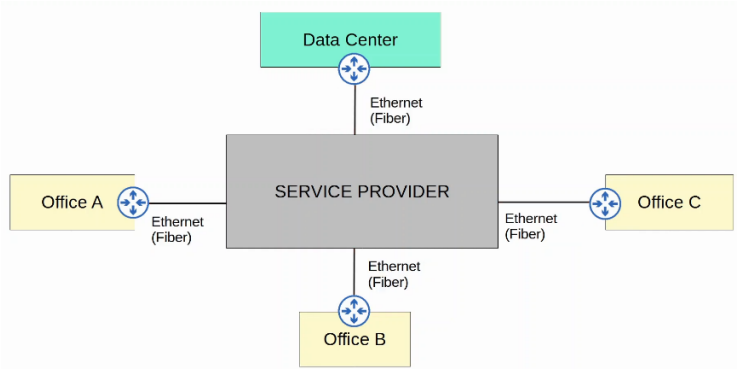
On peut voir que plusieurs sites sont connectés entre eux par l'intermédiaire du data center, la connexion est établi par des Leased Line qui sont des connexion spéciale pour connecter deux sites.

Ce n'est pas une connexion partagé mais une connexion privée afin de connecter plusieurs sites entre eux. Ce type de topologie est d'ailleurs appelé topologie en étoile pour un réseau LAN, mais en terme de WAN on utilise les termes : Hub et Spoke, le Data center est appelé le Hub tant dis que les office sont appelés les Spoke. C'est une topologie Hub and Spoke.

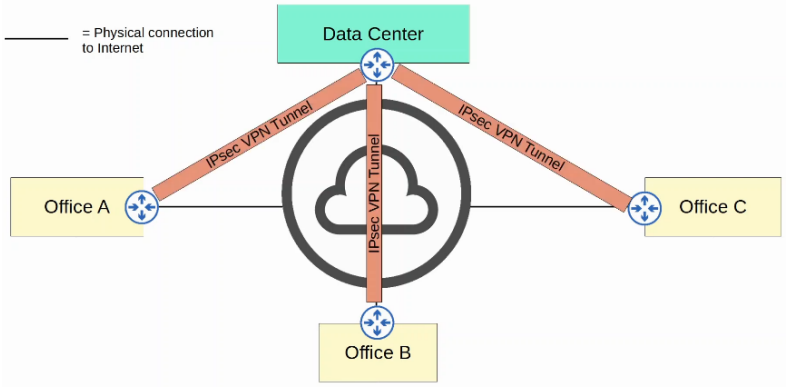


Une représentation plus précise de Least Line est le suivant, dans lequel le Data center ainsi que les offices sont connectés au fournisseur Internet par le moyen de câble Serial.

Il est aussi possible de remplacer les connexions Serial par des câbles fibre.



Internet peut aussi être utilisé pour des connexion WAN entre différents sites, mais Internet n'est pas un réseau privé mais un réseau publique partagé donc envoyer des données à travers internet de façon non protégé n'est pas une bonne idée, pour mettre en place une connexion entre ces sites de manière sécurisé l'entreprise peut mettre en place des VPN comme sur le réseau suivant :



Un Leased Line est un lien physique dédié qui permet de connecter deux sites. Les Leased Line utilisent des connexions Serial (PPP ou HDLC encapsulation). Il y a une grande variété de standard qui fournissent différentes vitesses et différents standard sont disponibles dans différents pays.

System	North American	Japanese	European (CEPT)
Level zero (channel data rate)	64 kbit/s (DS0)	64 kbit/s	64 kbit/s
First level	1.544 Mbit/s (DS1) (24 user channels) (T1)	1.544 Mbit/s (24 user channels)	2.048 Mbit/s (32 user channels) (E1)
(Intermediate level, T-carrier hierarchy only)	3.152 Mbit/s (DS1C) (48 Ch.)	–	–
Second level	6.312 Mbit/s (DS2) (96 Ch.) (T2)	6.312 Mbit/s (96 Ch.), or 7.786 Mbit/s (120 Ch.)	8.448 Mbit/s (128 Ch.) (E2)
Third level	44.736 Mbit/s (DS3) (672 Ch.) (T3)	32.064 Mbit/s (480 Ch.)	34.368 Mbit/s (512 Ch.) (E3)
Fourth level	274.176 Mbit/s (DS4) (4032 Ch.)	97.728 Mbit/s (1440 Ch.)	139.264 Mbit/s (2048 Ch.) (E4)
Fifth level	400.352 Mbit/s (DS5) (5760 Ch.)	565.148 Mbit/s (8192 Ch.)	565.148 Mbit/s (8192 Ch.) (E5)

Par exemple aux Etats-Unis le standard commence par T1 - T2 – T3.

En Europe le standard commence par E1 - E2 – E3.

A cause des coût chère, du temps d'installation élevé et de de la vitesse lente des Leased Lines, Les technologies Ethernet WAN deviennent plus populaire.

MPLS est l'acronyme de « Multi Protocole Label Switching », de manière similaire à Internet le fournisseur Internet MPLS partagent des Infrastructures car plusieurs entreprises clientes s'y connectent et partagent la même infrastructure pour faire des connexions WAN.

Le label switching dans le nom de MPLS permet aux VPNs d'être crées à travers l'infrastructure MPLS pour l'utilisation de labels, ces labels sont fait pour séparer le trafique de différents clients afin d'être certain qu'il ne mélange pas le trafique avec d'autres clients.

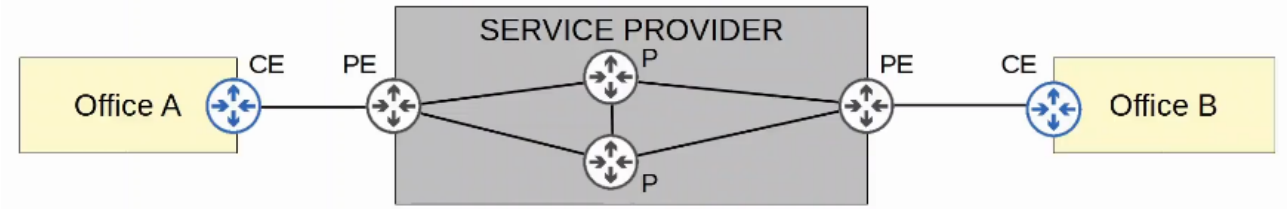
Certains termes important sont :

CE Router : Customer Edge Router

PE Routeur : Provider Edge Router

P Router : Provider Core Router

Le diagramme suivant permet de mieux comprendre :



Le routeur CE est à la limite du routeur client et sont connectés au routeur PE (Provider Edge Router) chez le service fournisseur il y a les routeurs Core Provider qui fournissent le réseau interne à l'infrastructure mais qui ne sont pas connectés directement au routeur du client.

Lorsque le routeur PE reçoit une trame du routeur CE il ajoute un label à la trame.

Ces labels sont utilisés pour faire partager des décisions du service fournisseur du réseau et non pas la destination IP.

Le routeur CE n'utilise pas MPLS, MPLS est utilisé uniquement pas les routeurs PE et P.

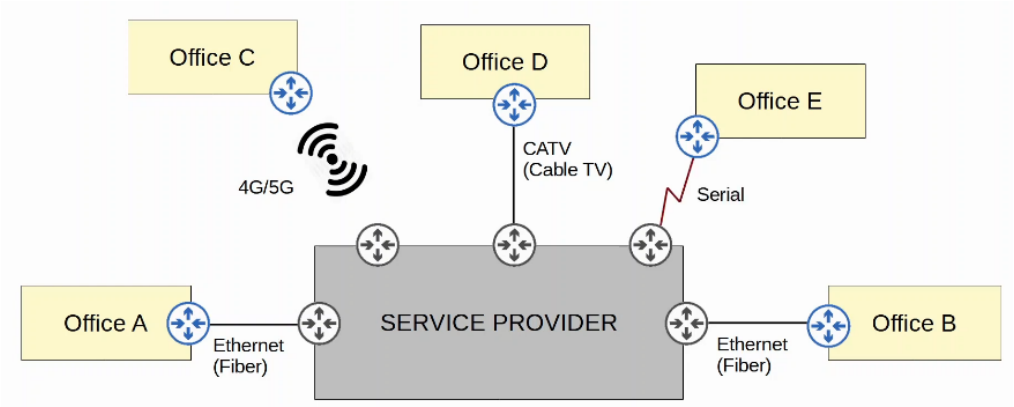
Lorsque l'on utilise la couche 3 MPLS VPN, les pairs de routeurs CE et PE utilisant OSPF par exemple pour partager des informations de routage.

Par exemple dans le diagramme ci dessus le CE d'Office A s'appareille avec le PE et le CE d'Office B s'appareille avec le PE. Le CE d'Office A apprend à propos de Office B ses routes par l'appairage OSPF, et le CE d'Office B apprend aussi des routes d'Office A.

Lorsque l'on utilise une couche 2 de MPLS VPN, les routeurs CE et PE ne forment pas un appairage. Le fournisseur de service réseau est totalement transparent avec les routeurs CE.

Dans les fais c'est comme les deux routeurs CE directement connectés, leurs interfaces WAN sera dans le même sous réseau. Si un protocole de routage est utilisé les deux routeurs CE vont s'appairer directement entre eux, le fournisseur de service fais alors comme office de Switch pour faire passer le réseau des deux routeurs CE.

Différentes technologies peuvent être utilisés pour connecter un fournisseur Internet de réseau MPLS pour un service WAN, Office A et Office B utilisent un câble Ethernet Fibre, Office C utilise La 4G/5G, Office D utilise des câble TV, Office E utilise un câble Serial.



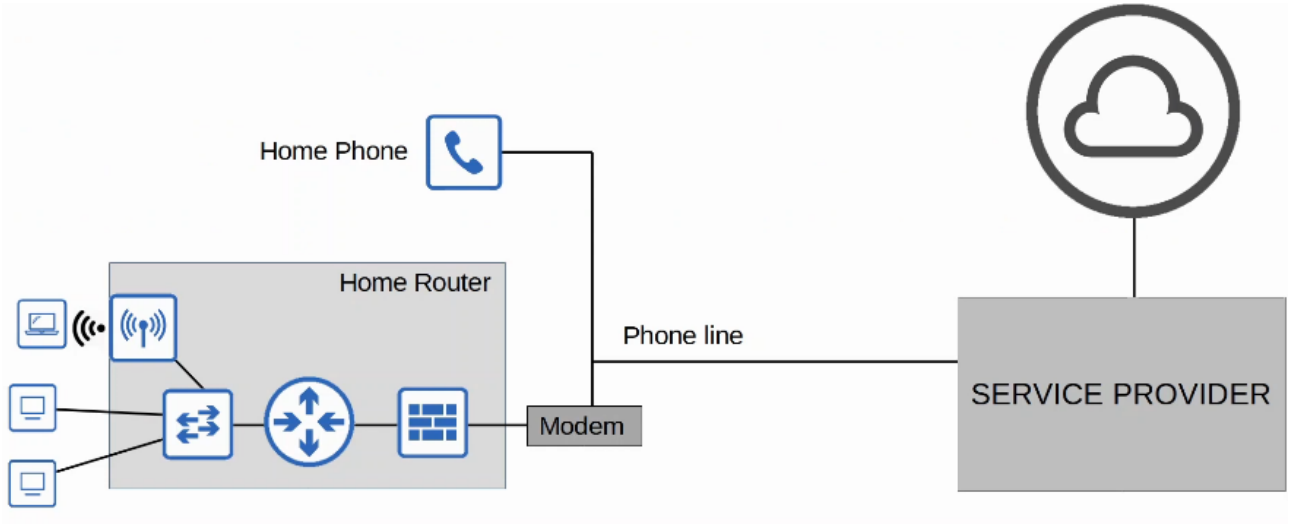
Il existe un tas de manières pour une entreprise de se connecter à Internet. Par exemple une technologie WAN privée comme les Leased Lines et MPLS VPN peuvent être utilisés pour connecter à des fournisseurs de service d'infrastructure Internet.

En plus des technologies comme CATV et DSL communément utilisés par les clients (accès Internet maison) peuvent être utilisés par les entreprises.

De nos jours les entreprises et clients d'accès Internet, les connexions fibres optiques Ethernet augmentent en popularité avec la haute vitesse de connexion qu'ils fournissent sur des longues distances. Voyons à présent deux technologies d'accès Internet mentionnés auparavant :

CATV et DSL.

DSL est l'acronyme de Digital Subscriber Line, il fournit une connectivité Internet aux clients à travers les lignes téléphoniques et peuvent partager la même ligne de téléphone qui est installé dans la plupart des maisons comme sur le diagramme ci dessous :

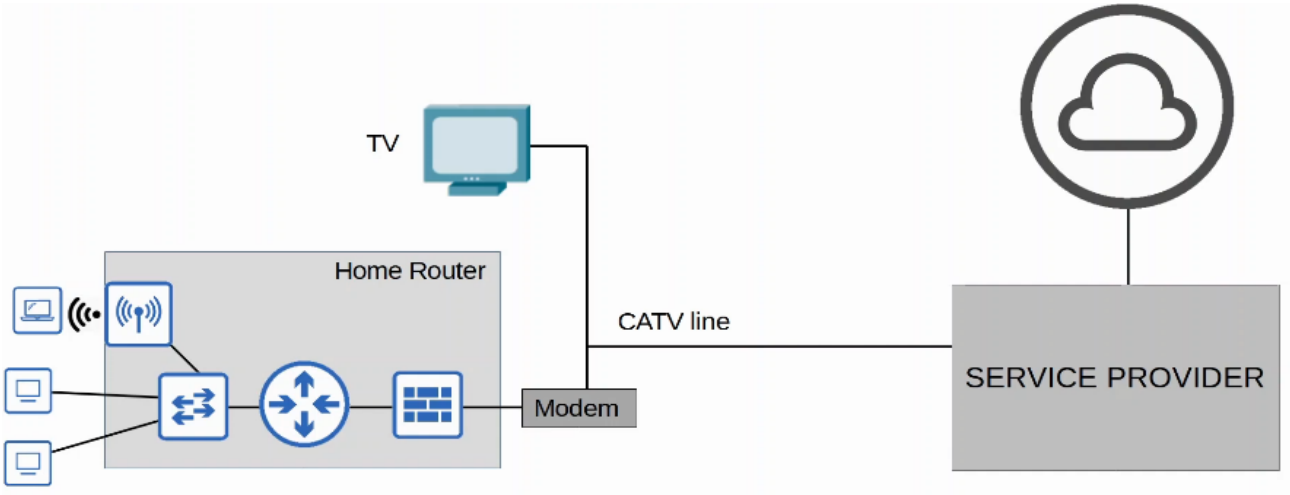


Le modem DSL (Modular demodulator) est requis pour convertir les données en un format approprié pour être envoyé à travers des lignes téléphonique.

Le modem peut être un appareil séparé ou bien être intégré au « routeur de maison ».

Les câbles Internet est un concept similaire à DSL et fournissent un accès Internet par le même CATV (Câble de Télévision) la ligne utilisé pour le service TV.

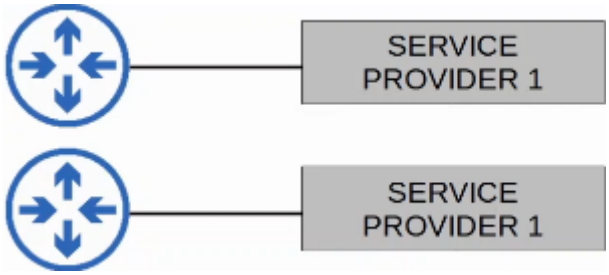
Tout comme DSL un câble modem est requis pour convertir des données en un format adapté pour être envoyé à travers le câble CATV. Tout comme DSL, cela peut être un appareil séparé ou intégré à un « routeur de maison »



Pour les réseaux de maison la perte de connexion Internet n'est pas grave, mais pour une entreprise une perte de connexion peut poser problème, c'est pour cela qu'il est préférable d'avoir une solution redondante. Voici les termes à connaître :

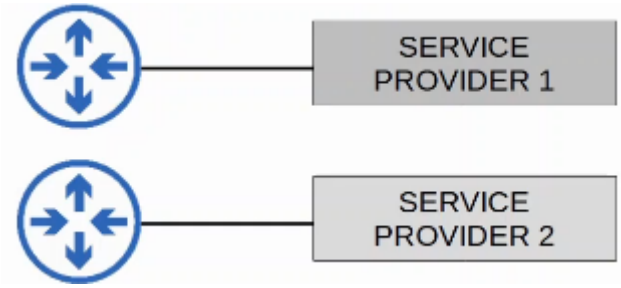
Si une seule connexion à l'ISP est fait cela est appelé : « Single Homed », c'est comme une connexion standard de maison, pour une entreprise ça n'est pas idéal.

S'il y a deux connexions vers l'ISP cela est appelé « Dual Homed », cela fournit une redondance mais n'est pas le plus efficace.

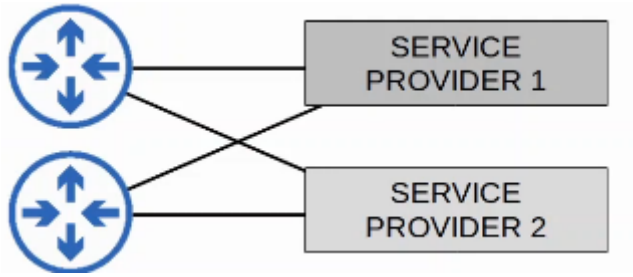


Si une connexion est présente pour chaque ISP cela est appelé « Multihomed ».

Cela améliore la redondance car dans ce cas de figure, si un ISP (fournisseur Internet) rencontre un problème, le deuxième ISP prend le relais.



Lorsque deux connexions pour deux ISP sont présente cela est appelé « Dual Multihomed ». Cela permet la meilleure redondance.



Les services privés WAN comme Leased Lines et MPLS fournissent de la sécurité car chaque trafique client est séparé en utilise une connexion physique dédié (Leased Line) ou par un tag MPLS.

Lorsque l'on utilise Internet avec WAN pour connecter des sites entre eux, il n'y a pas de sécurité préconçu par défaut. Pour fournir une communication sécurisé à travers Internet, les VPNs (Virtual Private Networks) sont utilisés.

Nous verrons deux types de connexions VPN :

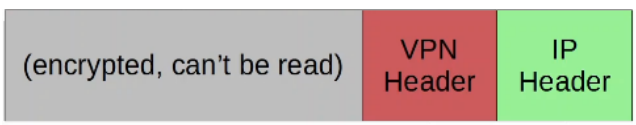
- VPN Site to Site utilisant IPsec

- Remote access VPN utilisant TLS

Un VPN site to Site est un VPN entre deux appareils et est utilisé pour connecter deux sites entre eux à travers Internet.

Sur un VPN « un tunnel » est crée entre les deux appareils en encapsulant le paquet IP original avec l'entête VPN et la nouvelle entête IP.

Lorsque l'on utilise IPsec, le paquet original est crypté avant d'être encapsulé avec la nouvelle entête.



Le paquet suivant est envoyé à travers le réseau :

L'appareil envoyeur combine le paquet original et la session de clé (cryptage de clé) et les lance à travers une formule de cryptage. L'appareil envoyeur encapsule le paquet crypté avec une entête VPN et la nouvelle entête IP. L'appareil envoyeur envoie le nouveau paquet à l'appareil placé de l'autre côté du tunnel. Celui ci décrypte les données pour recevoir le paquet original et repartage le paquet original vers sa destination.

Dans un VPN Site to Site, un tunnel est formé seulement entre les deux extrémités du tunnel (Par exemple, les deux routeurs connectés à Internet)

Tous les autres appareils de chaque site n'ont pas besoin de créer un VPN pour eux même. Ils peuvent envoyer des données non cryptés à leur routeur de site qui va crypter et repartager dans le tunnel comme décrit auparavant.

Il y a certaines limitations au standard IPsec :

IPsec ne supporte pas le trafique Broadcast et Multicast mais seulement le Unicast. Cela signifie que le protocole de routage comme OSPF ne peuvent pas être utilisés à travers les tunnels puisqu'ils se basent sur un trafique multicast. Cela peut être résolu avec « GRE over IPsec »

Un autre problème majeur est que la configuration d'un tunnel totalement maillé entre plusieurs sites est une tâche laborieuse et demande beaucoup de temps.

Cela peut être résolu avec la solution Cisco DMVPN.

Voyons rapidement chacune des solutions précédemment évoqués :

- GRE over IPsec, GRE est l'acronyme de Generic Routing Encapsulation, permet de créer des tunnels comme Ipsec, quand bien même cela n'encrypte pas le paquet original donc cela n'est pas sécurisé.

Il a tout de même l'avantage d'avoir la possibilité d'encapsuler un grande variété de protocole de couche 3 comme messages Broadcast et multicast.

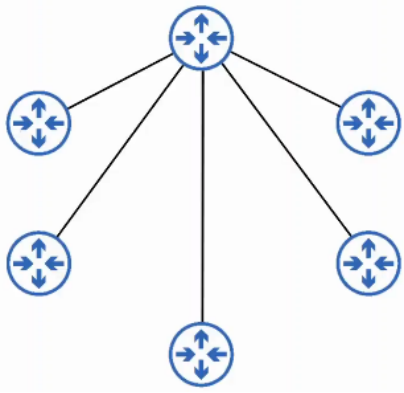
Pour avoir la fléxibilité de GRE avec la sécurité de IPsec, « GRE over IPsec » peut donc être utilisé.

Le paquet original sera encapsulé par une entête GRE et une nouvelle entête IP, puis le paquet GRE est crypté et encapsulé, l'entête IPsec VPN avec la nouvelle entête IP est ensuite ajouté.

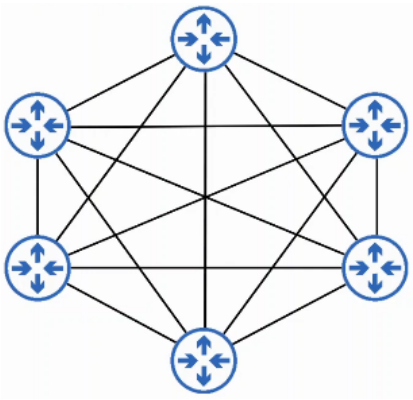
- DMVPN est l'acronyme de Dynamic Multipoint VPN, cette solution a été développé par Cisco et permet aux routeur de dynamiquement créer un maillage total d'un tunnel IPsec sans avoir à configurer manuellement chaque tunnel.

Voici comment le configurer en deux étapes :

1. configuration du tunnel IPsec à un site hub, ci dessous le routeur au dessus est le hub et les autres routeurs sont les spoke qui sont connectés avec un tunnel IP vers le hub.



2. Le routeur hub donne à chaque routeur l'information à propos de comment former un tunnel IPsec avec les autres routeurs.



Les VPN site to Site sont utilisés pour faire une connexion site to site entre deux sites à travers Internet, les remote access VPN sont utilisés pour autoriser les appareil finaux (PC, téléphone) d'accéder aux ressources interne de manière sécurisé à travers Internet.

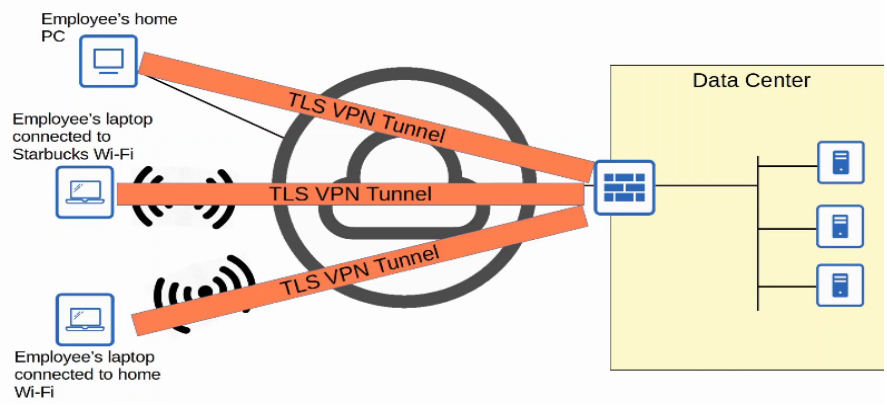
Remote Access VPN utilise TLS (Transport Layer Security), TLS est aussi ce que fournit comme sécurité pour HTTPS (HTTP sécurisé), TLS est comme SSL (Secure Sockets Layer) et a été développé par Netscape, mais a été renommé en TLS lorsqu'il a été standardisé par IETF.

Les logiciels clients VPN (comme Cisco Anyconnect) est installé sur l'appareil final (par exemple une entreprise fournit des ordinateurs portables à ses employés qui utilisent pour travailler depuis chez eux)

Ces appareil finaux forment un tunnel sécurisé à l'une des compagnies du routeur/Firewall qui fonctionne comme serveur TLS.

Cela permet à l'utilisateur final de sécuriser l'accès des ressources du réseau interne de l'entreprise sans être directement connecté au réseau de l'entreprise.

Voici un diagramme pour mieux visualiser :



Les différences entre Site to Site VPN et Remote Access VPN sont les suivantes :

Les VPN Site to Site utilisent IPsec tandis ce que les Remote Access VPN utilisent TLS.

Les VPN Site to Site fournissent des services à plusieurs appareils au sites auxquelles ils sont connectés tandis ce que les Remote Access VPN fournissent des services à l'appareil final sur lequel est installé le logiciel client VPN.

Les VPN site to Site sont utilisés pour connecter de manière permanente deux sites à travers internet tandis ce que les Remote Access VPN sont utilisés pour fournir un accès sur demande pour les appareils finaux qui veulent accéder de manière sécurisé aux ressources d'une entreprise au lieu de se connecter à un réseau qui n'est pas sécurisé.

Cours 54 : Virtualisation & Cloud

Dans ce cours nous verrons deux sujet qui sont la virtualisation et le Cloud.

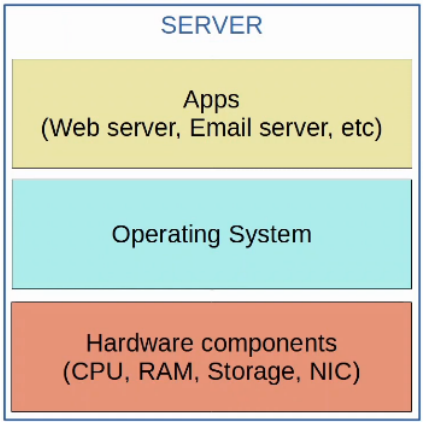
Nous ferons d'abord une introduction du fonctionnement de la virtualisation avec les serveurs virtuels et les réseaux virtuels. Puis nous ferons une introduction sur le Cloud Computing en donnant ses caractéristiques essentiels, ses modèles de services et les modèles de déploiements.

Nous verrons comment faire pour se connecter à des Cloud publique.

Cisco est l'un des vendeurs les plus connu pour ses appareils de réseaux comme les routeurs, les Switchs, les Firewall, ils offrent aussi du matériel de serveurs comme UCS (Unified Computing System). Voici un exemple de à quoi ressemble UCS :

D'autres grand vendeurs sont aussi présent sur le marché qui sont Dell EMC, HPE et IBM.

Voyant tout d'abord comment un serveur fonctionne sans virtualisation :



Avant la virtualisation il y avait une relation direct entre le serveur physique et le système d'exploitation (OS pour Operating System). Dans ce système d'exploitation des applications qui fournissent différents services comme le serveur Web, le serveur de Mail, etc...

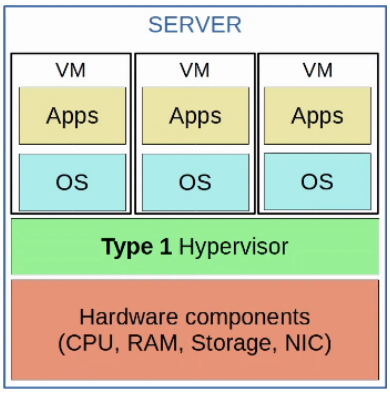
Un serveur physique serait utilisé pour le serveur Web, le serveur Mail, la base de données, etc...

Cela n'est pas efficace pour plusieurs raisons :

Chaque serveur physique coûte chère et prend de la place, de l'énergie etc...

Les ressources pour chacun des serveurs (CPU, Stockage, RAM, NIC) sont parfois sous utilisés.

Avec la virtualisation cela est différent :



La virtualisation permet de changer la traditionnel relation entre le matériel de l'OS ce qui permet de lancer plusieurs systèmes d'exploitation dans un seul serveur physique.

Chaque instance est appelé une VM (Virtual Machine ou Machine Virtuel en Français).

Sur le diagramme précédent, 3 systèmes d'exploitation sont lancés sur un seul serveur.

Un hyperviseur est utilisé pour gérer et allouer les ressources matériel (CPU, RAM, etc.) pour chacune des VM.

Un autre nom pour les hyperviseur est VMM (Virtual Machine Monitor)

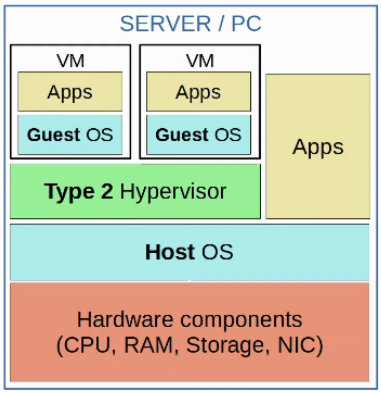
Le type d'hyperviseur qui se lance directement en haut du matériel est appelé hyperviseur de type 1.

Des exemples incluent Vmware ESXi, Microsoft Hyper-V, etc...

Les hyperviseurs de type 1 sont aussi appelés « bare-metal hypervisors » car ils se lancent directement sur le matériel (qui est du métal). Un autre terme est « native hypervisor ».

C'est le type d'hyperviseur qui est utilisé dans un environnement de centre de données.

Les hyperviseurs de Type 2 se lancent comme un programme sur un système d'exploitation comme des programmes normaux. Par exemple cela inclus VMware Workstation, Oracle VirtualBox, etc.



L'OS qui se lance directement sur le matériel est appelé Host OS, et l'OS qui se lance dans la VM est appelé Guest OS. Un autre nom pour les hyperviseurs de Type 2 est « hosted hypervisor »

Les hyperviseurs de type 2 sont rarement utilisés dans des environnements de data center, ils sont commun dans l'utilisation personnel des appareil (par exemple si un utilisateur MAC/Linux a besoin de lancer une application qui est seulement supporté par Windows et l'inverse).

Sur le site de Vmware est donné plusieurs informations sur la virtualisation. La virtualisation permet le partitionnement, en lançant plusieurs systèmes d'exploitations sur une seule machine physique, et permet de diviser les ressources systèmes entre des machines virtuel.

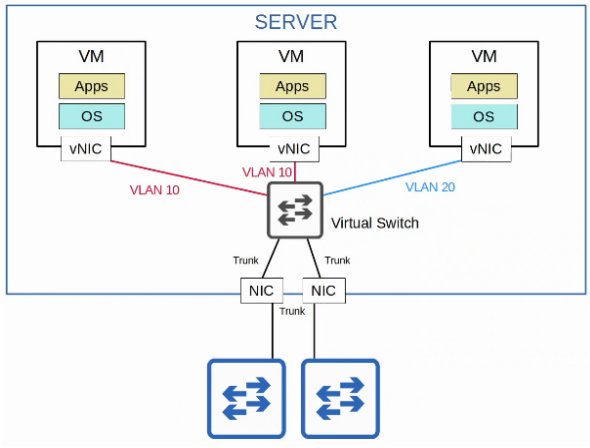
La virtualisation permet aussi l'isolation en fournissant isolation de faute et de sécurité à un niveau matériel mais aussi en préservant les performance avec un contrôle avancé des ressources.

La virtualisation permet l'encapsulation qui sauvegarde l'état entier d'un machine virtuel dans un fichier, et change et copie la machine virtuel aussi facilement que déplacer et copier des fichiers.

La virtualisation permet l'indépendance du matériel en provisionnant ou migrant n'importe quelle machine virtuel vers n'importe quelle serveur physique.

La virtualisation permet une réduction des coût conséquent car il y a moins de serveurs physique et donc de coût dans l'utilisation de l'énergie du serveur. Les VM requièrent aussi moins de temps à configurer. Les VM sont bien plus rapide et productives comparé au traditionnel serveur uniquement physique.

Voici une explication de comment les VM peuvent se connecter entre elles vers un réseau externe de l'hôte physique.



Les VM sont connectés entre elles avec le réseau externe par le Switch virtuel qui se lance sur l'hyperviseur. Tout comme un Switch normal l'interface vSwitch peut fonctionner comme port trunk ou access et utiliser des VLANs pour séparer les VM en couche 2.

Les interfaces sur le vSwitch se connectent ensuite au NIC physique du serveur pour communiquer avec le réseau externe.

Il est possible d'utiliser un VPC (Virtual Port Channel) pour former un port channel entre deux Switch pour une redondance.

Voyons à présent le fonctionnement du Cloud.

Le déploiement d'infrastructure IT traditionnel se fait en général comme suit :

- On Premises : Tous les serveurs, appareils réseau et autres infrastructure sont localisé par la propriété de l'entreprise. Tous les équipements sont achetés et acquis par l'entreprise pour être utilisé. L'entreprise est responsable de l'espace nécessaire, la puissance et la ventilation.

- Colocation : Les Data center qui vendent des espaces pour leurs clients afin qu'ils placent leurs infrastructures (Serveurs, appareils réseau). Le Data center fournit l'espace, l'électricité et la ventilation pour tous les appareils. Les serveurs, appareils réseau, etc.. sont toujours sous la responsabilité du client final, bien que le matériel ne soit pas localisé dans les locaux de l'entreprise cliente.

Les services Cloud fournissent une alternative très populaire et qui continue à augmenter en popularité. La plupart des gens associent « cloud » avec un fournisseur cloud publique comme AWS car c'est l'utilisation la plus commune du Cloud mais ça n'est pas la seule option existante.

Le American NIST (National Institute of Standards and Technology) définit le Cloud Computing dans le SP (Special Publication) 800-145. Il est possible de le lire sur ce lien : https ://csrc.nist.gov/publications/detail/sp/800-145/final

Voici la définition en Anglais donnée par NIST :

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Voici les 5 caractéristiques essentiels du cloud computing :

- On demand self-service : le service sur demande consiste en un client qui utilise des capacités informatique comme le temps d'un serveur, le stockage dans un réseau et qui change automatiquement selon le besoin sans que cela ne requière d'interaction humaine avec chaque fournisseur de service. Le client a donc la possibilité d'utiliser le service (ou le stopper) gratuitement (par le portail web) sans avoir à communiquer à un fournisseur de service.

- Broad network access : Les capacités sont disponible sur le réseau et accessible par un mécanisme standard qui encourage l'utilisation fine ou épaisse de plateforme client (Par exemple des téléphones mobile, tablettes, ordinateurs portable, et station de travail). Le service est disponible par une connexion réseau standard (comme une connexion internet privée ou WAN) et peut être accessible par plusieurs type d'appareils.

- Ressource pooling : Les ressource Informatique du fournisseur sont regroupés pour servir plusieurs clients en utilisant un model multi locataire, avec différentes ressources physique et virtuels qui sont assignés dynamiquement et réassignés en accord avec la demande client. Il y a un sens de localisation indépendant dans cela le client généralement n'a pas de contrôle ou connaissances de la localisation exacte des ressources fournis mais est capable de spécifier la localisation à un haut niveau d'abstraction (par exemple le pays, ou le datacenter). Des exemples de ressources incluent le stockage, le processeur, la mémoire, et la bande passante réseau. Pour résumer, un groupement de ressource est fournit par le fournisseur de service et lorsque l'utilisateur fait la requête d'un service (par exemple créer une nouvelle VM), les ressources pour répondre à cette requête sont alloués depuis le groupement de ressource partagés.

- Rapid elasticity : Ces capacités peuvent être de manière élastique fournit et publiés, dans certains cas automatiquement, pour échelonner rapidement vers l'intérieur ou l'extérieur proportionnelement à la demande. Pour le client ces capacités sont disponible pour fournir souvent et apparaissent comme étant illimité et peuvent être acquise dans n'importe quelle quantité à n'importe quelle moment. Le client peut dont rapidement étendre les services qu'ils utilisent dans le Cloud (par exemple ajouter des Vms, étendre le stockage, etc...) depuis un groupe de ressource qui apparaissent comme illimités. Dans un autre sens ils peuvent rapidement réduire leurs services lorsque non nécessaire.

- Measured service : Le système Cloud contrôle automatiquement et optimise la ressource utilisé en mesurant la capacité à un certain niveau d'abstraction appropriés au type de service (par exemple le stockage, le processeur, la bande passante, et les comptes utilisateurs). L'utilisation de ressource peut être géré, contrôlé, et reporté, ce qui fournit une transparence pour le fournisseur et le client des service utilisés. Le fournisseur de service Cloud mesure l'usage client en ressource Cloud, et le client peut mesurer sa propre utilisation. Les clients sont chargés basé sur l'utilisation (Par exemple, X Dollar par Gigabyte de stockage par jour)

Voyons les 3 modèles de service du Cloud. Dans le cloud computing tout est fournit par un modèle de service. Par exemple au lieu que le client achète un serveur physique, le monte sur un rack, installe un hyperviseur, crée les Vms, etc. le fournisseur de service offre tout cela comme un service. Il y a une variété de services qui prend la forme « …. as a Service » or « ...aaS »

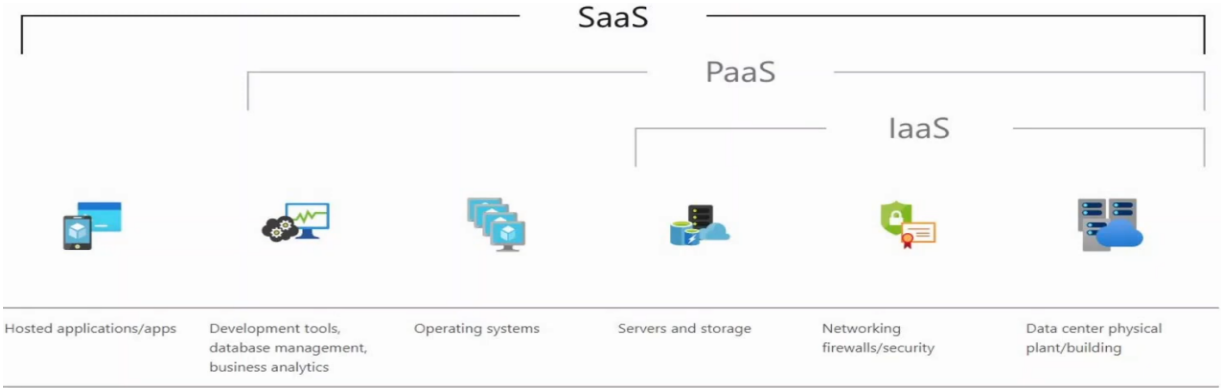
Les trois modèles de service du cloud computing sont :

- Software as a Service (SaaS) : La capacité fournit au client est d'utiliser le fournisseur d'applications lancé sur l'infrastructure Cloud. Les applications sont accessible depuis des appareils client variés comme une interface client « fine » comme un navigateur web, ou une interface d'un programme. Le client ne gère ou ne contrôle pas les sous niveaux de l'infrastructure Cloud en incluant le réseau, les serveurs, l'OS, le stockage, ou même les capacité individuel de chaque applications, mais une possible exception sur la limite d'un utilisateur spécifique des paramètres de configuration. Microsoft 365 est un exemple populaire de SaaS.

- Platform as a Service (PaaS) : La capacité fournit au client est de déployer dans l'infrastructure Cloud les créations du clients ou applications acquises créés en utilisant un langage de programme, des librairies, des services, et des outils supportés par le fournisseur. Le client ne gère ou ne contrôle pas les sous couches de l'infrastructure Cloud en incluant le réseau, les serveurs, l'OS, ou le stockage, mais a le contrôle à travers les applications déployés et possiblement les paramètres de configuration pour environnement de l'application hôte. Des exemples d'offres sont AWS Labda et Google App Engine.

- Infrastructure as a Service (IaaS) : La capacité fournit au client est de provisionner le stockage, le réseau et d'autres ressources fondamentales Informatique ou le client est capable de déployer et lancer des logiciel arbitraire, qui incluent des OS et applications. Le client ne gère pas ou ne contrôle pas les sous couches de l'infrastructure cloud mais a le contrôle sur le système d'exploitation, le stockage, les applications déployés, et possibilité de limiter le contrôle de composants réseau sélectionné (par exemple un Firewall). Des exemples d'offres sont Amazon EC2 et Google Compute Engine.

Voici une image provenant de chez Microsoft qui permet de résumer les modèle de Cloud :



Voyons les 4 formes de déploiement de Cloud.

Pour la plupart des gens le Cloud signifie des fournisseur Cloud comme AWS, Azure et GCP.

Le Cloud publique est le modèle de déploiement le plus commun mais il ne s'agit du seul moyen de déploiement Cloud. Il en existe 4 :

- Private Cloud : L'infrastructure cloud est fournit pour l'utilisation exclusive par une seule organisation comprenant plusieurs clients (Par exemple des unités de Business). Cela peut être acquis, géré, et modifié par l'organisation, un tiers partie, ou certaines combinaisons de cela, et peut exister en « on » ou « off » premise. Les Cloud privée sont généralement uniquement utilisé par de grandes entreprises. Bien que le cloud soit privée, il peut appartenir à des tiers partie comme par exemple des service fournisseurs AWS privée cloud pour le American DoD. Les cloud privée peuvent être « on » ou « off » premise. De nombreuses personne assument que le Cloud et On premise sont deux choses différentes, mais ça n'est pas toujours le cas. Le même type de services offerts sont les même que sur le cloud publique (SaaS, PaaS, IaaS), mais l'infrastructure est réservé pour une seule organisation.

- Community Cloud : Cette infrastructure cloud fournit une utilisation exclusive par une communauté spécifique de clients depuis des organisations qui concernent des partages (par exemple des mission, des prérequis de sécurité, prérequis de conformité). Cela peut appartenir, être géré et fonctionner par un ou plus des organisation dans la communauté, un tiers partie, ou une combinaison des deux, et peut exister en « on » et « off » premise. C'est le type de déploiement Cloud le moins fréquent. C'est similaire au Cloud privée mais l'infrastructure est réservé pour être utilisé uniquement par un groupe spécifique ou des organisations.

- Public Cloud : Cette Infrastructure cloud est fourni pour une utilisation ouverte par un publique général. Cela peut appartenir, être géré, et fonctionner par un Business, une académie, ou une organisation du gouvernement, ou une combinaison de cela. Il peut exister en on premise du fournisseur Cloud. C'est le modèle de Cloud le plus fréquent. Des service cloud populaire inclus :

AWS (Amazon Web Service), Microsoft Azure, GCP (Google Cloud Platform), OC (Oracle Cloud Infrastructure), IBM Cloud, Alibaba Group.

- Hybrid Cloud : Cette infrastructure Cloud est composé de deux ou plus d'infrastructures Cloud (Privée, communauté, ou publique) qui reste une entité unique mais qui sont lié ensemble par une technologie propriétaire ou standardisé qui permet la portabilité des données et des application (Par exemple Cloud bursting pour load balance entre plusieurs Cloud). Cette une combinaison des trois différents type de déploiement précédemment vu. Par exemple un cloud privée qui peut décharger ses ressources vers un cloud publique lorsque nécessaire.

Les bénéfices du cloud computing sont les suivant :

- Coût : CapEx (Capital Expenses) les coûts de l'achat du matériel et logiciel, mise en place des data centers, etc. sont réduis ou éliminés.

- Scalabilité Global : Les services Cloud peuvent être évolutif globalement avec une mis en place rapide. Les services peuvent être mis en place et offets aux clients depuis une localisation géographique proche de chez eux.

- Rapidité/Agilité : Les services sont fournis sur demande, et un vaste montant de ressources peut être provisionné dans les minutes.

- La productivité : Les services Cloud supprime le besoin pour plusieurs tâches de consommation de temps comme de se procurer un serveur physique, le rackage, le câblage, l'installation, la mise à jour des systèmes, etc...

- Fiabilité : Les Backups sur le Cloud sont vraiment facile à fonctionner. Les données peuvent être mis en miroir (copiés) sur plusieurs sites sur des localisations géographique différentes pour supporter la récupération des désastres (feu, inondation etc..)

Il faut garder en tête que le Cloud n'est pas toujours la meilleure option. La plupart des compagnies de nos jours utilisent une combinaison d'équipements On premises, Colocation et Publique Cloud.

Une compagnie ne devrait pas utiliser le Cloud seulement puisque c'est populaire de nos jours.

Voyons comment une entreprise fait pour connecter son réseau aux ressources d'un cloud publique.

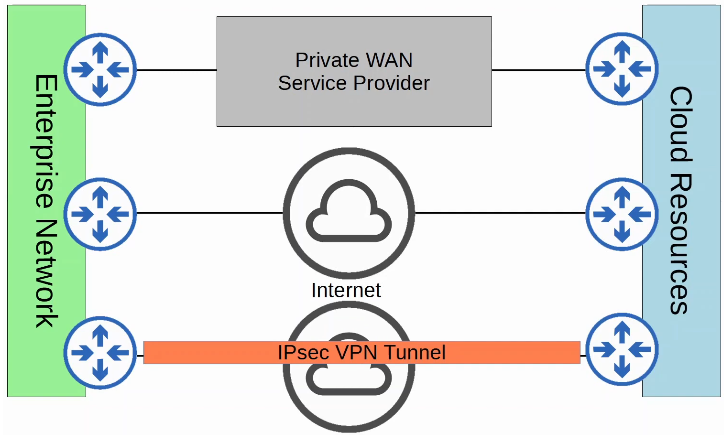
Il existe plusieurs moyens qui peuvent être :

- Un WAN privée par un fournisseur de services

- Un service Internet

- Un tunnel VPN IPsec

Le schéma suivant résume cela :



Cours 55 : Fondamental Sans Fil

Dans ce cours nous allons apprendre les fondamentaux d'un réseau sans fil.

Nous commencerons par faire une introduction sur les fréquences radios (RF), RF est un classement de fréquences électromagnétiques qui ont été assignés pour plusieurs but différents, en incluant les AM et FM radio, micro-onde, radar et le Wifi.

Le Wifi est l'onde qui nous intéresse le plus, bien évidemment puisque les LANs sans fil, utilisent le Wifi. Nous verrons ensuite les différents Standard Wifi comme définis dans le IEEE 802.11. Comme de nombreuses variétés de Standard Ethernet sont définis dans le IEEE 802.3, de nombreux Standard LAN sans fil sont définis dans le IEEE 802.11.

Nous verrons certains fondamentaux des LAN sans fil et en quoi ils diffèrent des LAN en câblé.

Commençons par donner une introduction des réseaux sans fil spécifiquement les LANs sans fil. Aussi nous verrons rapidement d'autres types de réseaux sans fil, dans cette section du cours nous nous concentrerons sur les LANs sans fil qui utilisent le Wifi.

Les standards que l'on utilise pour les LANs sans fil sont définis dans le IEEE 802.11

Le terme Wifi est une marque déposé de « Alliance Wifi », pas directement connecté au IEEE. Le « Wifi Alliance » teste et certifie l'équipement pour les Standard 802.11 et la compatibilité avec et l'interopérabilité avec les autres appareils.

Les appareils qui ont eu la certification « Wifi », peuvent utiliser le symbole Wifi Certified comme ci dessus. Dans l'exemple de la photo sur l'arrière d'un point d'accès sans fil Cisco on peut voir que le produit a été certifié par le Wifi Alliance.

Le Wifi n'est pas techniquement le bon terme pour se référer au 802.11 LAN sans fil.

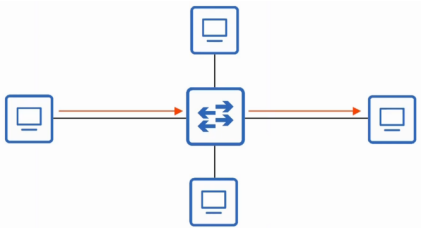
Quand bien même le Wifi est devenue le terme le plus commun pour se référer au 802.11 sans fil et aux LANs nous utiliserons les deux termes dans ce cours.

Les réseaux sans fil ont des problèmes dont nous avons déjà parlés.

Tout d'abord tous les appareils avec un classement d'adresses reçoivent tous les paquets, comme les appareils connectés à un Hub Ethernet.

Lorsque l'on utilise un Switch Ethernet, le Switch est capable de partager les paquets seulement à l'appareil concerné. En plus de cela les Switchs permettent aux appareils de fonctionner en utilisant le full duplex, donc les appareils peuvent tout deux envoyer et recevoir des trames en même temps et les collisions ne devraient pas se passer jusqu'à qu'il y ait certains problèmes dans le réseau.

Par exemple dans cette exemple :



Lorsque l'on utilise un Hub Ethernet, le hub va toujours inonder tous les paquets qu'il reçoit et les collisions peuvent se passer lorsque plusieurs appareils essayent d'envoyer et de recevoir le trafique au même moment.

1) Comme avec les appareils connectés à un hub Ethernet lorsqu'un appareil sans fil transmet un paquet tous les appareils sans fil connectés avec certaines ondes seront capable de recevoir cette trame le signal n'est pas contenu dans un appareil câblé puisque le signal est composé d'ondes électromagnétiques à partir des appareils de transmission.

- Cela peut faire apparaître des problèmes aux données privés lorsque les collisions apparaissent puisqu'en comparaison aux réseaux câblés on n'encrypte pas les données contenus dans le LAN, seulement lorsque l'on envoie des données sur un réseau partagé comme sur Internet. Pour les réseaux sans fil il est très important d'encrypter les données dans le LAN, ou bien n'importe qui d'autre avec un appareil dans le classement du transmetteur pourra accéder à ces données.

- Aussi pour éviter les collisions et faciliter les communications Half-Duplex , CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) est utilisé pour faciliter les communications Half-Duplex.

Ce mot ressemble à un autre mot similaire qui est le CSMA/CD qui est utilisé dans les réseaux câblés pour détecter et empêcher les collisions.

Le CSMA/CA est lui utilisé pour les réseaux sans fil pour empêcher les collisions.

Lorsque l'on utilise CSMA/CA, un appareil attendra en stoppant son trafique pour attendre que les autres appareils aient transmis leurs données avant de transmettre lui même ses données.

Si on voit le processus étape par étape le processus est le suivant :

Les trames sont tout d'abord assemblés, puis il attendra pour une période aléatoire de temps, si la chaîne n'est pas disponible, il attendra pour une période de temps aléatoire puis il écoutera le réseau encore une fois. Si la chaîne est disponible cette fois, il va transmettre la trame. C'est une simplification du processus en vérité il y a une fonction optionnel pour savoir sur quelle appareil de transmission l'appareil enverra une « request to send » (RTS), le paquet va attendre pour un « clear to send », (CTS), le paquet depuis le récepteur avant d'actuellement envoyer les données du paquet. Mais c'est seulement une information en plus.

2) Les communications sans fil sont régulés par des variétés de firmes national et international. Il n'est pas permis de transmettre des données sur n'importe quelle chaîne que l'on désire, et la chaîne l'on est permis d'utiliser peut aussi varier sur le pays.

Le Standard 802.11 précise quelles fréquences peuvent être utilisés pour des LANs sans fil, et les appareils sont conçus pour utiliser ces fréquences.

3) Nous devons aussi prendre en considération les zones de recouvrement des signaux sans fil. Dans les connexions câblés nous devons considérer la longueur du câble et dans certains cas les interférences électromagnétiques, mais avec les connexions sans fil, il y a d'autres facteurs à prendre en compte :

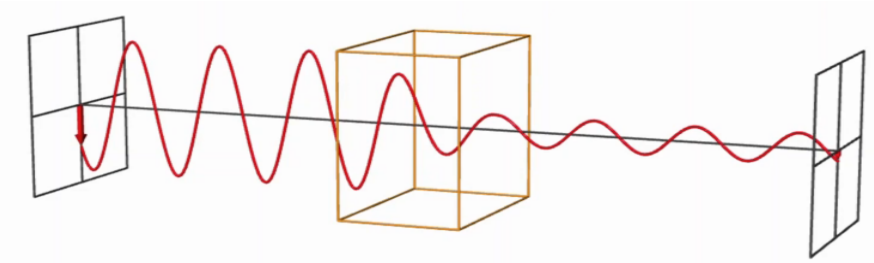
- Tout d'abord le classement du signal, à quelle puissance le signal peut actuellement traverser. Et il y a plusieurs facteurs qui affectent à quelle puissance le signal peut traverser sans être modifié. Ces facteurs sont : l'absorption, la réflexion, la réfraction, la diffraction, et l'éparpillement.

Regardons plus en détail chacun de ces facteurs :

- L'absorption se passe lorsqu'un signal sans fil passe à travers un matériel et est modifié par la chaleur ce qui abîme le signal original.

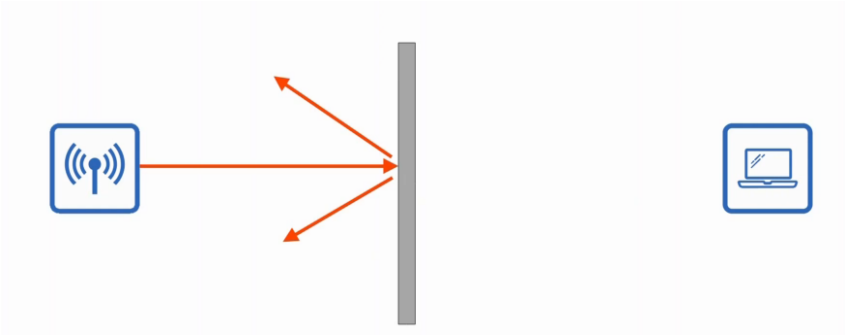
Le signal rebondie en dehors du métal et ne le pénètre que très peu par exemple à l'intérieur d'un ascenseur.

En voici un exemple :

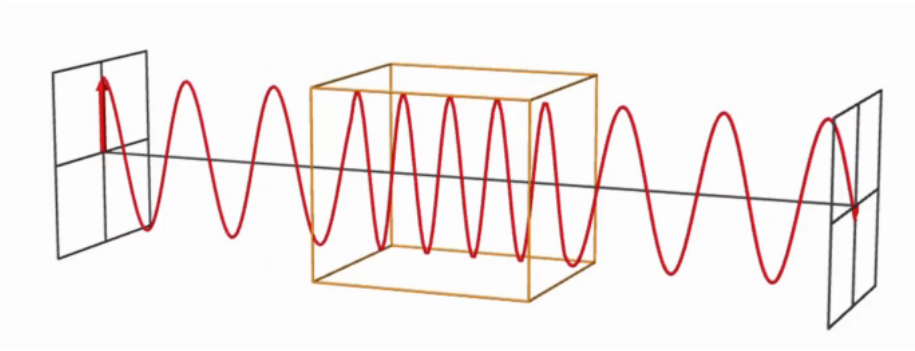


- La réflexion se passe lorsqu'un signal rebondit sur du matériel par exemple du métal.

C'est pourquoi la réception du Wifi est mauvaise dans un ascenseur, car le signal rebondit sur le métal et ne passe que très peu à travers l'ascenseur.

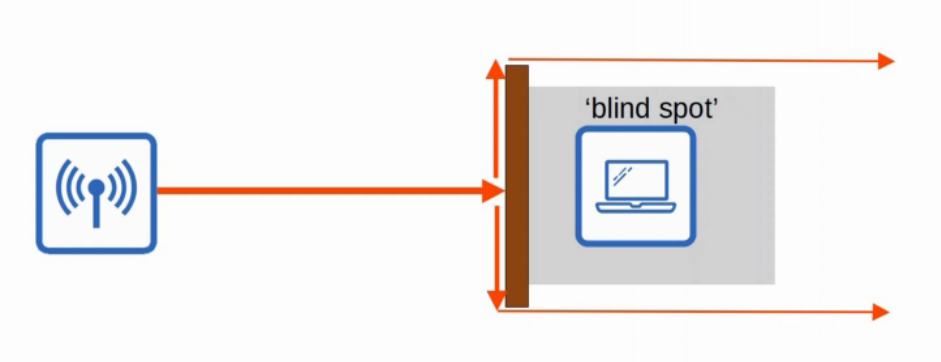


- La réfraction se passe lorsqu'une onde passe dans une surface ou le signal peut traverser différentes vitesses. Par exemple de la glace, de l'eau, peuvent réfracter les ondes.

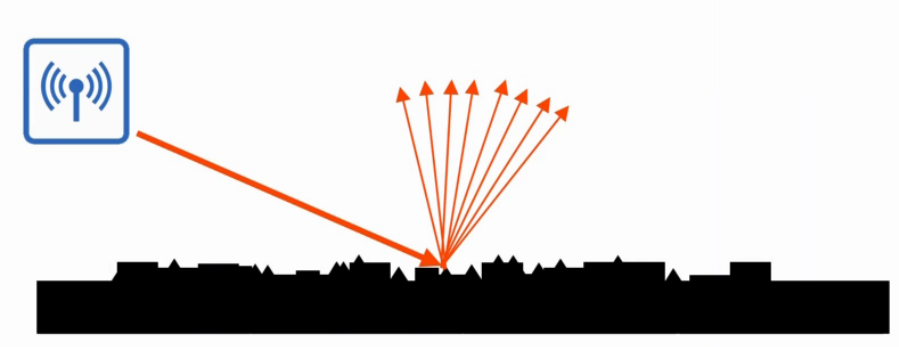


- La diffraction se passe lorsqu'une onde rencontre un obstacle et le contourne.

Cela peut résulter à ce qu'il y ait des « zones aveugles ».



- La dispersion apparaît lorsqu'un matériel cause au signal de se disperser dans toutes les directions. Par exemple le sable, la poussière, etc.. peuvent causer le dispersement du signal.



Tous ces phénomènes : l'absorption, la réflexion, la réfraction, la diffraction, et l'éparpillement sont des facteurs qui affectent la qualité d'un signal sans fil.

Lorsque l'on planifie une architecture sans fil, il faut prendre tous ces facteurs en compte.

4) Une dernière chose à prendre en considération est l'interférence. Les autres appareils qui utilisent la même chaîne peuvent causer des interférences.

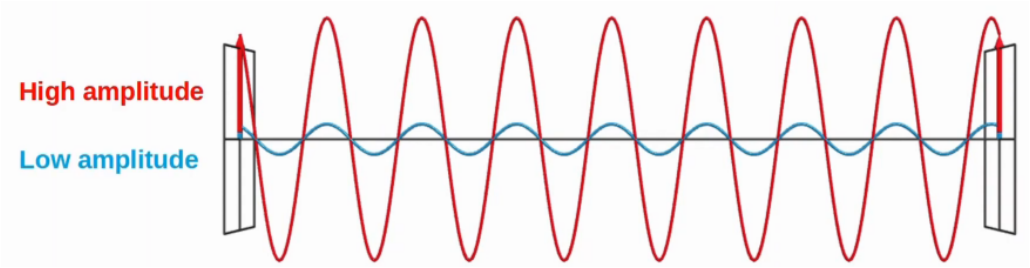
Par exemple, une LAN sans fil dans un appartement ou une maison connecté avec des appareils dans la même chaîne.

Pour envoyer des signaux sans fil, l'envoyeur applique un courant alternatif à une antenne. Cela crée des champs électromagnétique qui se propagent en ondes.

Les ondes électromagnétiques peuvent se mesurer de plusieurs façons, par exemple en amplitude et en fréquences.

L'amplitude est la longueur maximal d'un champs électrique et magnétique.

Par exemple dans l'exemple suivant, en bleu se trouve la plus haute amplitude et en rouge la plus basse amplitude.



La fréquence mesure le nombre de cycle haut/bas par unité de temps donné.

La mesure la plus commune de mesure de fréquence est le hertz.

→ Hz (Hertz) = cycles par secondes.

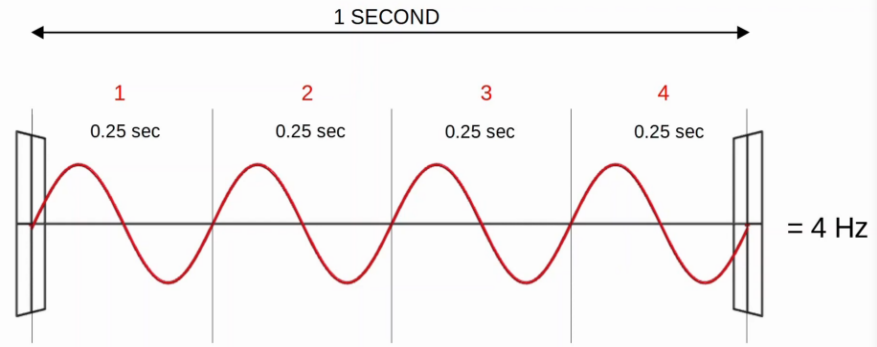
→ kHz (Kilohertz) = 1 000 cycles par secondes.

→ mHz (Megahertz) = 1 000 000 cycles par secondes.

→ GHz (Gigahertz) = 1 000 000 000 cycles par secondes.

→ THz (Terahertz) = 1 000 000 000 000 cycles par secondes.

Par exemple si l'on veut mesurer le nombre en Hz de fréquences de cette image ci dessous :



On compte le nombre de cycle pour la seconde, ici on peut en compter 4, ce qui fais 4Hz.

Un autre terme important est la période, il s'agit du montant de temps d'un cycle.

Donc si une fréquence est de 4Hz, la période est de 0,25 secondes.

Les fréquences visible ont pour classement de 400THz à 790THz.

Les fréquences radio on pour classement de 3Hz à 300GHz et ont de nombreuses utilités.

Voici un tableau tiré de Wikipédia avec contenu tout le classement des différentes fréquences :

Désignation internationale	Désignation francophone	Fréquence	Longueur d'onde	Autres appellations	Exemples d'utilisation
ELF (*extremely low frequency*)	EBF (extrêmement basse fréquence)	3 Hz à 30 Hz	100 000 km à 10 000 km		Détection de phénomènes naturels
SLF (*super low frequency*)	SBF (super basse fréquence)	30 Hz à 300 Hz	10 000 km à 1 000 km		Communication avec les sous-marins
ULF (*ultra low frequency*)	UBF (ultra basse fréquence)	300 Hz à 3 000 Hz	1 000 km à 100 km		Détection de phénomènes naturels
VLF (*very low frequency*)	TBF (très basse fréquence)	3 kHz à 30 kHz	100 km à 10 km	ondes myriamétriques	Communication avec les sous-marins, Implants médicaux, Recherches scientifiques…
LF (*low frequency*)	BF (basse fréquence)	30 kHz à 300 kHz	10 km à 1 km	grandes ondes ou ondes longues ou kilométriques	Radioamateur, Radionavigation, Radiodiffusion GO, Radio-identification
MF (*medium frequency*)	MF (moyenne fréquence)	300 kHz à 3 MHz	1 km à 100 m	petites ondes ou ondes moyennes ou hectométriques	Radioamateur, Radiodiffusion PO, Service maritime, Appareil de recherche de victimes d'avalanche
HF (*high frequency*)	HF (haute fréquence)	3 MHz à 30 MHz	100 m à 10 m	ondes courtes ou décamétriques	Organisations diverses, Militaire, Radiodiffusion OC, Maritime, Aéronautique, Radioamateur, Météo, Radio de catastrophe, etc.
VHF (*very high frequency*)	THF (très haute fréquence)	30 MHz à 300 MHz	10 m à 1 m	ondes ultra-courtes ou métriques	Radiodiffusion FM, Radiodiffusion RNT, Aéronautique, Maritime, Radioamateur, Gendarmerie nationale française, Pompiers, SAMU, Réseaux privés, taxis, militaire, Météo, etc.
UHF (*ultra high frequency*)	UHF (ultra haute fréquence)	300 MHz à 3 GHz	1 m à 10 cm	ondes décimétriques	Réseaux privés, militaire, GSM, GPS, téléphones sans fil (DECT), Téléphonie mobile, Wi-Fi, Télévision, Radioamateur, etc.
SHF (*super high frequency*)	SHF (super haute fréquence)	3 GHz à 30 GHz	10 cm à 1 cm	ondes centimétriques	Réseaux privés, Wi-Fi, Téléphonie mobile, Micro-onde, Radiodiffusion par satellite (TV), Faisceau hertzien, Radar météorologique, Radioamateur, etc.
EHF (*extremely high frequency*)	EHF (extrêmement haute fréquence)	30 GHz à 300 GHz	1 cm à 1 mm	ondes millimétriques	Réseaux privés, Téléphonie mobile, Radars anticollision pour automobiles, Liaisons vidéo transportables, Faisceau hertzien, Radioamateur, etc.
Térahertz	Térahertz	300 GHz à 3 000 GHz	1 mm à 100 µm	ondes submillimétriques	scanner corporel

Le wifi utilise deux bandes principale (classement de fréquence)

- La première est la bande en 2,4GHz, le classement de cette bande est de 2,4000GHz à 2,4835GHz

- Il y a aussi la bande de 5GHz, le classement de cette bande est de 5,5150GHz à 5,825GHz.

Cette bande est divisé en 4 petites bandes allant de :

5,150GHz à 5,250GHz

5,250GHz à 5,350GHz

5,470GHz à 5,725GHz

5,725GHz à 5,825GHz

La bande de 2,4GHz fournit un étendue plus élevé dans les grands espaces et une meilleure pénétration des obstacles comme les murs.

Cependant il y a plus d'appareils à utiliser les bandes de 2,4GHz donc les interférences peuvent être un plus grand problème comparé aux bandes de 5GHz.

Le Wifi 6 (802.11ax) à étendu le classement des ondes pour inclure une bande dans les 6GHz.

Chaque bande est divisé en plusieurs chaînes et les appareils sont configurés pour transmettre et recevoir le trafique sur une (ou plus) de ces chaînes.

Les bandes de 2,4GHz sont divisés en plusieurs chaînes, chacune d'un classement de 22MHz.

Voici une capture d'écran des chaînes utilisés en fonction des pays :

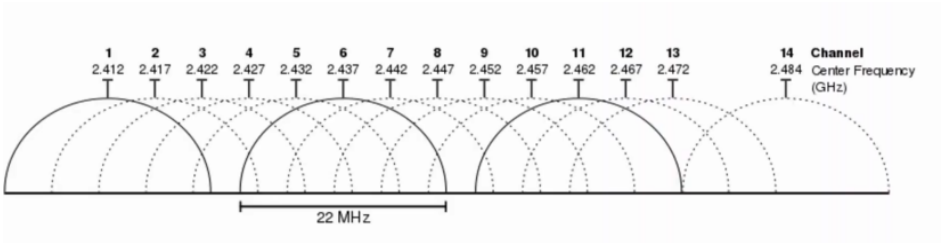
Canal	Fréquence (GHz)	Pays	Note
1	2,412	Japon, Europe ETSI, États-Unis FCC	
2	2,417	Japon, Europe ETSI, États-Unis FCC	
3	2,422	Japon, Europe ETSI, États-Unis FCC	
4	2,427	Japon, Europe ETSI, États-Unis FCC	
5	2,432	Japon, Europe ETSI, États-Unis FCC	
6	2,437	Japon, Europe ETSI, États-Unis FCC	
7	2,442	Japon, Europe ETSI, États-Unis FCC	
8	2,447	Japon, Europe ETSI, États-Unis FCC	
9	2,452	Japon, Europe ETSI, États-Unis FCC	
10	2,457	Japon, Europe ETSI, États-Unis FCC	Ancien plan de bande en France et en Espagne
11	2,462	Japon, Europe ETSI, États-Unis FCC	Ancien plan de bande en France et en Espagne
12	2,467	Japon, Europe ETSI	Ancien plan de bande en France
13	2,472	Japon, Europe ETSI	Ancien plan de bande en France
14	2,484	Japon	

La chaîne 14 est uniquement utilisé au Japon.

Il faut faire attention auxquelles chaînes utiliser pour éviter les interférences.

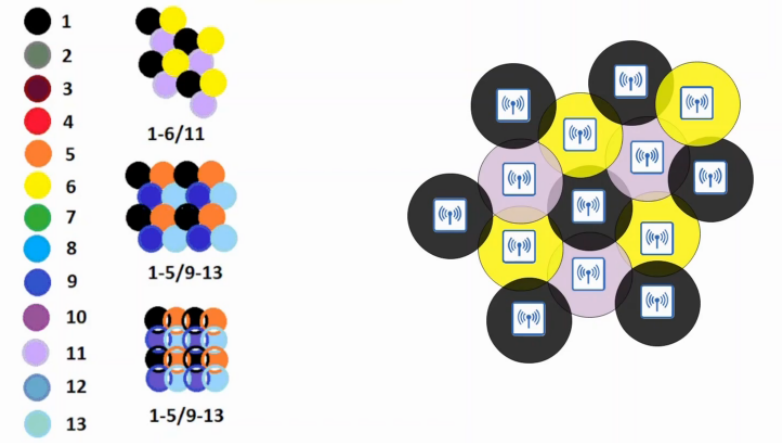
Dans un petit réseau sans fil LAN avec seulement un seule point d'accès on peut utiliser n'importe quelle chaîne. Cependant dans de grandes WLANs avec plusieurs points d'accès, il est important que les points d'accès adjacents n'utilisent pas et ne passe pas sur les autres chaînes. Cela aide à éviter les interférences.

Dans une bande en 2,4GHz, il est recommandé d'utiliser les chaînes : 1, 6 et 11.



On remarque sur le diagramme ci dessus que les trois chaînes 1, 6 et 11 n'interfèrent pas l'une à l'autre.

Sur les bandes de 5GHz consistent à ne pas passer sur les chaînes, donc il est plus facile d'éviter les interférences entre les points d'accès adjacent.



Il est possible de placer les point d'accès de sorte qu'ils n'interfèrent pas entre eux.

Voici les noms des principales fréquences utilisés :

Standard	Frequencies	Max Data Rate (theoretical)	Alternate Name
802.11	2.4 GHz	2 Mbps	
802.11b	2.4 GHz	11 Mbps	
802.11a	5 GHz	54 Mbps	
802.11g	2.4 GHz	54 Mbps	
802.11n	2.4 / 5 GHz	600 Mbps	'Wi-Fi 4'
802.11ac	5 GHz	6.93 Gbps	'Wi-Fi 5'
802.11ax	2.4 / 5 / 6 GHz	4*802.11ac	Wi-Fi 6'

Voyons à présent un dernier point à propos des ensembles de services qui sont les groupes d'appareils de réseau sans fil.

Il y a trois types d'ensemble de services :

→ Independant

→ Infrastructure

→ Mesh

Tous les appareils dans un ensemble de service partagent le même SSID (service set identifier). Le SSID est un nom lisible par l'humain qui identifie l'ensemble du service.

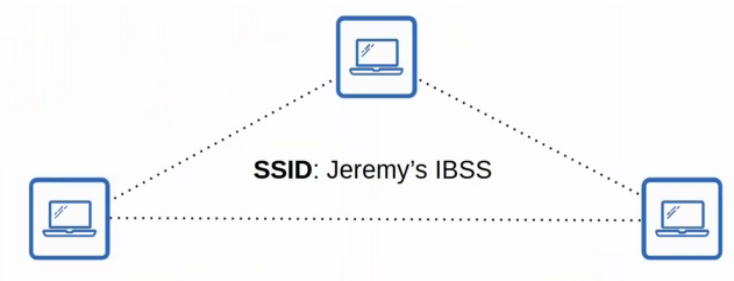
Le SSID n'a pas à être unique.

Un IBSS (Independant Basic Service Set) est un réseau sans fil dans un réseau dans lequel deux ou plus d'appareils sans fil se connectent directement sans utiliser de point d'accès.

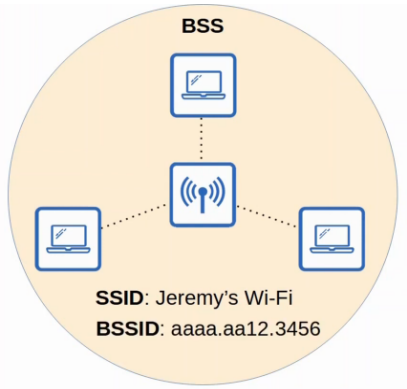
Il est aussi appelé un réseau ad hoc

Il peut être utilisé pour le transfert de fichier (Par exemple par Airdrop)

Il n'est pas évolutif pour quelques appareils.



Un BSS (Basic Service Set) est un type d'Infrastructure Service Set dans lequel les clients se connectent à chacun par un point d'accès, mais pas directement l'un à l'autre.



Le BSSID (Basic Service Set ID) est utilisé pour uniquement identifier le point d'accès.

D'autres points d'accès peuvent utiliser le même SSID mais pas le même BSSID.

Le BSSID est l'adresse MAC de la radio du point d'accès.

Les appareils sans fil font la requête de s'associer avec le BSS.

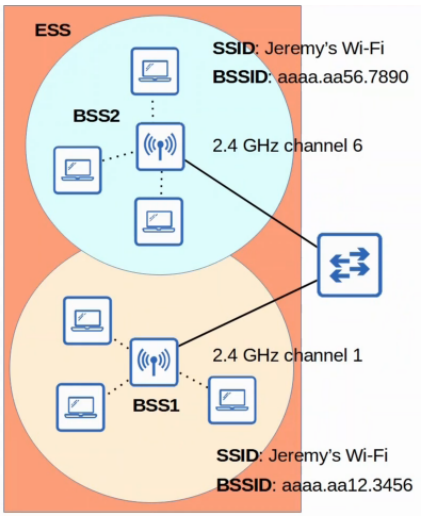
Les appareils sans fil qui se sont associés avec le BSS sont appelés les « clients » ou « stations ».

Un autre terme important est le BSA (Basic Service Area) qui permet d'identifier la zone ou le signal est utilisable.

Dans un BSS les appareils sont directement connectés par une IP, le BSA se réfère seulement à la zone d'utilisation pour joindre un appareil au BSS.

Dans ce type d'infrastructure les clients devraient communiquer par le point d'accès et non pas directement entre eux.

Pour créer un large accès sans fil pour agrandir l'étendu d'un seule point d'accès, on utilise un ESS (Extended Service Set)



Les points d'accès avec leurs propre BSS sont connectés avec un réseau câblé.

Chaque BSS utilise le même SSID

Chaque BSS possède son propre BSSID

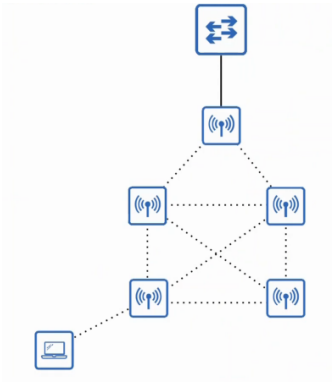
Chaque BSS utilise différentes chaînes pour éviter les interférences.

Les clients peuvent passer entre les points d'accès sans avoir à se reconnecter, ce qui évite les changements de Wifi lorsque l'on bouge entre les points d'accès.

Ceci est appelé le « roaming » ou itinérance.

Les BSA devrait se chevaucher d'à peut près 10 à 15 %

Le dernier type de Service Set que nous étudierons est le MBSS (Mesh Basic Service Set) qui peut être utilisé dans des situations ou il est difficile de lancer une connexion Ethernet à tous les points d'accès.



Les points d'accès Mesh utilisent deux radios : un pour fournir un BSS au point d'accès du client et un autre pour former une liaison terrestre qui est utilisé pour partager en pont le trafique d'un point d'accès vers un autre point d'accès.

Il n'y a qu'un point d'accès qui est connecté au réseau câblé il est appelé le RAP (Root Access Point) Dans le réseau précédent le RAP est celui connecté au Switch.

Les autres points d'accès sont appelés des MAP (Mesh Access Point)

Un protocole est utilisé pour déterminer le meilleur chemin à emprunter (de la même manière que le routage dynamic est utilisé pour déterminer le meilleur chemin vers une destination).

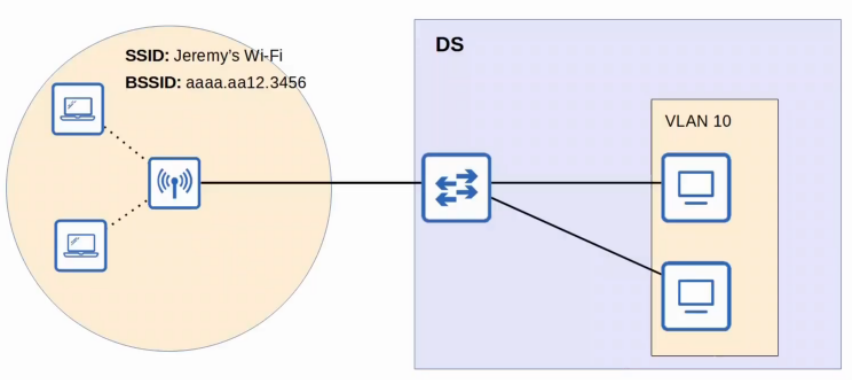
La plupart des réseau sans fil ne sont pas seules.

Il y a une manière pour les clients sans fil de connecter à un appareil câblé à une infrastructure réseau.

Dans le 802.11 le réseau câblé est appelé le DS (Distribution System)

Chaque BSS sans fil ou ESS est cartographié vers le VLAN du réseau câblé.

Voici un exemple :

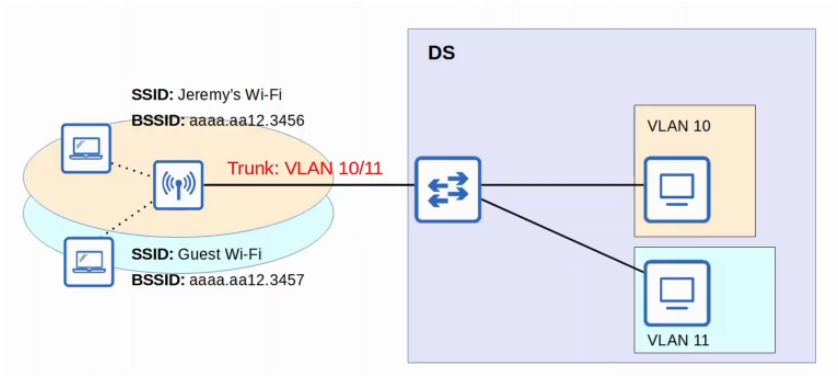


Il est possible pour un point d'accès de fournir plusieurs LAN sans fil chacune avec un SSID unique.

Chaque WLAN est cartographié à une VLAN séparé et connecté à un réseau câblé par un Trunk.

Chaque WLAN utilise un BSSID unique et incrémente le dernier chiffre du BSSID par un.

Voici un exemple :



Un point d'accès peut fonctionner avec des modes additionnels.

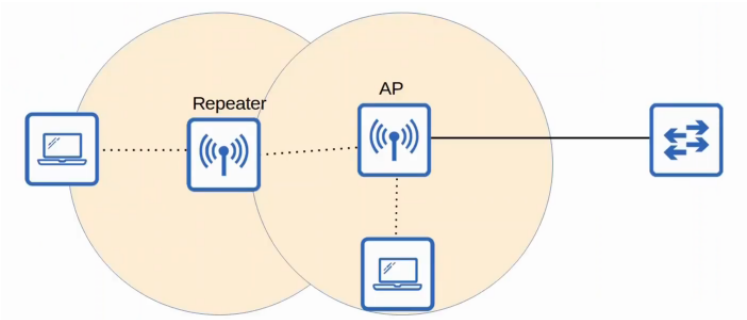
Un point d'accès en mode répéteur peut être utilisé pour étendre la réception du BSS.

Le répéteur va simplement retransmettre n'importe quelle signal reçu du point d'accès.

Un répéteur avec une seule radio va fonctionner sur la même chaîne que le point d'accès. Mais cela peut drastiquement réduire le débit de la chaîne de 50 %

Un répéteur avec deux radios peut recevoir sur une chaîne et puis retransmettre dur une autre chaîne.

Voici un exemple :



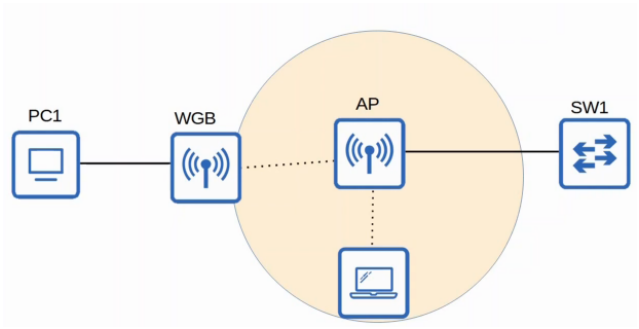
Un Workgroup bridge (WGB) fonctionne comme un client sans fil d'un autre point d'accès, et peut être utilisé pour connecter des appareil filaire à un réseau sans fil.

Dans l'exemple ci dessous, le PC1 ne peut pas avoir de capacités sans fil et aussi n'a pas accès à des connexion câblé vers le SW1

Cependant le PC1 à une connexion câblé vers le WGB qui a une connexion sans fil vers le point d'accès.

Il existe deux type de WGBs :

Universal WGB (uWGB) qui est un standard 802.11 qui permet à un appareil d'être en pont avec les appareils sans fil.



Le WGB est une version propriétaire Cisco du standard 802.11 qui permet plusieurs appareil câblé d'être en pont avec le réseau sans fil.

Un Outdoor Bridge peut être utilisé pour connecter un réseau sur de longues distances sans utiliser de câble physique les connectant.
Les points d'accès utilisent des antennes spécialisés qui concentrent le signal sur une direction, qui permettent au connexion sans fil de se faire sur de longues distances.

La connexion peut être de point à point comme dans le diagramme ci dessous ou en point à multipoint dans lequel plusieurs sites se connectent à un seule site central.

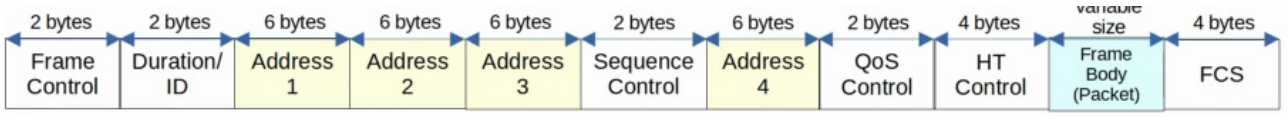
Cours 56 : Architectures Sans Fil

Dans ce cours nous verrons tout d'abord le fonctionnement des messages 802.11 et le format des trames. Le Standard 802.11 pour le Wireless LANs fonctionne différemment du Standard 802.3 Wired Ethernet LANs, donc le type de messages ainsi que les trames sont aussi différentes. Nous verrons ensuite différentes Architectures de points d'accès sans fil : Autonomous AP, Lightweight AP, Cloud-based AP.

En dernier temps nous verrons le déploiement d'un Wireless LAN Controller (WLC)

Le Wireless LAN Controller est utilisé pour centraliser afin de gérer et contrôler les Point d'Accès (« AP » pour Access Point). Ils sont important dans de grands réseaux et peuvent avoir des centaines voir des milliers de point d'accès sans fil.

Voyons de quoi est composé une trame du Standard 802.11 :



Comme on peut le voir les trames 802.11 ont un format différent des trames Ethernet 802.3, les trames 802.11 sont plus compliqués que les trames Ethernet.

Cela peut dépendre de la version du 802.11 et du type de message mais certaines partie peuvent ne pas être présentes dans la trame.

Par exemple il y a dans la trame 4 parties différentes pour les adresses mais pas tous les messages utilisent 4 parties d'adresses différentes.

Voici en un peu plus détaillé chaque partie de la trame :

- Le Frame Control fournit des informations comme le type et sous-type de message.

- Le Duration/ID dépend du type de message, cette partie peut indiquer : le temps (en microsecondes) la chaîne dédié pour la transmission de la trame. Il peut aussi indique l'identifiant pour la connexion.

- Adresses : Jusque 4 adresses peuvent être présentes dans la trame 802.11. L'adresse présente et leur ordre dépend du type de message transmis. L'adresse peut indiquer l'adresse de destination (« DA » pour Destination Address) la réception final de la trame.

L'adresse Source (« SA » pour Source Address) l'envoyeur original de la trame.

L'adresse de réception (« RA » pour Receiver Address) le récepteur immédiat de la trame.

L'adresse de Transmission (« TA » pour Transmetter Address) l'envoyeur immédiat de la trame.

Avoir 4 adresses comme celles ci n'est pas nécessaire pour un réseau Ethernet, mais le Standard 802.11 avec les réseaux sans fil à des conditions requises spécifiques.

- Le Sequence Control est utilisé pour réassembler les fragments et éliminer les trames dupliqués.

- Le QoS Control est utilisé dans le Quality of Service pour prioriser certains trafiques.

- Le HT (High Throughput) Control est ajouté dans le 802.11n pour activer les « opération haut débit ».

802.11n est aussi connu comme le wifi Haut Débit (High Throughput)

802.11ac est aussi connu sous le nom de Wifi Thrès Haut Débit (Very HT)

- Le Frame Body est la trame ou est encapsulé le paquet transmis

- FCS (Frame Check Sequence) a la même fonction que la trame Ethernet et est utilisé pour vérifier les erreurs de la trame.

Voyons la procédure d'association 802.11, il y a un trafique des points d'accès par pont entre la station sans fil et les autres appareils. Pour une station pour envoyer le trafique par ce point d'accès, il faut que l'appareil soit associé avec le point d'accès.

Il y a 3 états de connexion possible avec 802.11 :

- Lorsque l'appareil n'est pas authentifié ou associé avec le point d'accès

- Lorsque l'appareil est authentifié mais pas associé

- Lorsque l'appareil est authentifié et associé avec le point d'accès

La station doit être authentifié et associé avec le point d'accès pour envoyer le trafique.

La station envoie d'abord un message de sonde pour savoir quelles points d'accès et BSS sont disponibles et le point d'accès envoie une réponse sonde pour indiquer qu'il est disponible.

Il y a deux manières pour la station de scanner pour un BSS :

- Le scan actif : la station envoie des requêtes probe et écoute pour une réponse probe depuis un point d'accès.

- Le scan passif : la station écoute pour des messages de balise depuis un point d'accès.

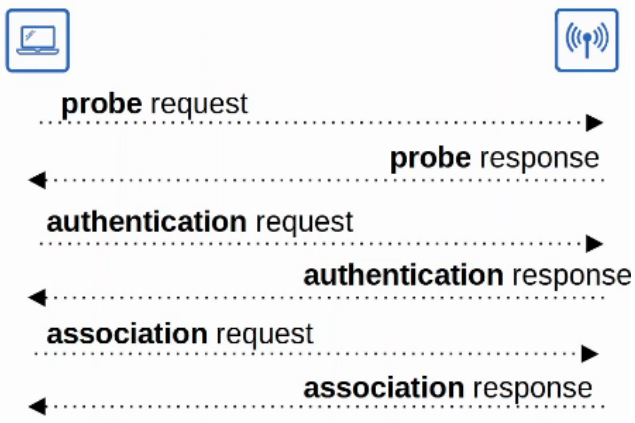
Les messages de balises sont envoyés périodiquement par les point d'accès pour avertir le BSS.

Il y a ensuite l'authentification qui se fait avec la requête de l'appareil, par exemple la station envoie un mot de passe au point d'accès et le point d'accès authentifie l'appareil.

Si cela marche l'appareil est authentifié mais pas encore associé.

Une fois l'authentification faite il y a une requête pour l'association et une réponse.

Si cela a marché l'appareil est authentifié et associé. La station et l'appareil peuvent communiquer.



Il y a trois types de messages 802.11 :

- Trame de gestion : utilisés pour gérer le BSS, par exemple les balises, les requêtes de sonde, les réponses de sonde, l'authentification, les requêtes d'association et les réponses d'association.

- Contrôle : est utilisé pour contrôler l'accès vers le support (Par fréquence radio). Assiste avec distribution de gestion des données de trame.

Par exemple les messages : RTS (Request to Send), CTS (Clear to Send) et ACK

- Les données : sont utilisés pour envoyer des paquets de données actuel.

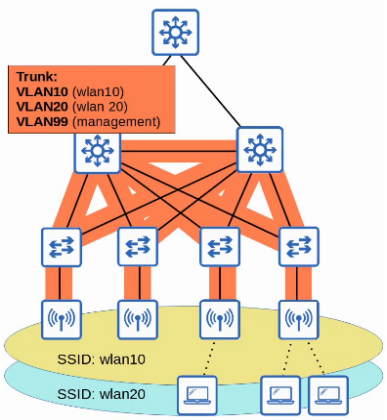
Il y a trois méthodes principales dans le déploiement de point d'accès sans fil :

1. Autonomous AP : les point d'accès autonome contiennent eux même le système ils ne s'appuient pas sur un WLC (Wireless LAN Controler), ils sont autonome et sont configurés individuellement, ils peuvent être configurés par un câble console (CLI), telnet/SSH (CLI), ou HTTP/HTTPS connexion web (GUI) une adresse IP pour la gestion distante doit être configuré. Les paramètres de radio fréquence doivent être configurés manuellement (transmettre l'énergie, la chaîne, etc.) Les politiques de sécurité sont traités individuellement par chaque point d'accès.

Les règles QoS, etc.. sont configurés individuellement sur chaque point d'accès.

Il n'y a pas de gestion central ou de gestion des points d'accès.

Voici un exemple d'un point d'accès autonome :



Les points d'accès autonome se connectent au réseau câblé avec un lien Trunk.

Le trafique de donnée depuis les clients sans fil ont un chemin direct vers le réseau câblé ou vers d'autres clients sans fil associés avec le même point d'accès.

Chaque VLAN doit s'étirer sur tout le réseau. Cela est considéré comme mauvaise pratique car il y aura en conséquence un large domaine de broadcast, mais aussi le spanning Tree va désactiver les liens. Ajouter et supprimer des VLAN est un travail très laborieux.

Les points d'accès autonome peuvent donc être utilisés dans de petits réseaux mais ils ne sont pas viables dans un milieu avec de grands réseaux. Les grands réseaux peuvent être constitués de plusieurs milliers de points d'accès et configurer chacun de ces points d'accès n'est pas réaliste.

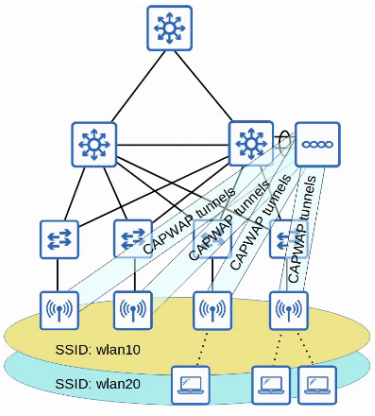
Les points d'accès autonome peuvent fonctionner dans les modes vue auparavant qui sont : Repeater, Outdoor Bridge, Workgroup Bridge.

2. Lightweight AP : Les fonctions d'un point d'accès peuvent être divisés en le point d'accès et le Wireless LAN Controller (WLC). Le Lightweight AP gère en temps réel des opérations comme transmettre/recevoir le trafique par radiofréquence et le trafique de cryptage/décryptage, envoyer des balises et sondes, etc. d'autres fonctions sont possible avec WLC par exemple la gestion des radiofréquence, la gestion de sécurité/QoS, l'authentification client, association client/itinérant, etc.

Cela est appelé l'architecture split-MAC puisque les fonctions sont divisés entre l'AP lightweight et le WLC. Le WLC est aussi utilisé pour configurer de manière centralisé les points d'accès lightweight. Le WLC peut être localisé dans le même sous réseau/VLAN que le point d'accès lightweight qu'il gère, ou bien sur des sous réseau/VLAN différents.

Le WLC et les points d'accès lightweight s'authentifient chacun en utilisant des certificats digitaux installés sur chaque appareils (certificat X.509 standard). Cela assure que seulement les points d'accès autorisés peuvent joindre le réseau.

Voici un exemple d'un point d'accès lightweight :



Le WLC et les points d'accès lightweight utilisent un protocole appelé CAPWAP (Control And Provisioning Of Wireless Access Points) pour communiquer.

Basé sur un ancien protocole appelé LWAPP (Lightweight Access Point Protocol).

Deux tunnels sont crées entre chaque point d'accès et le WLC :

Un Control Tunnel (UDP port 5246). Ce tunnel est utilisé pour configurer le point d'accès, et contrôler/gérer les opérations. Tout le trafique dans ce tunnel est crypté par défaut.

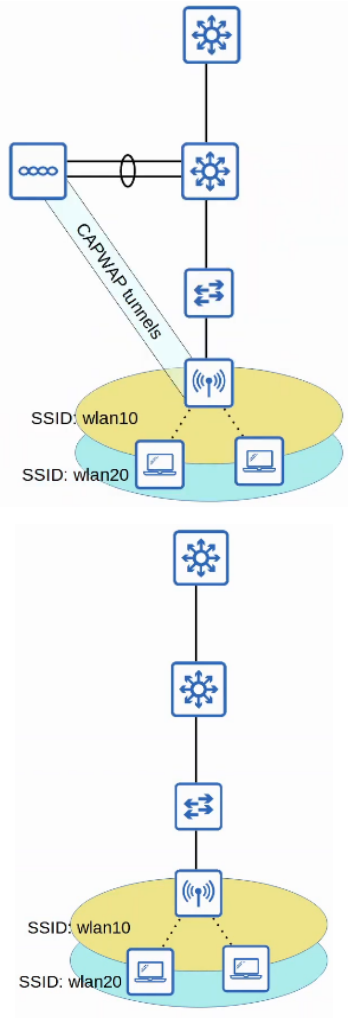
Un Tunnel de données (UDP port 5247). Tout le trafique depuis le client sans fil est envoyé sur ce tunnel vers le WLC. Il ne va pas directement vers le réseau câblé.

Le trafique dans ce tunnel n'est pas crypté par défaut, mais il est possible de le crypter avec DTLS (Datagram Transport Layer Security).

Puisque tout le trafique depuis des clients sans fil est en tunnel vers le WLC avec CAPWAP, les points d'accès se connecte au port d'accès du switch, non pas des ports Trunk.

Voyons comment démontrer cela avec des diagrammes :

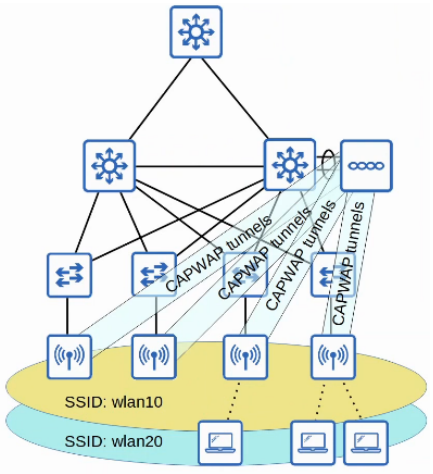
Ci dessous à gauche est présent une architecture avec des points d'accès Lightweight, à droite est présent des points d'accès autonomes :



Lorsque l'on utilise un réseau autonome, un lien trunk est utilisé pour se connecter. Il doit y avoir un VLAN pour chaque SSID que le point d'accès propose.

Avec un point d'accès lightweight le client n'a pas besoin de se connecter avec un lien Trunk, un lien d'accès est suffisant. Un lien Trunk est cependant nécessaire pour connecter le WLC vers le réseau câblé.

Il y a des avantages à utiliser des architectures split MAC :



- Scalability ou évolutivité : avec WLC (ou il est bien plus simple de construire et supporter un réseau avec plusieurs milliers de points d'accès.

- Dynamic Channel Assignment : Le WLC peut automatiquement sélectionner quelle chaîne chaque point d'accès devra utiliser.

- Transmettre l'optimisation de puissance : le WLC peut automatiquement placer la puissance approprié pour chaque point d'accès.

- Re-génération de la couverture sans fil : Lorsque le point d'accès arrête de fonctionne, le WLC peut augmenter la transmission de puissance des points d'accès pour empêcher les espaces non couvert.

- Gestion de Securité/QoS : La gestion central de la sécurité et la politique QoS s'assure de la consistance à travers le réseau.

Les points d'accès lightweight peuvent être configurés pour fonctionner sur des modes variés :

- Local : c'est le mode par défaut ou le point d'accès offre un BSS (plusieurs BSS) au client pour s'associer avec lui.

- FlexConnect : Comme un point d'accès lightweight dans un mode local, il offre un ou plusieurs BSS aux clients pour s'associer avec. Seulement FlexConnect permet au point d'accès de changer localement entre le réseau câblé et sans fil si le tunnel WLC ne fonctionne plus.

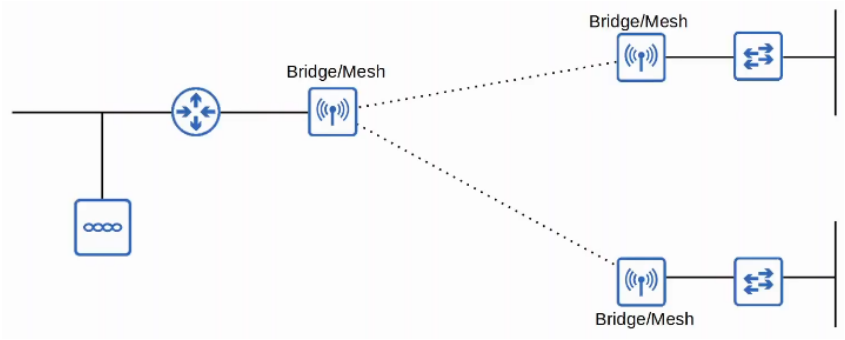
- Sniffer : Le point d'accès n'offre pas un BSS au client. Il est dédié pour capturer les trames 802.11 et les envoyer vers l'appareil qui lance un logiciel comme Wireshark.

- Monitor : Le point d'accès n'offre pas un BSS aux clients. Il est dédié à recevoir une trame 802.11 pour détecter les appareil rogues. Si un client est trouvé et est un appareil rogue, un point d'accès peut désauthentifier les messages pour disocier l'appareil rogue depuis le point d'accès.

- Rogue Detector : Le point d'accès n'utilise pas sa radio. Il écouté le trafique sur le réseau câblé uniquement, mais il reçoit une liste d'appareil suspectés être des clients rogue et les adresse MAC des points d'accès par le WLC. En écoutant les messages ARP sur un réseau câblé et faisant le rapport avec les informations qu'il reçoit du WLC, il peut détecter les appareils rogue.

- SE-Connect (Spectrum Expert Connect) : Le point d'accès n'offre pas un BSS aux clients.il est dédié au spectre d'analyses radiofréquences sur toutes les chaînes. Il peut envoyer des informations au logiciel comme Cisco Spectrum Expert sur un PC pour collecter et analyser les données.

- Bridge/Mesh : Tout comme les point d'accès autonome Outdoor Bridge Mode, les points d'accès lightweight peuvent être un pont dédié entre des sites, même sur une longue distance. Un maillage peut être fait entre les points d'accès. Voici un exemple :



- Flex plus Bridge : ajoute la fonctionnalité FlexConnect au mode Bridge/Mesh. Ce qui permet aux points d'accès sans fil de partager localement le trafique même si la connectivité vers le WLC est perdu.

3. Cloud based : Les architecture basé sur le Cloud est à moitié entre les points d'accès autonome et les architecture split-MAC. Cela implique des points d'accès autonome qui sont géré de manière centralisé dans le Cloud.

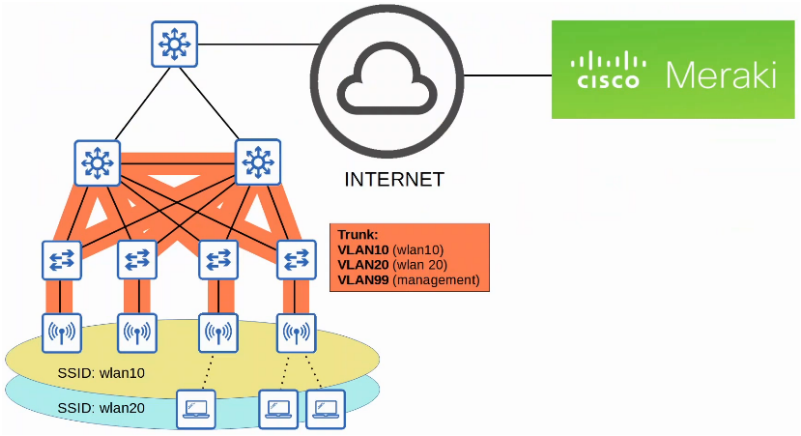
Cisco Meraki est une solution Wifi populaire basé sur le Cloud.

Le dashboard Meraki peut être utilisé pour configurer des points d'accès, gérer le réseau, générer des rapports de performance, etc...

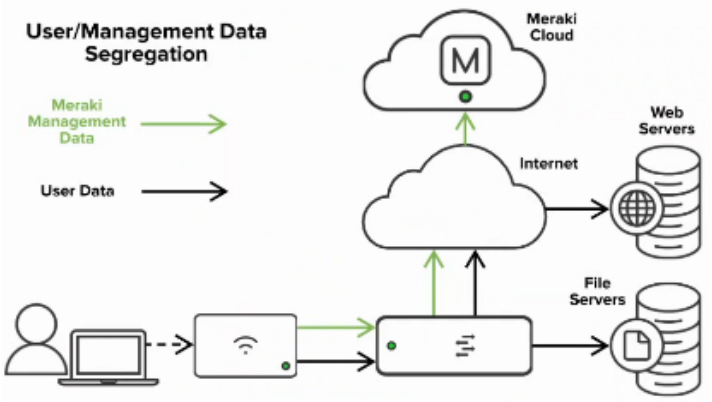
Meraki indique à chacun de ses points d'accès quelle chaîne utiliser, quelle énergie transmettre, etc.

Le trafique de donnée n'est pas envoyé vers le Cloud. Il est envoyé directement vers le réseau câblé comme lorsque des points d'accès autonome sont utilisés.

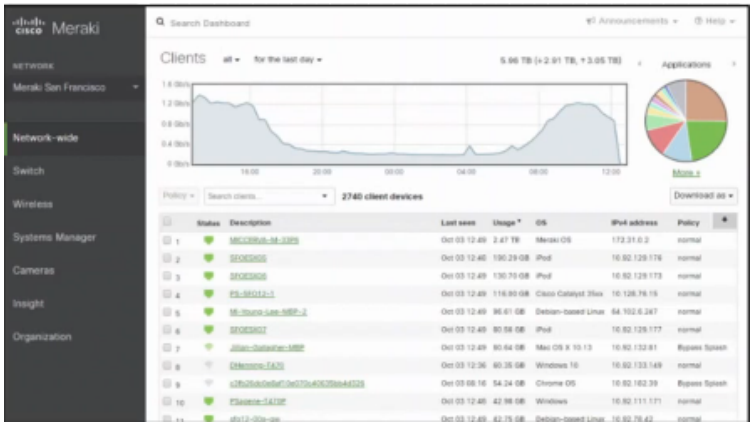
Seulement ce que l'on appelle gestion/contrôle du trafique est envoyé vers le Cloud. Voici un exemple :



Sur le site de Cisco Meraki est donné ce schéma :



Le dashboard Meraki ressemble à cela :



Voyons à présent le déploiement WLC.

Dans une architecture split-MAC, il y a 4 principaux modèles de déploiement :

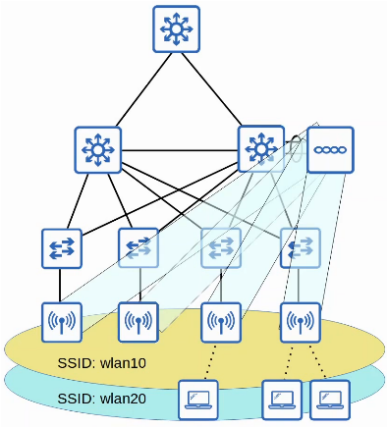
- Unified : Le WLC est un matériel dans une localisation central du réseau.

- Cloud-Based : Le WLC est une VM lancé dans un serveur, de manière normal dans un cloud privée dans un data center. Ce n'est pas le même que l'architecture Cloud Based comme discuté auparavant.

- Embedded : Le WLC est intégré dans le Switch

- Mobility Express : Le WLC est intégré dans le point d'accès.

Voici un exemple d'un WLC unifié :

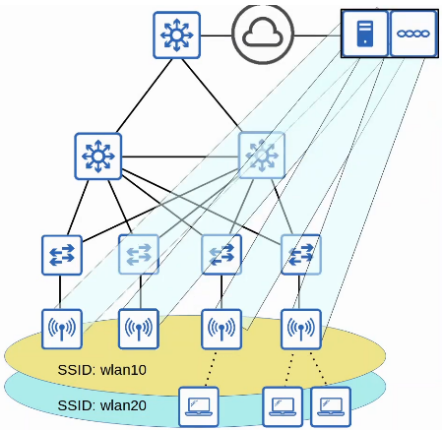


Le WLC est un matériel déployé dans une localisation centralisé du réseau.

Un WLC unifié peut supporter jusqu'à 6000 points d'accès.

S'il y en a plus de 6000, des WLC supplémentaires peuvent être ajoutés au réseau.

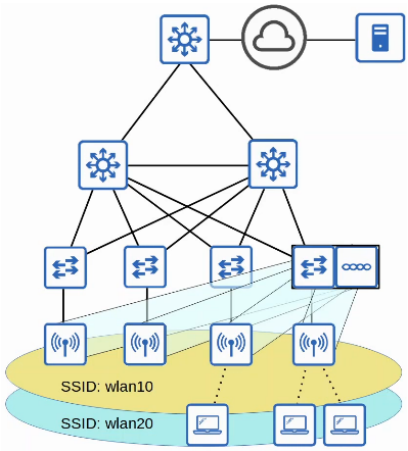
Voici un exemple de WLC Cloud Based :



Le WLC est une VM lancé sur un serveur, de manière normal dans un cloud privée dans un data center. Un WLC cloud-based peut supporter jusqu'à 3000 points d'accès.

Si plus de 3000 points d'accès sont nécessaire, plus de VM WLC peuvent être déployés.

Voici un exemple de Embedded WLC :

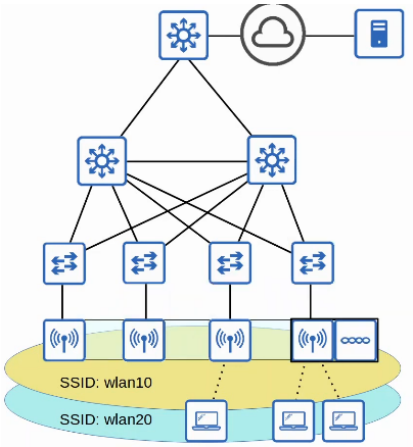


Le WLC est embedded dans le Switch.

Un WLC Embedded peut supporter jusqu'à 200 points d'accès.

Si plus de 200 points d'accès sont nécessaire, plus de Switchs avec un WLC embedded peuvent être ajoutés.

Voici un exemple de Cisco Mobility Express WLC :



Le WLC est embedded dans le point d'accès.

Un Mobility Express WLC peut supporter jusqu'à 100 points d'accès.

Si plus de 100 points d'accès sont nécessaires plus de points d'accès embedded Mobility Express WLC peuvent être ajoutés.

Cours 57 : Sécurité Sans Fil

Dans cette vidéo nous verrons la sécurité sans fil.

Nous verrons de nouveaux concepts en sécurité informatique en donnant d'abord une introduction à la sécurité dans le réseau sans fil puis des différentes méthodes d'authentification. Nous verrons aussi les différentes méthodes de cryptage et de l'intégrité des données puis nous verrons le fonctionnement de WPA (Wifi Protected Access).

La sécurité est importante dans tous les réseaux et même encore plus essentiel dans les réseaux sans fil. La raison principale est que les signaux sans fil ne sont pas contenu dans des câbles, n'importe quelle appareil avec une certaine plage de signal peut recevoir ce trafique.

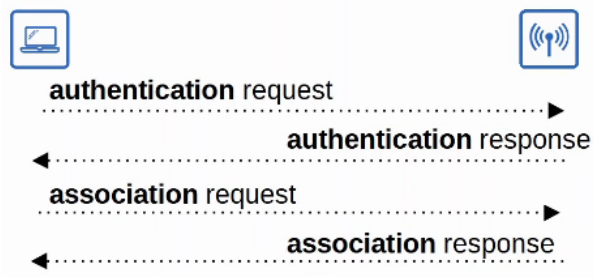
Dans des réseaux câblés, le trafique est souvent seulement crypté lorsqu'il est envoyé à travers un réseau qui n'est pas de confiance comme Internet. Dans les réseaux sans fil il est très important de crypté le trafique envoyé entre des appareils clients sans fil et un point d'accès car n'importe quelle appareil peut le réceptionner.

Nous verrons 3 concepts pour mieux comprendre cela :

- Authentification : Tous les clients doivent être authentifiés avant d'être associé avec le point d'accès. Dans un paramétrage d'entreprise, seulement les appareils et utilisateurs doivent aveoir accès au réseau. Un SSID séparé qui n'a pas accès au réseau de l'entreprise peut aussi être mis en place pour les invités. Ces invités ont moins de restrictions d'accès et ont un accès uniquement à Internet et non pas aux ressources internes de l'entreprise. De manière idéal les clients devraient aussi authentifier le point d'accès afin d'éviter de s'associer avec un point d'accès frauduleux.

Il y a plusieurs manière pour s'authentifier : Mot de passe, NomUtilisateur/MotDePasse, Certificats.

L'authentification se fais comme suit :



- Cryptage : Le trafique envoyé par les clients et les points d'accès doivent être cryptés afin qu'il ne soit pas interceptés par quelqu'un d'autre.

Il y a plusieurs protocoles qui peuvent être utilisés afin de crypter le trafique.

Tous les appareils sur le WLAN utiliseront le même protocole, seulement chaque client utilisera une clé unique de cryptage/décryptage donc les autres appareils ne pourront pas lire son trafique.

Un « groupe de clé » est utilisé par le point d'accès pour crypter le trafique qu'il veut pour l'envoyer à tous ses clients. Tous les clients associés avec le point d'accès garderont cette clé donc ils pourront décrypter le trafique.

- Intégrité : Comme expliqué auparavant, l'intégrité s'assure que le message n'est pas modifié par un tiers partie. Le message qui est envoyé par l'hôte source devrait être le même que le message reçu par l'hôte de destination. Un MIC (Message Integrity Check) est ajouté aux messages pour aider à protéger leurs intégrité.

L'envoyeur calcule le MIC pour le message et l'attache au message, puis il crypte et envoie la trame. Le récepteur reçoit le message et décrypte le message, le récepteur calcul indépendamment un MIC pour le message (en utilisant le même protocole que l'envoyeur).

Si les deux MIC sont les mêmes le récepteur en déduit que le message n'a pas été altéré.

Voyons à présent différentes méthodes d'authentifications qui sont :

- Open Authentification

- WEP (Wired Equivalent Privacy)

- EAP (Extensible Authentication Protocol)

- LEAP (Lightweight EAP)

- EAP-FAST (EAP Flexible Authentication via Secure Tunneling)

- PEAP (Protected EAP)

- EAP-TLS (EAP Transport Layer Security)

Voyons chacune de ces méthodes plus en détail.

Le standard original 802.11 inclus deux options pour l'authentification :

- Open Authentication : Le client envoie une requête d'authentification et le point d'accès l'accepte sans questions. Il ne s'agit pas d'une méthode d'authentification sécurisé. Après que le client est authentifié et associé avec le point d'accès, il est possible que cela requière pour l'utilisateur de s'authentifier par une autre méthode avant que l'accès au réseau soit autorisé. (Par exemple un point d'accès Wifi).

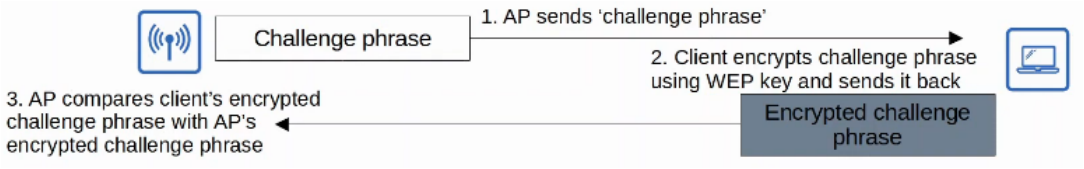
- WEP (Wired Equivalent Privacy) : WEP est utilisé pour fournir les deux authentifications et cryptage du trafique sans fil. Pour le cryptage, WEP utilise l'algorithme RC4.

WEP est un protocol « clef partagé », qui requière à l'envoyeur et récepteur d'avoir la même clef.

Ces clefs WEP peuvent être de 40bits ou 104bits de longueur. Les clefs peuvent être combinés avec un 24-bit 'IV' (Initialization Vector) pour apporter la longueur totale de 64 bits ou 128 bits.

Le cryptage WEP n'est pas sécurisé et peut être facilement cracké.

WEP peut être utilisé pour une authentification comme suit :



Le point d'accès envoie une phrase challenge. Le client crypte la phrase challenge en utilisant une clef WEP et la renvoie. Le point d'accès compare les phrase cryptés avec sa phrase crypté challenge. Si elles correspondent cela signifie que les deux appareils utilisent la même clef donc l'authentification fonctionne.

- EAP (Extensible Authentication Protocol) : EAP est un cadre dans l'authentification. Il définit un standard placé de fonctions d'authentifications qui sont utilisés par des méthodes EAP variés.

Nous verrons ces 4 méthodes EAP : LEAP, EAP-FAST, PEAP et EAP-TLS.

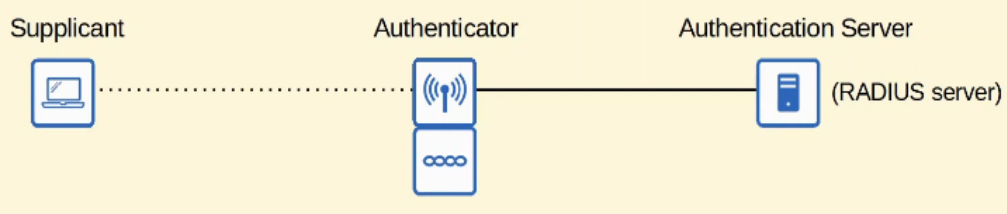
EAP est intégré avec 802.1X qui fournit un contrôle du réseau basé sur les ports.

802.1x est utilisé pour limiter l'accès aux clients connectés à un LAN ou un WLAN jusqu'à se qu'il s'authentifient. Il y a 3 principale entités dans 802.1X :

1. Le demandeur est l'appareil qui veut se connecter au réseau.

2. L'authenticator est l'appareil qui fournit l'accès au réseau.

3. L'authenticator Server (AS) est l'appareil qui reçoit les informations d'identification et permet ou bloque l'accès.



Voyons différentes méthodes d'authentifications EAP utilisés dans des LAN sans fil.

- LEAP (Lightweight EAP) : a été développé par Cisco pour une amélioration de WEP.

Les clients doivent fournir un nom d'utilisateur et un mot de passe pour s'authentifier.

En addition, une authentification mutuel est fournit par le client et le serveur qui envoie une phrase challenge entre chacun tout comme WEP.

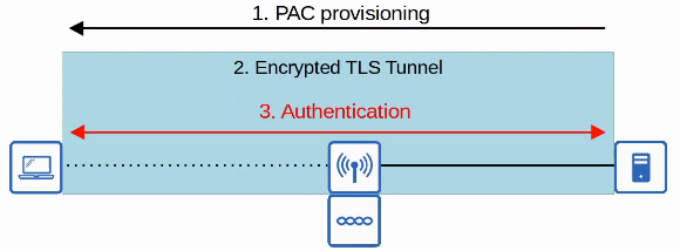
Pour améliorer la sécurité des clefs Dynamic WEP sont utilisés, signifiant que les clés sont changés fréquemment. Tout comme WEP, LEAP est considéré comme vulnérable et ne devrait plus être utilisé.

- EAP FAST (EAP Flexible Authentication via Secure Tunneling) : EAP FAST a aussi été conçu par Cisco. Cette méthode consiste en 3 phases :

1. Un PAC (Protected Access Credential) est généré et passé depuis le serveur vers le client.

2. un Tunnel sécurisé TLS est établi entre le client et le serveur d'authentification.

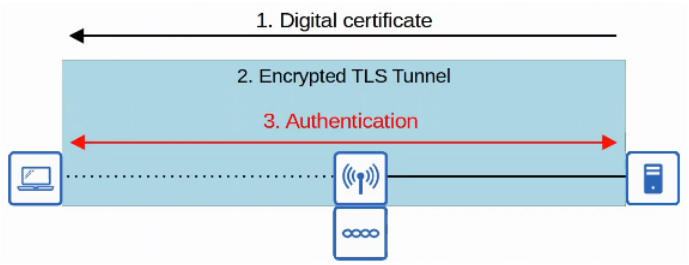
3. à l'intérieur du tunnel sécurisé (crypté), le client et le serveur communiquent pour authentifier/autoriser le client.



- PEAP (Protected EAP) : Tout comme EAP-FAST, PEAP implique d'établir un tunnel TLS sécurisé entre le client et le serveur. Au lieu d'utiliser PAC, le serveur a un certificat digital.

Il montre son certificat au client, le client utilise le certificat digital pour authentifier le serveur.

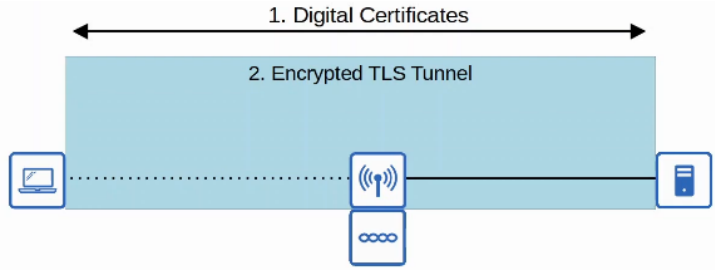
Puisque seulement le serveur fournit un certificat pour l'authentification, le client doit rester authentifié dans le tunnel sécurisé, par exemple en utilisant MS-CHAP (Microsoft Challenge-Handshake Authentication Protocol)



- EAP-TLS (EAP Transport Layer Security) : PEAP ne requière que le AS (Auithentification Serveur) pour avoir un certificat, EAP-TLS requière un certificat sur le AS et sur chacun des clients.

EAP-TLS est la méthode d'authentification sans fil la plus sécurisé, mais il est plus difficile de l'intégrer par rapport à PEAP car chaque appareil client a besoin d'un certificat.

Puisque le client et le serveur s'authentifient chacun avec un certificat digital il n'y a pas besoin pour authentifier le client dans un tunnel TLS.



Nous allons à présent voir les méthodes de cryptage et d'intégrité qui sont :

- TKIP (Temporal Key Integrity Protocol)

- CCMP (Counter/CBC-MAC Protocol)

- GCMP (Galois/Counter Mode Protocol)

Nous aurions pu ajouter WEP mais il a déjà été décrit auparavant.

Voyons chacune de ces méthodes plus en détail.

- TKIP (Temporal Key Integrity Protocol) : WEP a été trouvé trop vulnérable, mais le matériel sans fil de ce temps étaient conçus pour utiliser WEP. Une solution temporaire a été nécessaire jusqu'à ce qu'un nouveau standard soit crée et du nouveau matériel crée.

TKIP ajoute une grande variété de fonctionnalités de sécurité, par exemple :

Un MIC est ajouté pour protéger l'intégrité des messages. Un Algorithme de mélange des clef est utilisé pour créer une clef WEP unique pour toutes les trames.

Le vecteur d'initialisation est doublé en longueur de 24bits à 48bits, rendant l'attaque par brute force beaucoup plus difficile. Le MIC inclut l'adresse MAC de l'envoyeur pour identifier la trame de l'envoyeur. Un timestamp est ajouté au MIC pour éviter les attaques Replay (Attaque par réinsertion en Français). Les attaques Replay impliquent de ré-envoyer une trame qui a déjà été transmise. De plus une séquence de nombre TKIP est utilisé pour garder une trace de la trame envoyé depuis chaque adresse MAC source. Cela protège aussi contre les attaque Replay.

- CCMP (Counter/CBC-MAC Protocol) : CCMP a été développé après TKIP et est plus sécurisé.

Il est utilisé dans WPA2. Pour utiliser CCMP il faut qu'il soit supporté par le matériel de l'appareil. De vieux matériaux construit uniquement pour utiliser WEP/TKIP ne peuvent pas utiliser CCMP.

CCMP consiste dans l'utilisation de deux différents algorithmes pour fournir le cryptage et MIC.

1. AES (Advanced Encryption Standard) counter mode encryption.

AES est le protocole de cryptage le plus sécurisé actuellement utilisé. Il est largement utilisé dans le monde. Il y a plusieurs modes pour l'opération de AES. CCMP utilise le « counter mode ».

2. CBC-MAC (Cipher Block Chaining Message Authentication Code) est utilisé comme MIC pour s'assurer de l'intégrité des messages.

- GCMP (Galoisɯ/Counter Mode Protocol) : GCMP est plus sécurisé et efficace par rapport à CCMP. Il a augmenté son efficacité qui permet un haut transfert des données par rapport à CCMP.

Il est utilisé dans WPA3. GCMP consiste lui en deux algorithmes :

1. Cryptage AES counter mode

2. GMAC (Galois Message Authentication Code) est utilisé comme MIC pour s'assurer de l'intégrité des messages.

Voyons le fonctionnement de WPA.

L'alliance Wi-Fi a développé 3 certifications WAP pour les appareils sans fil :

WPA, WPA2, WPA3

Pour être certifié WPA, l'équipement doit être testé dans le lab de test autorisé.

Toutes ces certifications supportent deux modes d'authentification :

- Mode Personnel : Un pre-shared key (PSK) ou clé pré-partagé est utilisé pour l'authentification. Lorsque l'on se connecte à un réseau de maison Wi-Fi, on entre le mot de passe pour s'authentifier, cela est le mode personnel. C'est le plus commun dans de petits réseaux. Le PSK n'est pas envoyé dans les airs. Un four-Way handshake est utilisé pour l'authentification, et le PSK est utilisé pour générer une clé de cryptage.

- Le mode Entreprise : 802.1X est utilisé avec un serveur d'authentification (Serveur RADIUS).

Une méthode non spécifique EAP est spécifié, donc tous les autres méthodes d'authentifications sont supportés (PEAP, EAP-TLS, etc..)

La certification WPA a été développé après qu'il a été prouvé que WEP soit vulnérable et inclut les protocoles suivants :

- TKIP (Basé sur WEP) fournit le cryptage/MIC.

- authentification 802.1X (Mode Entreprise) ou PSK (Mode Personnel)

WPA2 a été publié en 2004 et inclut les protocoles suivant :

- CCMP qui fournit le cryptage/MIC

- authentification 802.1X (Mode Entreprise) ou PSK (Mode Personnel)

WPA3 a été publié en 2018 et inclut les protocoles suivant :

- GCMP fournit le cryptage/MIC

- authentification 802.1X (Mode Entreprise) ou PSK (Mode Personnel)

- WPA3 fournit aussi différentes fonctionnalités additionnel de sécurité comme par exemple :

- PMF (Protected Management Frames) qui protège la gestion des trames 802.11 de l'espionnage.

- SAE (Simultaneous Authentication of Equals) protège le four-way handshake lorsqu'il utilise le mode personnel d'authentification.

- Le partage de secret empêche les données d'être décryptés après avoir été transmises dans les airs. Donc un attaquant ne peut pas capturer la trame sans fil et donc essaie de les décrypter plus tard.

Cours 58 : Configuration Sans Fil

Dans ce cours nous verrons la configuration sans fil.

Nous ferons d'abord une introduction sur les topologies réseaux que nous utiliserons.

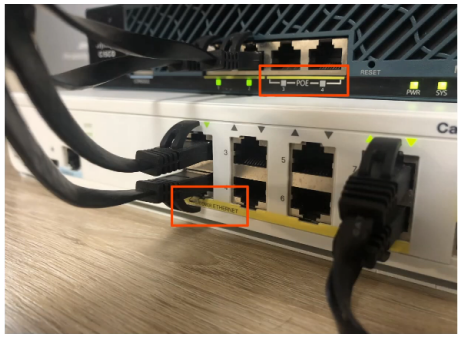
Nous ferons aussi une introduction sur la configuration nécessaire sur un Switch avant d'y connecter tous les appareils à ce Switch central.

Nous verrons la mise en place de base des contrôleurs sans fil LAN nous pourrons accéder au GUI (Graphical User Interface) et faire la configuration. Nous verrons ensuite comment configurer les interfaces WLC, et nous configurerons quelques WLAN. En dernier temps nous verrons quelques fonctionnalités additionnels de WLC.

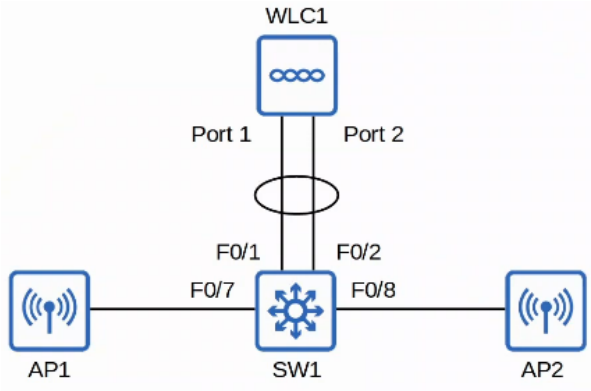
La topologie du réseau que nous utiliserons sera la suivante :

Sur cette topologie on peut voir 1 Switch, UN WLC controler et 2 point d'accès (AP pou access Point). On peut voir que les points d'accès ne sont pas alimentés par des alimentations, elles sont alimentés par les câbles Ethernet qui fournissent l'énergie en PoE.

Comme on peut le voir sur l'image suivante le Switch supporte le PoE, tout comme le WLC (en haut)



La topologie du réseau est le suivant :



Le WLC est connecté au Switch par un LAG (Link Aggregation Group) ou EtherChannel.

Les WLC supportent uniquement le LAG Statique et non pas PagP ou LACP.

Nous utiliserons dans ce réseau 3 VLAN :

VLAN 10 : Management, 192.168.1.0/24

VLAN 100 : Internal, SSID : Internal, 10.0.0.0/24

VLAN 200 : Guest, SSID : Guest, 10.1.0.0/24

Le Vlan 10 est utile seulement dans la gestion des appareils, modifier leurs configurations etc..

Les VLAN 100 et 200 seront utiles dans l'usage des utilisateurs.

Le switch aura un SVI pour chaque VLAN à chaque fois l'adresse finissant par .1 de chaque sous réseau. Le WLC aura lui aussi une adresse IP de chaque VLAN aussi avec pour adresse finissant par .100 dans chaque sous réseau.

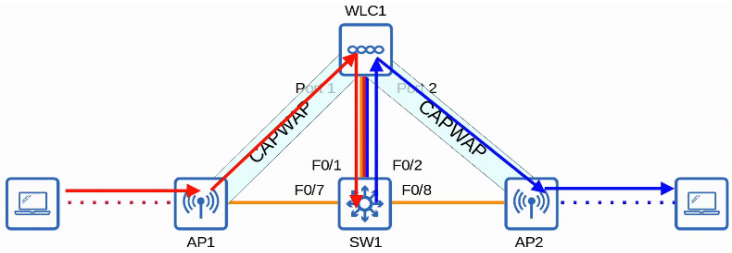
Le Switch a été configuré pour être à la fois le serveur DHCP et le serveur NTP.

Seulement le lien avec le WLC doit être configuré avec le Switch pour être un port Trunk.

Par exemple lorsqu'un client se connecte à l'uns des points d'accès, les données sont transmises avec CAPWAP vers le WLC qui lui même fais le transfert vers le Switch. Le Switch transfère ensuite ses données au client en passant par le même chemin, c'est à dire le WLC puis le point d'accès.

Maintenant que se passe t-il si le client du VLAN 100 (Interne) veut communiquer avec un autre client du VLAN 200 (Guest) ? Le client enverra le trafique vers son Gateway, qui transmettra au WLC puis au Switch. Le Switch va redistribuer le trafique vers le WLC qui le transmettra au Switch puis au point d'accès afin de le faire passer vers le client de l'autre VLAN.

Comme sur le schéma suivant :



Commençons par faire la configuration du Switch, on utilise les commandes suivantes :

```
SW1(config)#vlan 10
SW1(config-vlan)#name Management
SW1(config-vlan)#vlan 100
SW1(config-vlan)#name Internal
SW1(config-vlan)#vlan 200
SW1(config-vlan)#name Guest

SW1(config)#int range f0/6 - 8
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 10
SW1(config-if-range)#spanning-tree portfast

SW1(config-if-range)#interface range f0/1 - 2
SW1(config-if-range)#channel-group 1 mode on

SW1(config-if-range)#interface port-channel 1
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 10,100,200
```

On commence par créer les 3 VLANs, et leur donner un nom avec les commandes suivantes :

```
SW1(config)#vlan 10
SW1(config-vlan)#name Management
SW1(config-vlan)#vlan 100
SW1(config-vlan)#name Internal
SW1(config-vlan)#vlan 200
SW1(config-vlan)#name Guest
```

On configure ensuite les interface pour spécifier le « mode access » :

```
SW1(config)#int range f0/6 -- 8
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 10
SW1(config-if-range)#spanning-tree portfast
```

On configure les interfaces vers le WLC en Etherchannel (LAG) avec les commandes suivantes :

WLC supporte uniquement le LAG statique et non pas PagP ou LACP.

```
SW1(config-if-range)#interface range f0/1 -- 2
SW1(config-if-range)#channel-group 1 mode on
```

On configure les interface port channel en mode Trunk avec les commandes :

```
SW1(config-if-range)#interface port-channel 1
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 10,100,200
```

Une fois cela configuré on lance ensuite les commandes suivantes pour la configuration des SVI (Switch Virtual Interface) des VLAN, du serveur DHCP et de NTP :

```
SW1(config)#interface vlan 10
SW1(config-if)#ip address 192.168.1.1 255.255.255.0
SW1(config-if)#interface vlan 100
SW1(config-if)#ip address 10.0.0.1 255.255.255.0
SW1(config-if)#interface vlan 200
SW1(config-if)#ip address 10.1.0.1 255.255.255.0

SW1(config)#ip dhcp pool VLAN10
SW1(dhcp-config)#network 192.168.1.0 255.255.255.0
SW1(dhcp-config)#default-router 192.168.1.1
SW1(dhcp-config)#option 43 ip 192.168.1.100

SW1(config)#ip dhcp pool VLAN100
SW1(dhcp-config)#network 10.0.0.0 255.255.255.0
SW1(dhcp-config)#default-router 10.0.0.1

SW1(config)#ip dhcp pool VLAN200
SW1(dhcp-config)#network 10.1.0.0 255.255.255.0
SW1(dhcp-config)#default-router 10.1.0.1

SW1(config)#ntp master
```

On configure les SVI (Switch Virtual Interface) des Vlan, ces adresses sont utilisés comme passerelle par défaut de leur sous réseau, on lance les commandes suivantes :

```
SW1(config)#interface vlan 10
SW1(config-if)#ip address 192.168.1.1 255.255.255.0
SW1(config-if)#interface vlan 100
SW1(config-if)#ip address 10.0.0.1 255.255.255.0
SW1(config-if)#interface vlan 200
SW1(config-if)#ip address 10.1.0.1 255.255.255.0
```

On configure ensuite les pool DHCP avec les commandes suivantes :

Le VLAN 10 a la commande option 43 lancé, cette commande permet de dire aux points d'accès l'adresse IP de leur WLC (ici l'adresse du WLC est 192.168.1.100).

```
SW1(config)#ip dhcp pool VLAN 10
SW1(dhcp-config)#network 192.168.1.0 255.255.255.0
SW1(dhcp-config)#default-router 192.168.1.1
SW1(dhcp-config)#option 43 ip 192.168.1.100
SW1(config)#ip dhcp pool VLAN100
SW1(dhcp-config)#network 10.0.0.0 255.255.255.0
SW1(dhcp-config)#default-router 10.0.0.1
SW1(dhcp-config)#dhcp pool VLAN200
SW1(dhcp-config)#network 10.1.0.0 255.255.255.0
SW1(dhcp-config)#default-router 10.1.0.1
```

On lance en dernier temps la commande suivante pour activer le serveur NTP :

```
SW1(config)#ntp master
```

Voyons à présent comment configurer le WLC :

```
Welcome to the Cisco Wizard Configuration Tool
Use the '-' character to backup


Would you like to terminate autoinstall? [yes]:

System Name [Cisco_10:65:64] (31 characters max): WLC1
Enter Administrative User Name (24 characters max): admin
Enter Administrative Password (3 to 24 characters): *********
Re-enter Administrative Password                : *********

Enable Link Aggregation (LAG) [yes][NO]: yes

Management Interface IP Address: 192.168.1.100
Management Interface Netmask: 255.255.255.0
Management Interface Default Router: 192.168.1.1
Management Interface VLAN Identifier (0 = untagged): 10
Management Interface DHCP Server IP Address: 192.168.1.1
```

Lorsque l'on se connecte au WLC on utilise un câble console, le premier message que l'on peut voir apparaître est qu'il demande s'il est nécessaire de faire la terminer la configuration avec « autoinstall » qui va récupérer la configuration à partir d'un serveur TFTP.

On configure ensuite le nom de système, le nom d'utilisateur et le mot de passe.

On spécifie si l'on veut activer LAG (Link Aggregation), on indique « yes » car la réponse par défaut est ici « NO ». On indique les adresses voulue à configurer.

On continue la configuration basique du WLC en répondant aux question au lieu de lancer les commandes de configuration directement sur une ligne de commande :

```
Virtual Gateway IP Address: 172.16.1.1

Multicast IP Address: 239.239.239.239

Mobility/RF Group Name: jITlab

Network Name (SSID): Internal

Configure DHCP Bridging Mode [yes][NO]: no

Allow Static IP Addresses [YES][no]: yes

Configure a RADIUS Server now? [YES][no]: no
Warning! The default WLAN security policy requires a RADIUS server.
Please see documentation for more details.

Enter Country Code list (enter 'help' for a list of countries) [US]: FR
```

Les trois premières options sont utile, le Virtual Gateway IP est une adresse utilisé lorsque le WLC pour communiquer directement avec ses clients sans fil.

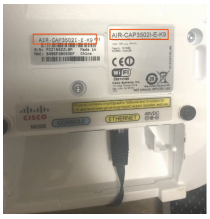
L'adresse Multicast est l'adresse utilisé pour transmettre le trafique vers ces IP.

Le Mobility/RF Group Name est utilisé lorsque l'on a par exemple plusieurs WLC et que l'on veut qu'ils fonctionnent ensemble.

A la suite de la configuration est demandé de configurer le SSID, on en configure pour l'instant 1 et on laisse le reste de la configuration avec les réponses par défaut.

On ne configure pas pour l'instant de serveur RADIUS nous changerons la politique de sécurité WLAN vers PSK donc il ne sera plus nécessaire de configurer de serveur RADIUS.

On entre le code du pays « FR » pour « France », ici on configure la France comme pays car le modèle est compatible avec l'Europe, le nom du modèle du point d'accès est avec un « E » pour Europe.



(Modèle : AIR-CAP3502I-E-K9)

On continue la configuration du WLC :

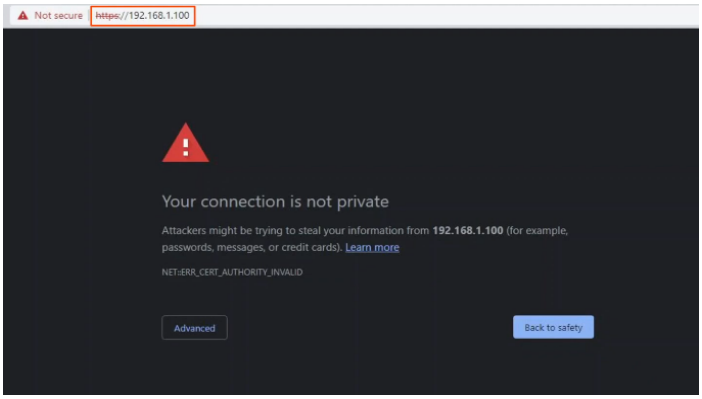
```
Enable 802.11b Network [YES][no]:
Enable 802.11a Network [YES][no]:
Enable 802.11g Network [YES][no]:
Enable Auto-RF [YES][no]:

Configure a NTP server now? [YES][no]: yes
Enter the NTP server's IP address: 192.168.1.1
Enter a polling interval between 3600 and 604800 secs: 3600

Configuration correct? If yes, system will save it and reset. [yes][NO]:
yes

Configuration saved!
Resetting system with new configuration...
```

On choisi ici d'activer les mode 802.11b, 802.11a, 802.11g.

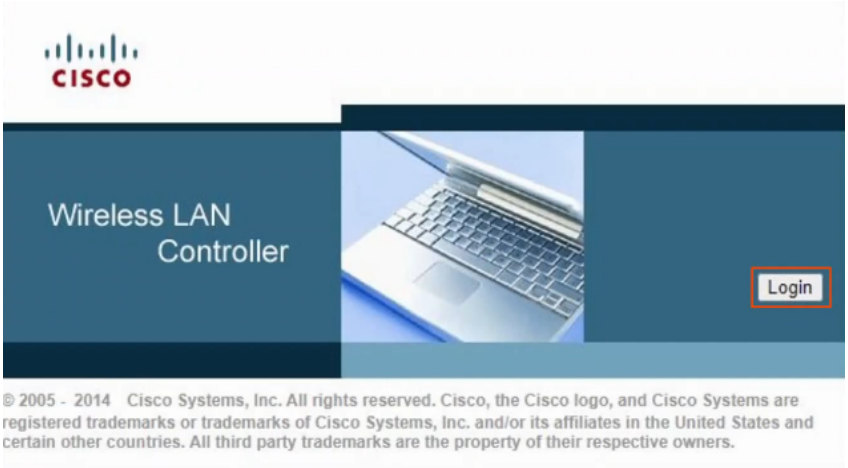
On configure ensuite le serveur NTP pour que le WLC ait le temps correct.

On sauvegarde les paramètre et l'appareil se réinitialise.

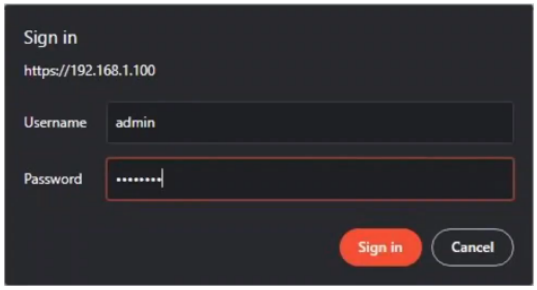
Une fois tout cela configuré il est possible de se connecter au WLC par le moyen d'un navigateur web. On lance donc l'adresse du WLC (192.168.1.100) depuis le navigateur.



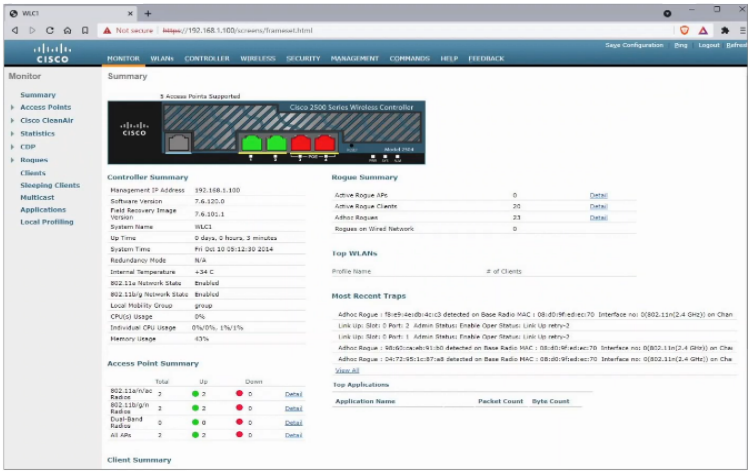
Il faut cliquer sur Advanced et « accéder à 192.168.1.100 »



On peut à présent accéder à l'interface de gestion du WLC, on clique sur « Login » puis on entre les identifiant et mot de passe configurés au départ :

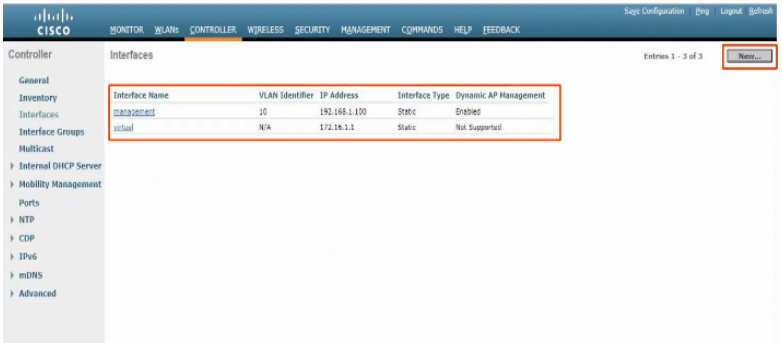


Voici le panel de gestion du WLC :



On peut voir quelles interfaces sont actives et des informations divers sur le WLC avec les points d'accès connectés etc..

Sur la page « Controller » on clique sur « Interfaces », on peut voir la VLAN que l'on a configuré au départ, il n'y a pour le moment que le VLAN 10. On clique sur « New » pour en ajouter une nouvelle.



Les ports de WLC sont des ports physique sur lesquelles des câbles se connectent.

Les interfaces WLC sont des interfaces logique inclus dans le WLC (Par exemple le SVI sur un Switch), WLC a plusieurs sorte de ports :

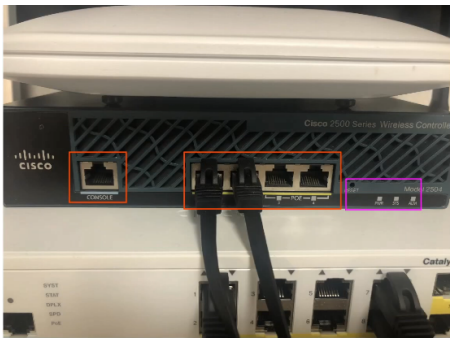
- Service port : Un port de gestion dédié utilisé pour la gestion. Il doit être connecté à un port access du Switch car il ne supporte qu'une seule VLAN. Ce port peut être utilisé pour se connecter à l'appareil tant que celui ci démarre, récupère le système, etc...

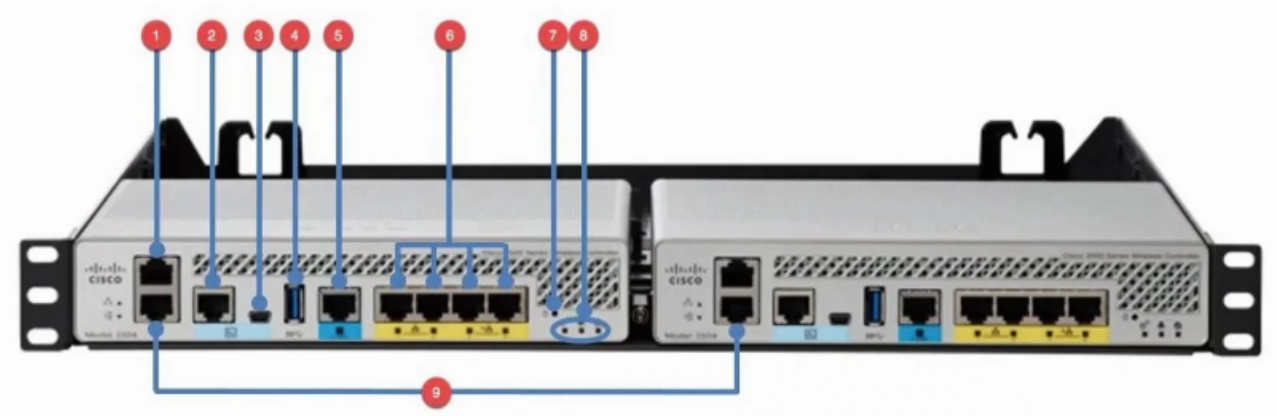
- Distribution system port : Ce sont les ports standard du réseau qui connectent au système de distribution (réseau câblé) et qui est utilisé pour le trafique de donnée. Ces ports se connectent aux ports Switch Trunk, et si plusieurs ports de distribution sont utilisés ils peuvent former un LAG.

- Port console : C'est le port console standard, avec un port RJ45 ou USB.

- Port de redondance : Ce port est utilisé pour se connecter à un autre WLC pour former une pair de haute disponibilité (High Availability en Anglais).

On peut voir ici 4 ports de distribution, et un port Console.





Voici un réseau avec une pair de 2 WLC, voyons chacun des numéro de 1 à 9 :

1) Le Service Port

2) Port Console (RJ45)

3) Port Console (USB)

4) USB (Pour mis à jour logicielle)

5) Port système de Distribution (Multi-gigabit)

6) Port système de Distribution (1-gig)

7) Bouton Reset

8) LED de Statut

9) Port de Redondance

Les WLC ont différents type d'interfaces :

- Interface de gestion : utilisé pour gérer le trafique comme Telnet, SSH, HTTP, HTTPS, Authentification RADIUS, NTP, Syslog, etc... Tunnels CAPWAP sont aussi formés vers/depuis l'interface de gestion du WLC.

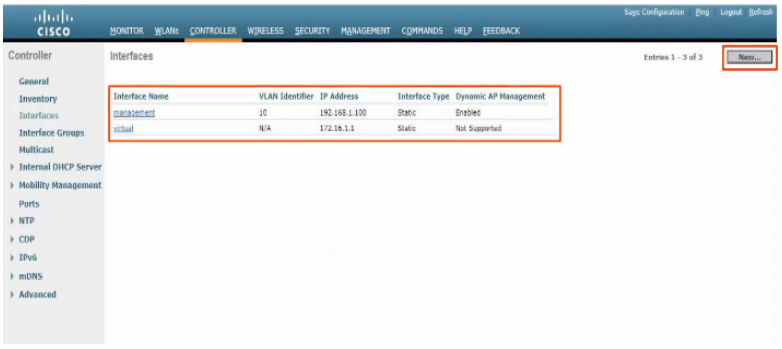
- Interface de gestion de redondance : Lorsque deux WLC sont connectés par leurs ports de redondance, un WLC est « active » et l'autre est en « standby ». Cette interface est utilisé pour connecter et gérer le WLC « standby ».

- Interface Virtuel : Cette interface est utilisé lorsque l'on communique avec les clients sans fil vers le relai de requête DHCP, que l'on fait fonctionner une authentification web client, etc...

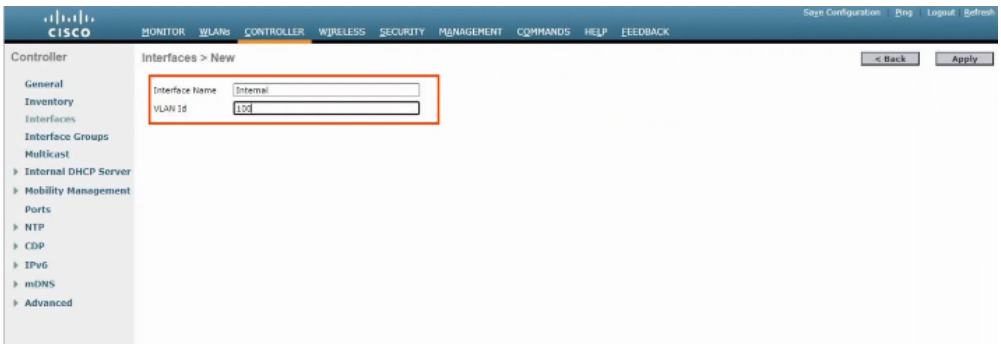
- Interface de port de Service : Si le port de service est utilisé, cette interface est lié vers celle ci et utilisé pour la gestion externe.

- Dynamic interface : Ce sont les interfaces utilisés pour cartographier un WLAN vers un VLAN. Par exemple, le trafique depuis le WLAN « internal » sera envoyé vers le réseau câblé depuis l'interface dynamic du WLC « internal »

Retournons à présent sur l'interface GUI pour configurer des interfaces Dynamique :

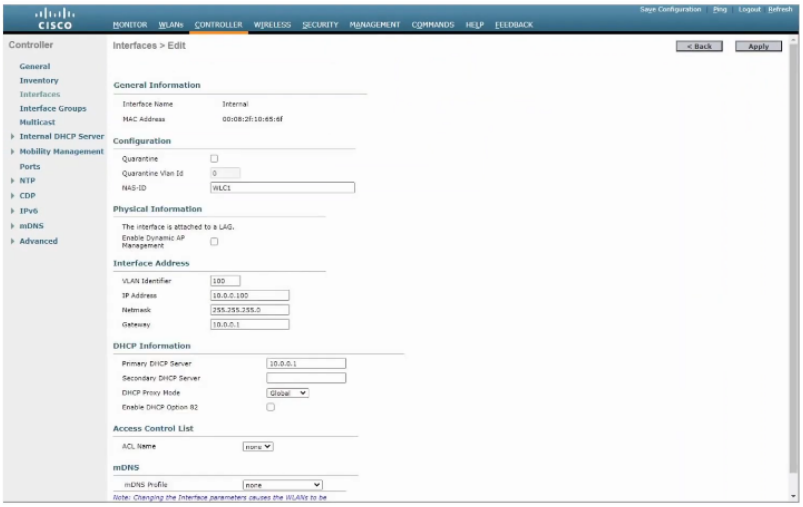


On configure l'interface pour le trafique WLAN interne :

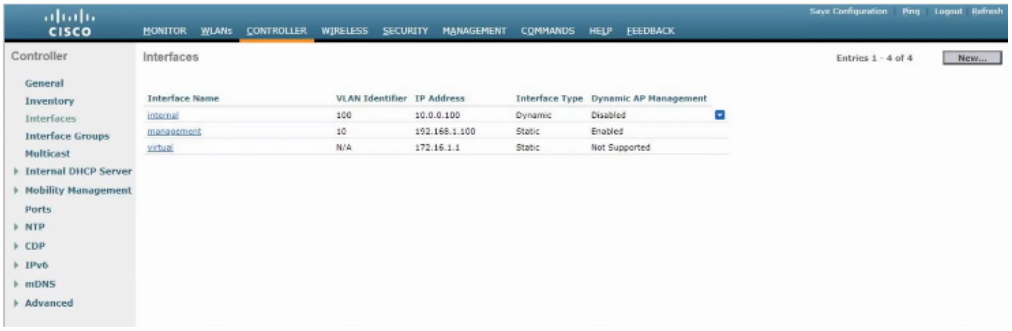


L'écran affiche la page suivante :

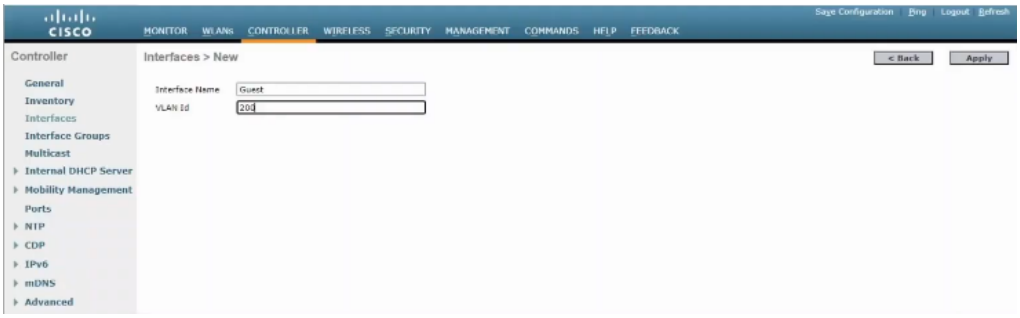
On indique l'adresse IP du vlan et son masque de sous réseau puis on applique les modifications en cliquant sur « Apply »



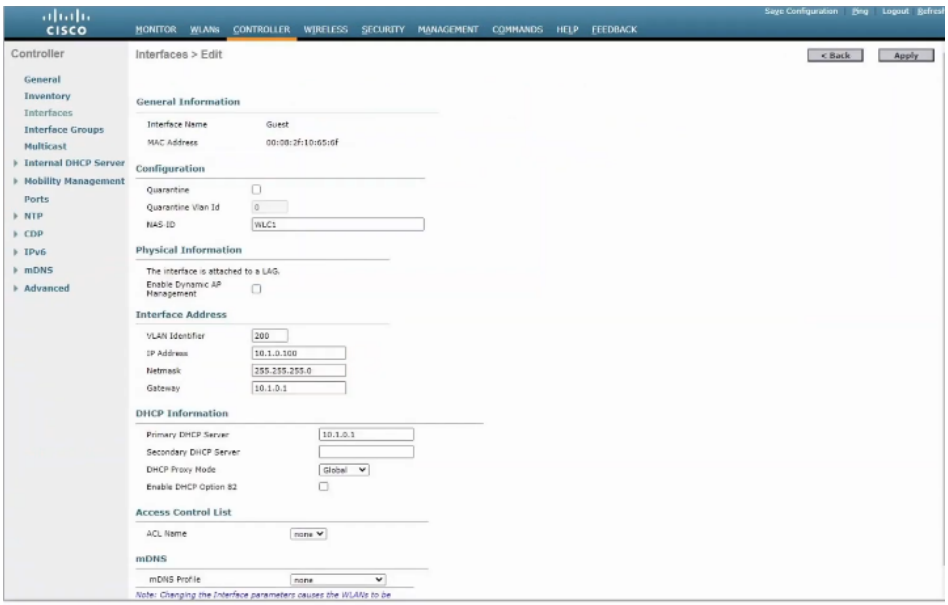
Depuis le menu Dynamic on peut à présent voir les 3 interfaces, dont celle venant d'être crée qui est « Internal »



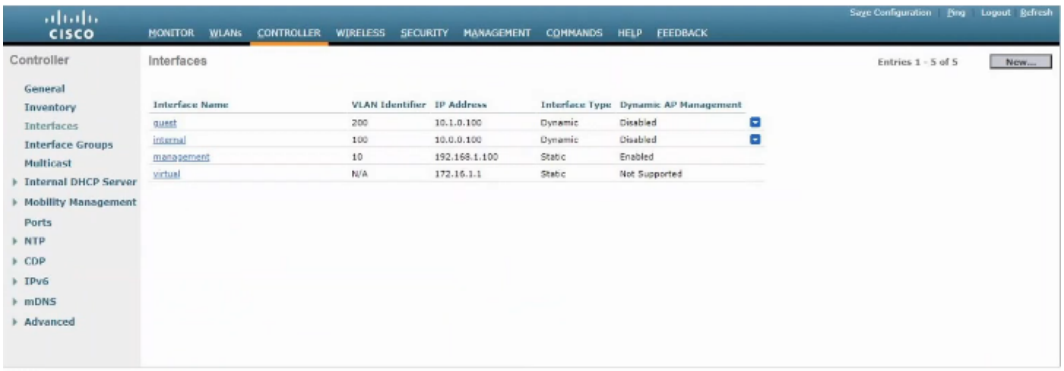
Il reste à créer l'interface « Guest », pour cela on procède comme auparavant en spécifiant les paramètre voulant être appliqué pour l'interface « Guest » :



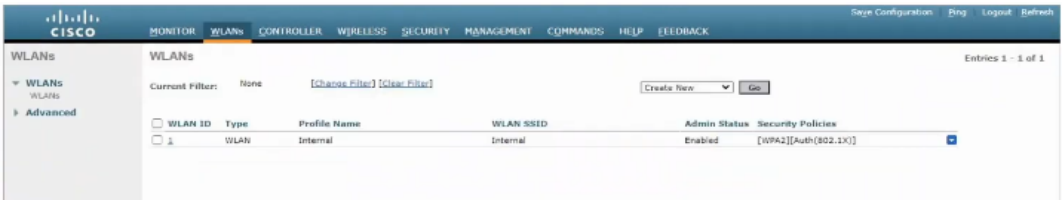
Sur cette page on indique les adresse IP nécessaire correspondant au VLAN :



Toutes les interfaces sont à présent bien configurés :

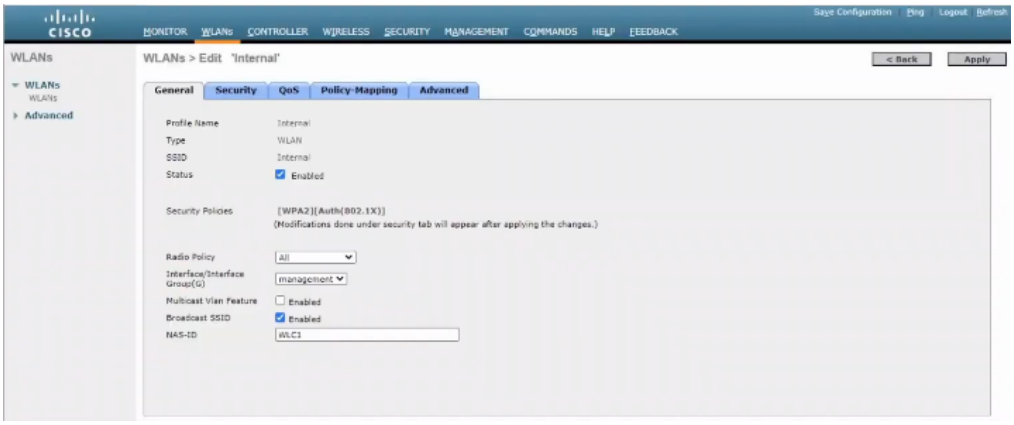


Faisons la configuration des WLAN :

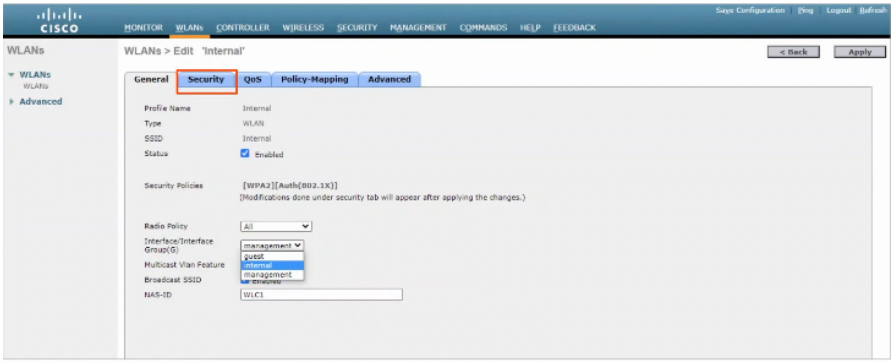


Il n'y en a ici qu'une seul, le mode de politique de sécurité est WPA2, 802.1X, donc le mode Enterprise, nous allons configurer le mode PSK.

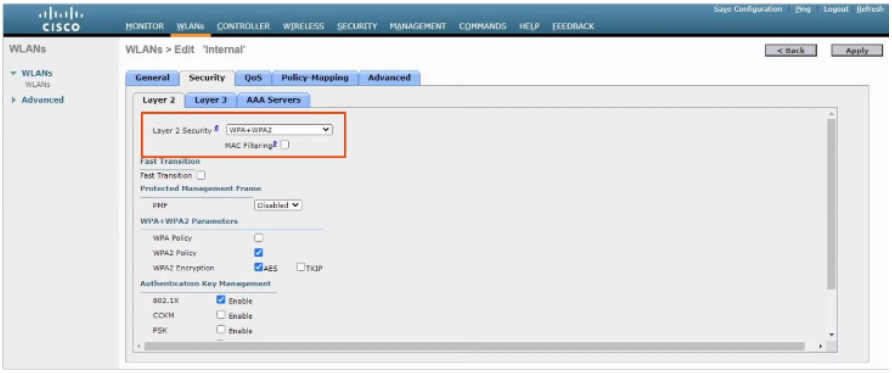
En cliquant sur le « 1 » à gauche il est possible de modifier ce WLAN.



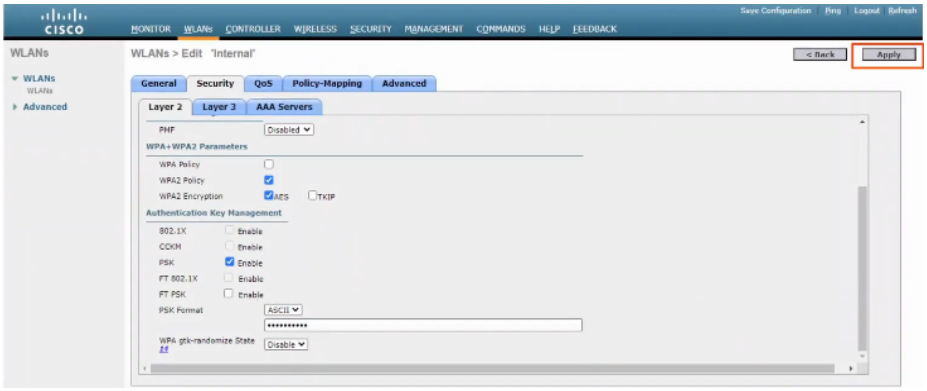
Nous allons modifier le « Interface Group » pour Internal et non plus pour « management ».



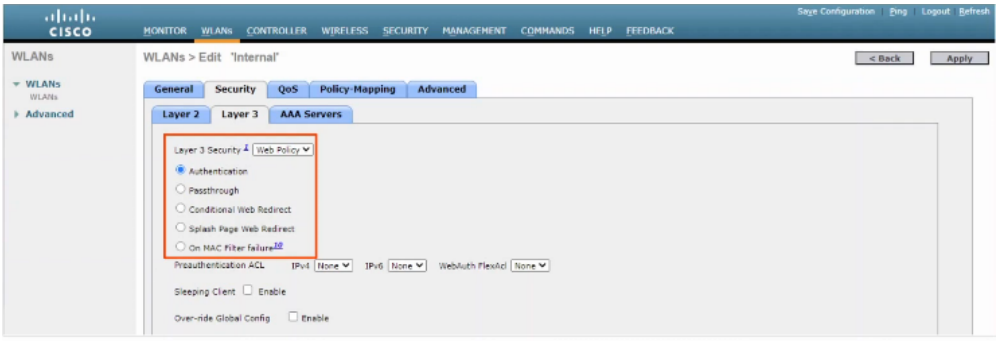
Pour modifier afin d'utiliser le mode PSK, on se rend sur l'onglet « Security »



Et nous modifions la gestion des clefs d'authentification et sélectionnons PSK :



On sélectionne aussi le PSK format pour ASCII et on entre un mot de passe et on applique les modifications. Il est possible de modifier le Layer 3 et de modifier le mode d'authentification :



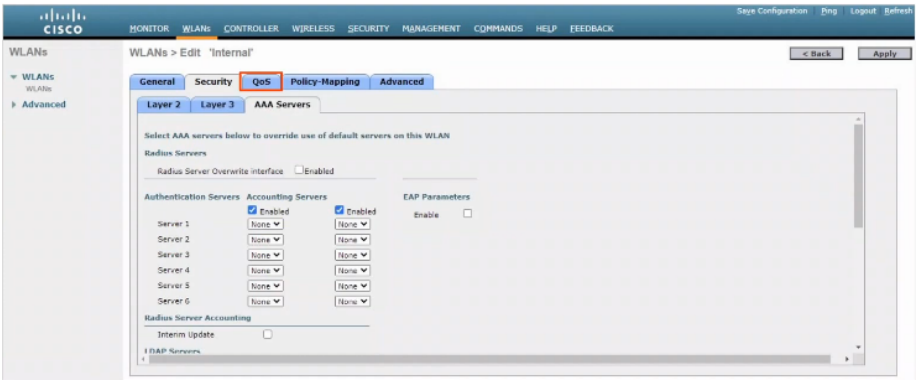
Plusieurs modes sont présents :

- Web Authentication : Après que les clients sans fil ont leurs adresse IP et essaient d'accéder à la page web, ils doivent entrer un nom d'utilisateur et un mode de passe pour s'authentifier.

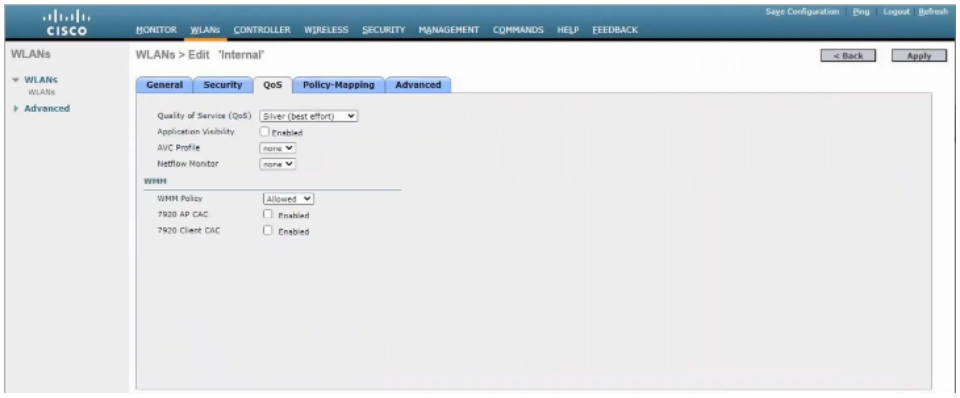
- Web Passthrough : même chose que Web Authentication mais aucun nom d'utilisateur ou mot de passe n'est requis. Un signal ou déclaration apparaît et le client doit tout simplement accepter pour avoir accès à Internet.

- Conditional et Splash Page, sont des options de redirection web similaire mais qui requière de manière additionnel une authentification 802.1X couche 2.

Il y a aussi un mode AAA :



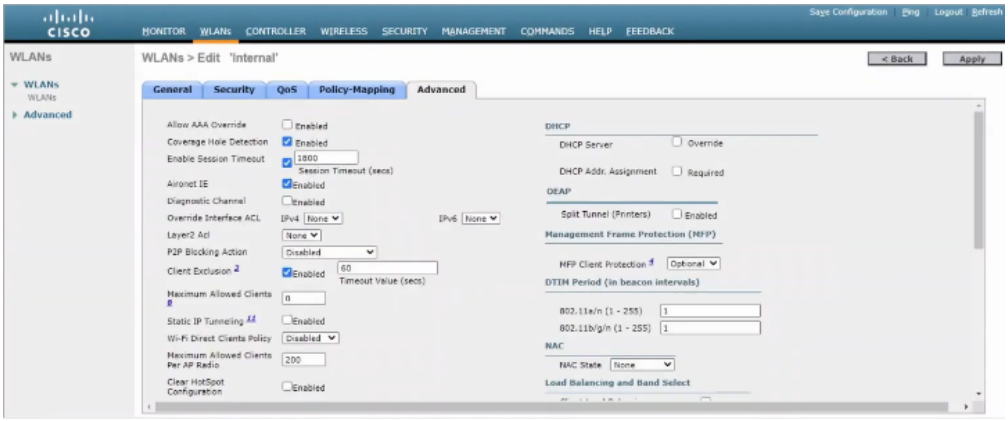
Voyons l'onglet QoS :



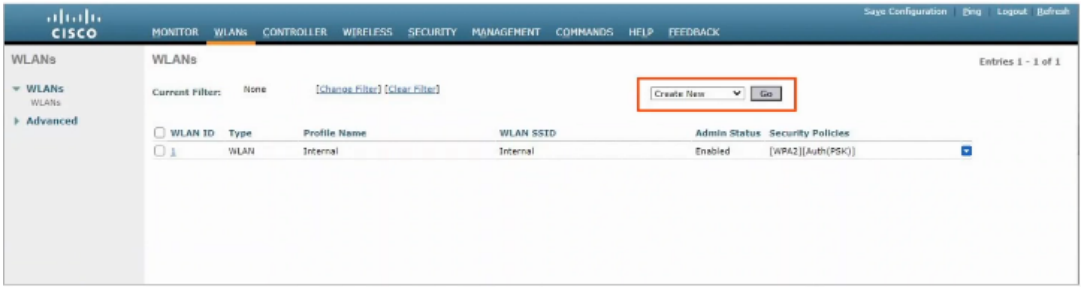
Il faut vérifier que le mode activé est bien le mode Silver (Best Effort).

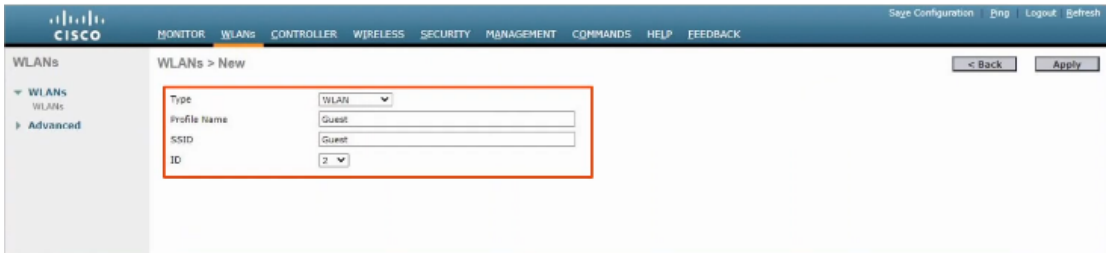
Il existe d'autres modes comme Gold (Vidéo), Platinum (Voice), Bronze (Background)

Dans l'onglet « advanced » on peut voir différents services :

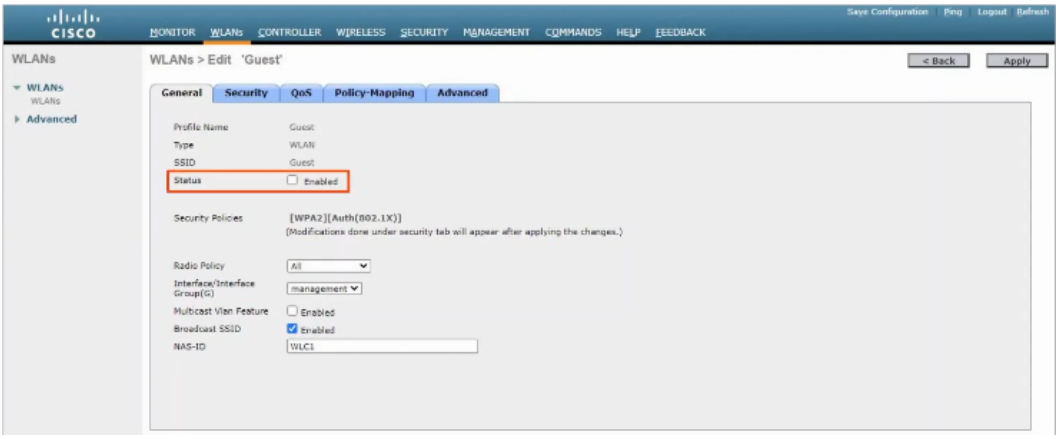


Pour créer un nouveau WLAN ou clique sur « Go », nous allons vréer le WLAN Guest.





On indique le Type, le Profile Name et le SSID, il n'est pas nécessaire qu'il soit identiques.

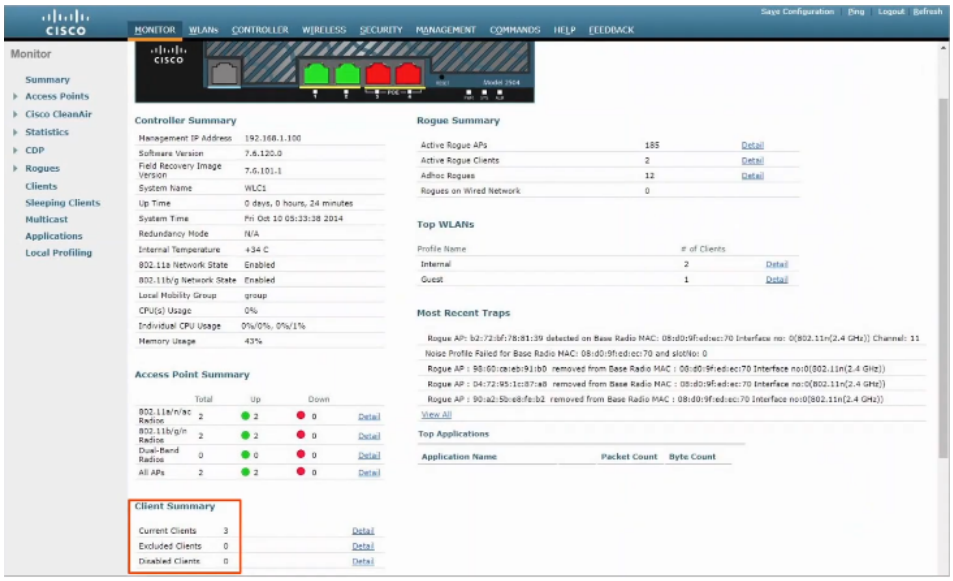


On arrive ensuite à cette page :

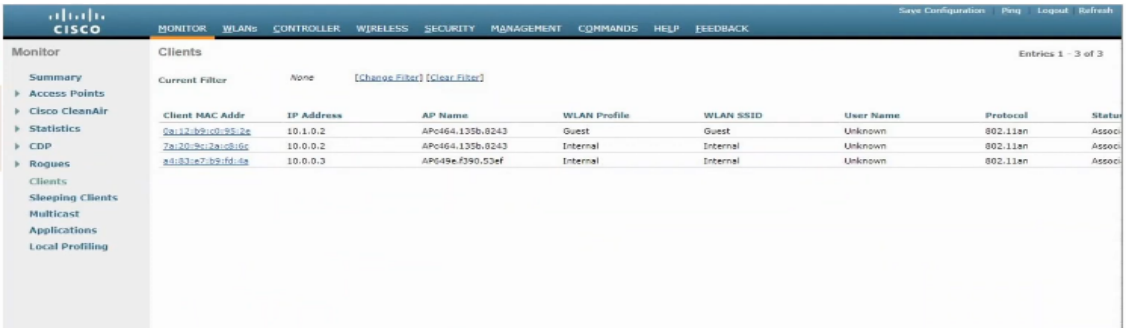
Il doit être changé différents paramètres. Le « status » doit être activé. Et l'interface Group doit aussi être changé pour « Guest ». On change aussi le mode d'authentification pour PSK comme vu précédemment.

Les deux WLAN sont à présent bien présent comme on peut voir :

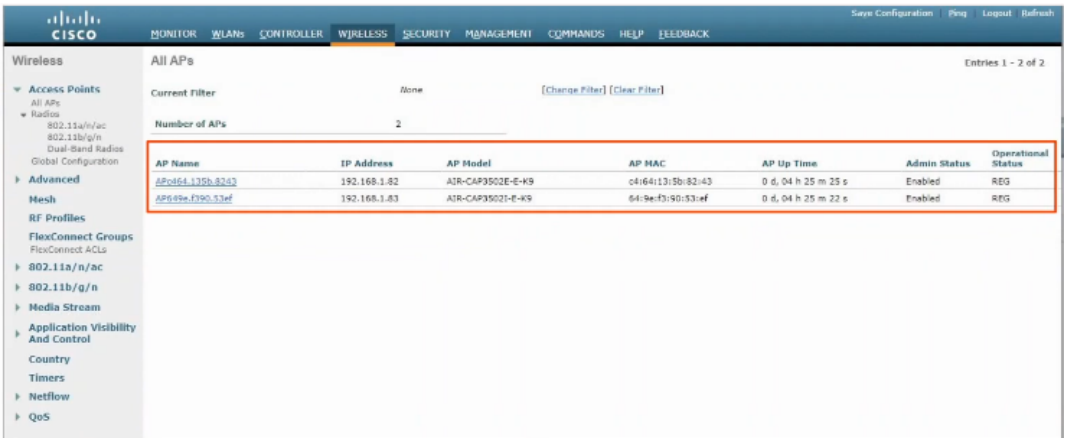
Lorsqu'un client se connecte aux point d'accès on peut voir que le nombre de client augmente.



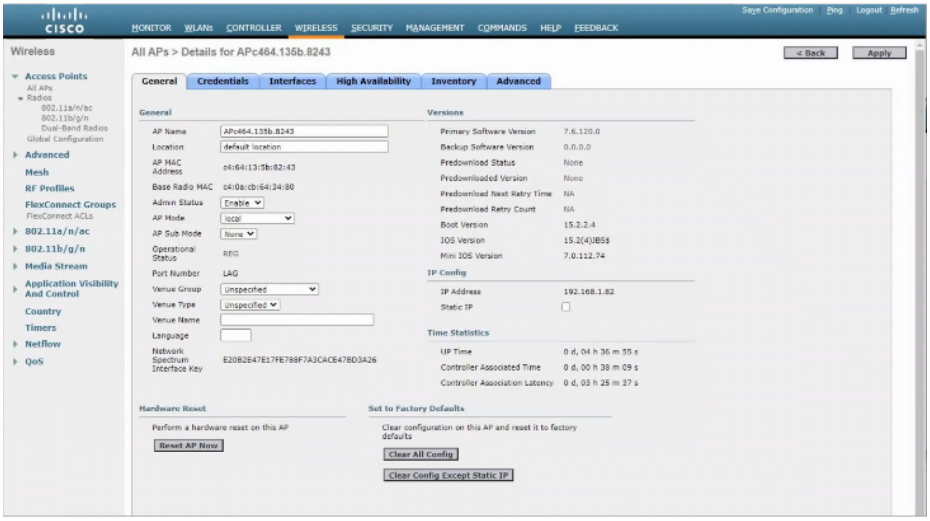
Pour afficher les clients ont clique sur l'onglet client on peut voir affiché leurs informations :

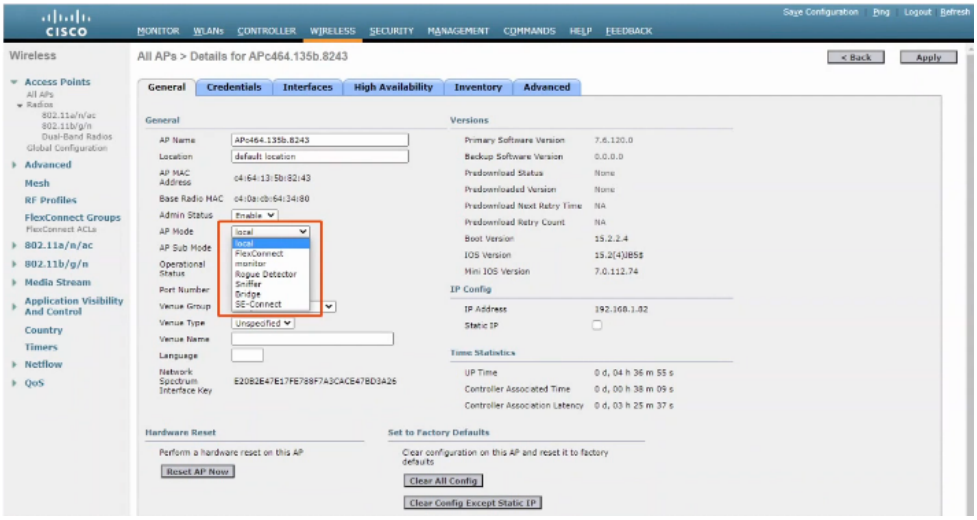


On peut aussi afficher les points d'accès avec leurs informations :

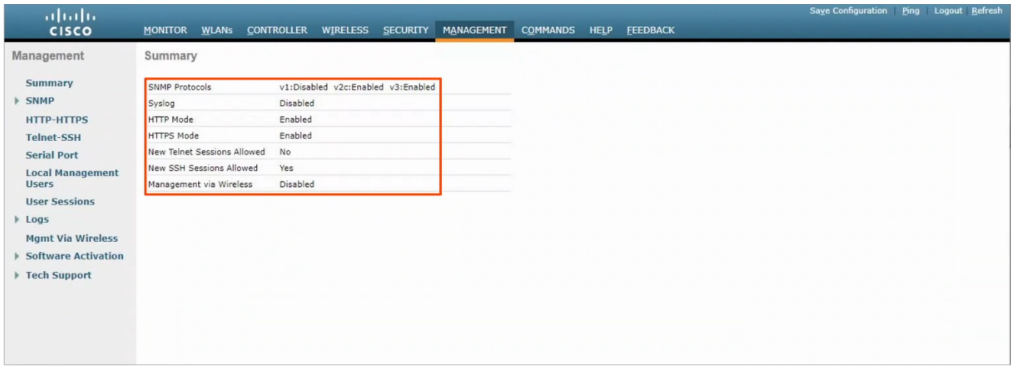


Pour modifier les paramètres de l'uns des points d'accès on clique sur l'un d'eux :





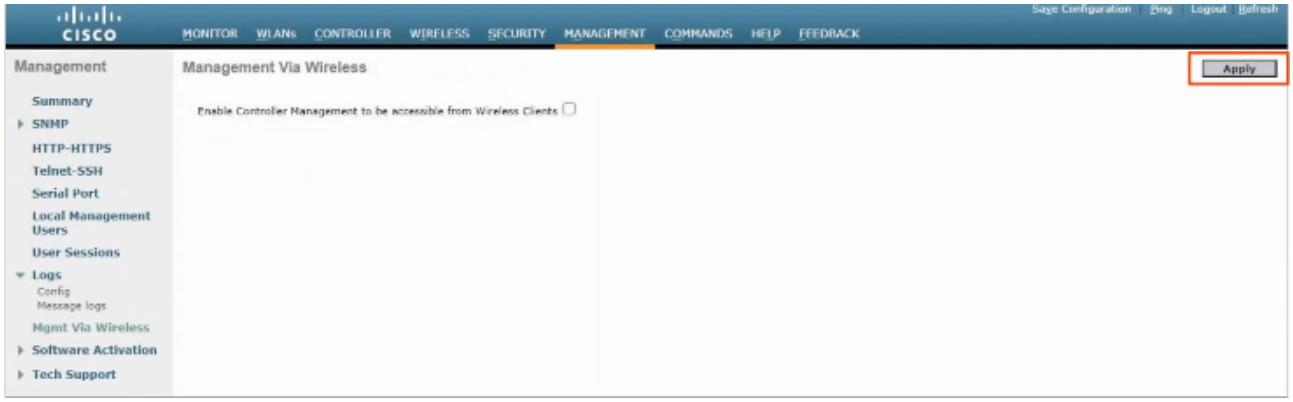
Sur la partie AP Mode on peut changer le mode de configuration, flexconnect, RogueDetector, Sniffer, etc.. comme vu auparavant.



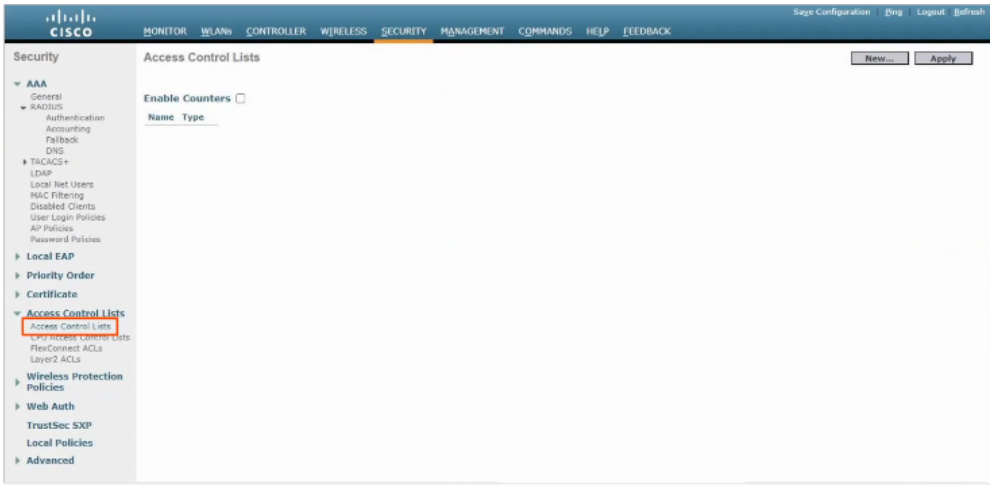
Sur l'onglet Management on peut voir différents paramètres, comme par exemple la version de SNMP activé, le mode HTTP activé, Syslog, SSH, etc.

Telnet est ici désactivé comme on peut le voir :

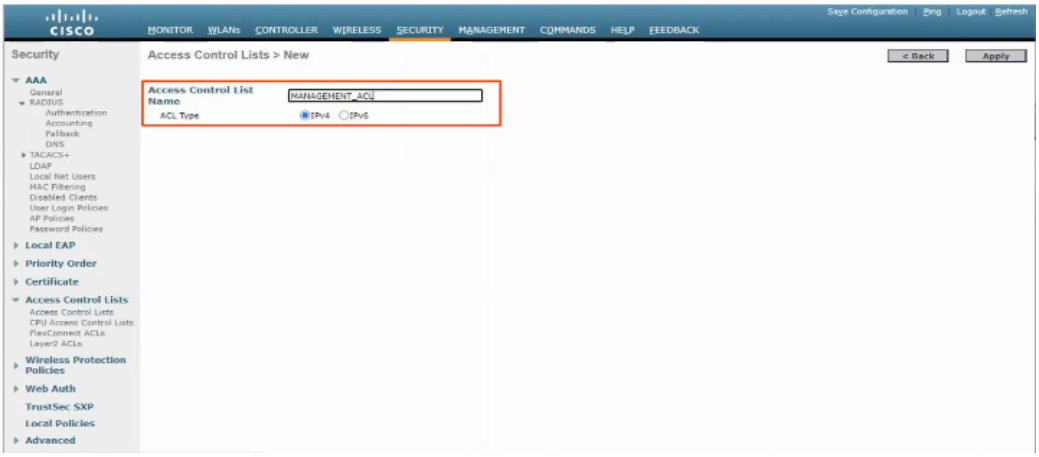
On peut activer un mode pour que les clients d'un point d'accès puisse avoir accès à la gestion du contrôleur.



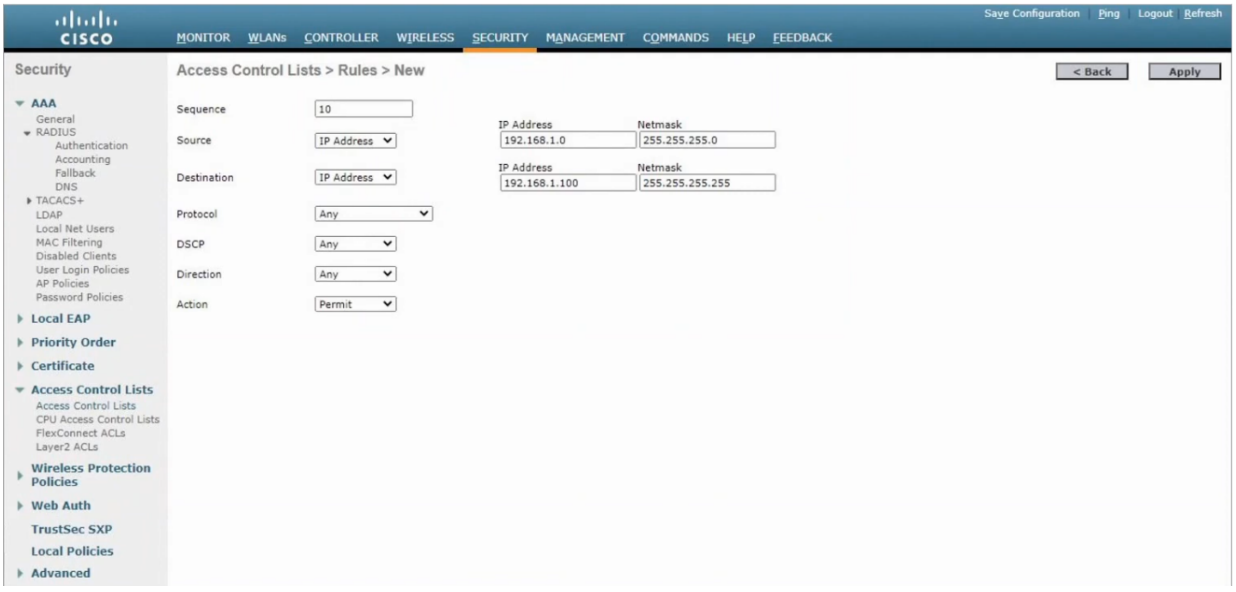
Dans l'onglet « Security » on peut configurer une ACL :



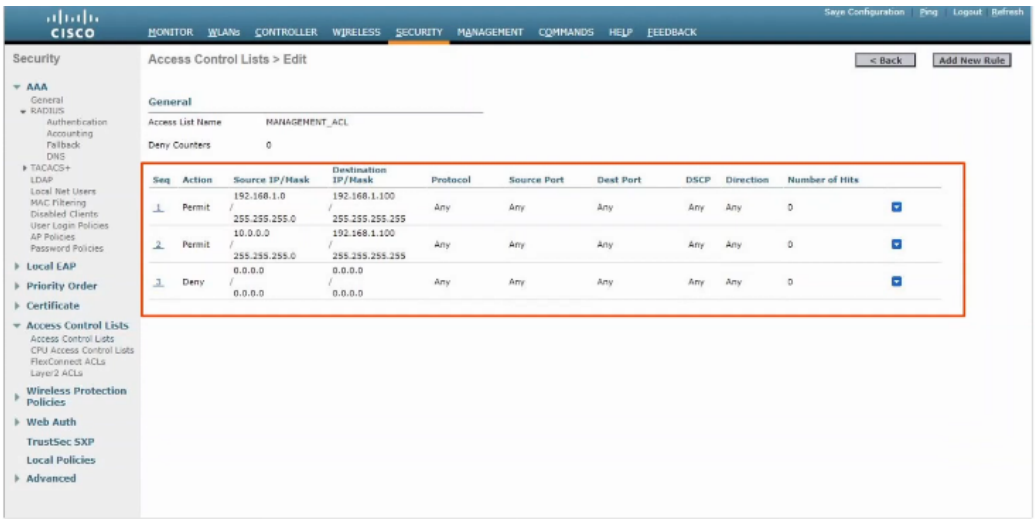
On donne un nom à l'ACL :



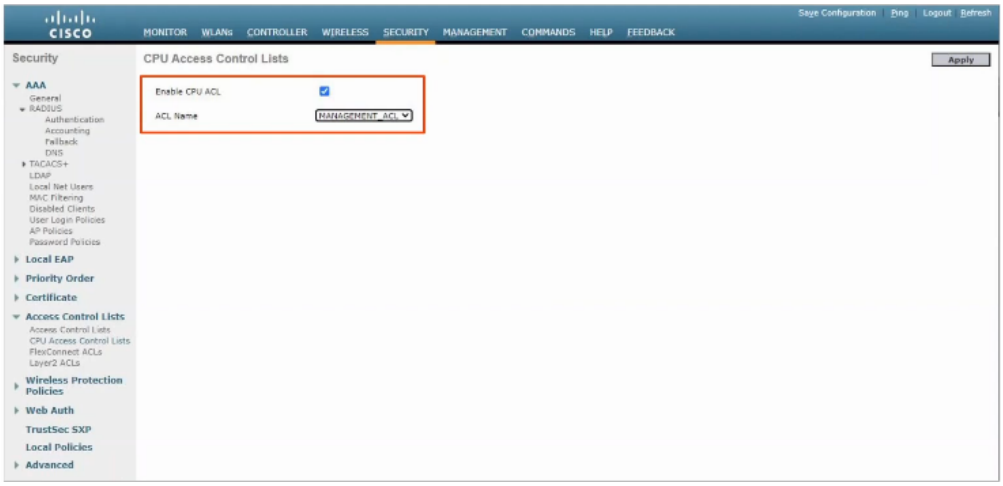
L'ACL est bien crée mais n'a pas de règle, on ajoute des règles en cliquant sur « add new rule »



On crée 3 règles comme suit :



Pour faire Appliquer ces ACL on clique sur : « CPU Access Control Lists »



Puis on sélectionne Enable CPU ACL et on clique sur Apply pour appliquer les modifications.

Cours 59 : Automatisation Réseau

Dans ce cours nous verrons l'automatisation du réseau, tout d'abord nous verrons pourquoi utiliser l'automatisation d'un réseau et les bénéfices à l'utiliser. Puis nous ferons un plan logique des fonctions du réseau (Plan de donnée, plan de contrôle, plan de gestion).

Et nous verrons ce qu'est le Software-define networking (SDN), les APIs et la sérialisation des données.

Il est important d'avoir une connaissance basique sur l'automatisation du réseau. Dans un modèle traditionnel, les ingénieurs gèrent les appareils un à un en se connectant à leur CLI (Command Line Interface) Par SSH. Il y a certains inconvénients de configurer des appareils un à un :

- De simple petites erreurs peuvent être présentes.

- Cela prend du temps et est très inefficace dans de grands réseaux.

- Il est difficile de s'assurer que tous les appareils adèrent bien aux standard de configurations de l'entreprise.

L'automatisation des réseaux offre plusieurs avantages :

- Les erreurs Humaines sont réduites (configurations, etc.)

- Les réseaux deviennent plus évoluable. Les nouveaux déploiements, changement de grand réseaus, et réglage de problèmes peut être implémenté dans une fraction du temps.

- Les politiques de réseau peuvent être assurés (Configuration de standard, version logiciel, etc.)

- L'efficacité amélioré des opération du réseau réduisent les coûts du réseau car chaque tâche peut demander plusieurs heures par personne.

Il existe une grande varité d'outils et de méthodes pouvant être utilisé pour automatiser des tâches dans le réseau :

- SDN

- Ansible

- Puppet

- Script Python

- etc...

Pour automatiser un réseau il nous faut établir un plan logique des appareils.

Qu'est ce qu'un Routeur fait ?

Il transmet les messages entre les réseaux en examinant les information de l'entête de couche 3. Il utilise un protocole comme OSPF pour partager le routage de l'information avec d'autres routeurs et construire une table de routage.

Il utilise ARP pour construire une table ARP, en cartographiant les adresses IP et les adresses MAC.

Il utilise Syslog pour garder les logs des évènements qui se passent.

Il permet à un utilisateur de se connecter par SSH et de gérer.

Qu'est ce qu'un Switch fait ?

Il transmet les messages d'un LAN en examinant l'information dans l'entête de couche 2.

Il utilise STP pour s'assurer qu'il n'y a pas de boucle de couche 2 dans le réseau.

Il construit une table d'adresse MAC en examinant les trames des adresses MAC source.

Il utilise Syslog pour garder les logs des évènements qui se passent.

Il permet à un utilisateur de se connecter par SSH et de gérer.

Les fonctionnalités des appareils d'un réseau peuvent être divisés logiquement (et catégorisés) en des plans :

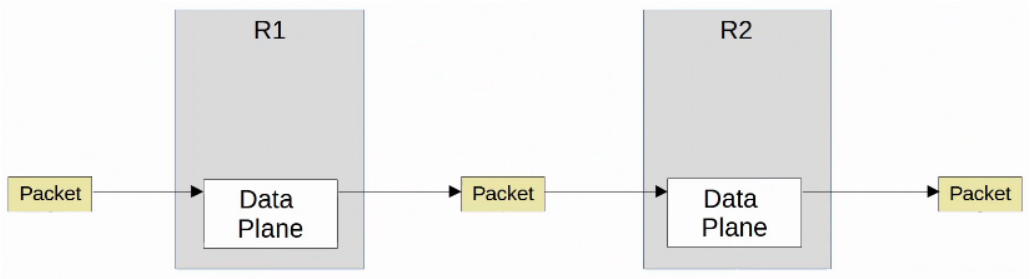
- Le plan de données : Toutes les tâches impliqués dans le partage/trafique des données de l'utilisateur depuis une interface vers une autre font partie du plan des données.

Un routeur qui reçoit un message, regarde la route qui correspond le plus à sa table de routage, et le transmet à l'interface la plus approprié pour le prochain bond. Il désencapsule l'entête original de couche 2, et la réencapsule avec une nouvelle entête destiné pour le prochain bond d'adresse MAC.

Le Switch reçoit le message, et regarde l'adresse MAC de destination, et le partage en dehors de l'interface approprié (Ou bien inonde le réseau). Cela inclut des fonctions comme ajouter ou supprimer des balises 802.1q VLAN. Le NAT (Change l'adresse source/destination avant de transmettre le message) cela fait aussi partie du plan des données.

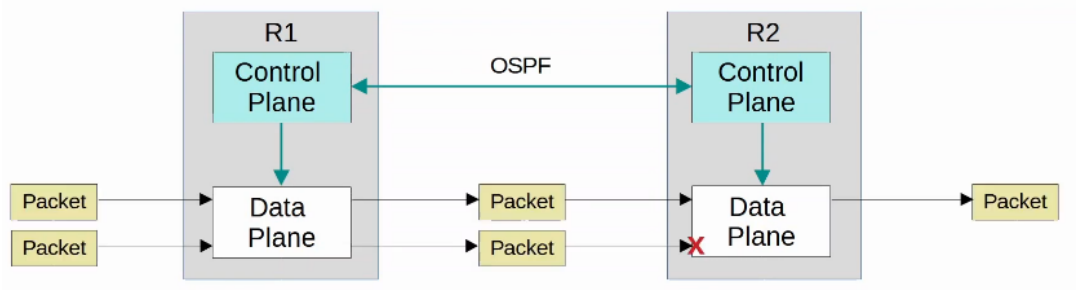
Décider de transmettre ou bloquer les messages à cause d'une ACLs, port Security, etc.. fait aussi partie du plan des données. Le plan de données est aussi appelé le « plan de transmission ».

Cela fonctionne comme sur le schéma ci dessous :



- Le plan de contrôle : Ce plan permet de répondre à comment un le plan de données d'un appareil faire pour transmettre ses décisions. Avec les tables de routage, les adresses MAC, les tables ARP, STP, etc... Les fonctions qui construisent ces tables (et d'autres fonctions qui influencent le plan de données) dont partie du plan de contrôle. Le plan de contrôle, contrôle ce que le plan de donnée fait par exemple en construisant la table de routage d'un routeur. Le plan de contrôle fait un travail « aerien » par exemple si OSPF ne transmet pas les paquets de données à un utilisateur, mais il informe le place de données comment les paquets devraient être transmis. Si STP n'est pas directement impliqué dans le processus de transmission des trames, mais il informe le plan de données à propos de quelles interface devrait ou ne devraient pas être utilisés pour partager les trames transmises. Les messages ARP ne sont pas des données utilisateurs, mais ils sont utilisé pour construire une table ARP qui est utilisé dans le processus de transmission de données.

Cela fonctionne comme sur le schéma suivant :



Dans un réseau traditionnel, le plan de donnée et le plan de contrôle sont tout les deux distribués. Chaque appareils à ses propres plans de données et ses propre plan de contrôle. Les plans sont distribués sur le réseau.

- Le plan de gestion : Tout comme le plan de contrôle, le plan de gestion fait un travail « aerien ».

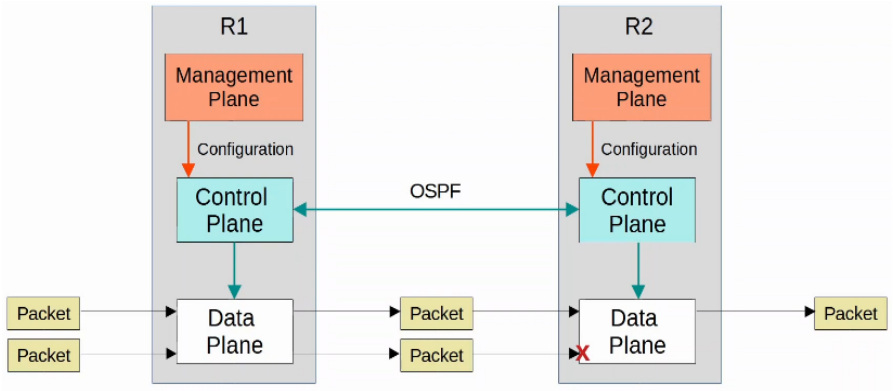
Le plan de gestion n'affecte pas directement le transfert des messages dans le plan de données.

Le plan de gestion consiste dans l'utilisation de protocoles qui sont utilisés pour gérer les appareils.

Comme par exemple SSH/Telnet, utilisés pour se connecter au CLI d'un appareil pour le configurer ou le gérer. Ou bien Syslog utilisé pour garder les logs des évènements qui se passent sur l'appareil.

SNMP utilisé pour monitorer les opérations de l'appareil. NTP est utilisé pour maintenir un temps précis sur l'appareil.

Cela peut se résumer par le schéma suivant :



Le plan de donnée est la raison pour laquelle l'on achète des routeurs et des Switchs (et des infrastructure réseau en général), pour transmettre les messages. Seulement le plan de contrôle et de gestion sont tout deux nécessaire pour que le plan de donnée fasse correctement son travail.

Les opérations de gestion et de contrôle sont géré par le CPU. Cela n'est pas utile pour les opérations du plan de donnée car le fonctionnement du CPU est lent. Au lieu de cela du matériel spécialisé ASIC (Application-Specific Integrated Circuit) est utilisé. ASICs sont des puces construits dans cette interêt spécifique.

En voici une image :

En utilisant un Switch comme exemple :

Lorsque la trame est reçu, le ASIC (Non pas le CPU) est responsable de la logique de Switching.

La table Adresse MAC est stocké dans une sorte de mémoire appelé TCAM (Ternary Content-Addressable Memory).

Un autre nom commun pour les adresses MAC est table CAM.

Le ASIC alimente l'adresse MAC de destination de la trame dans la TCAM, qui retourne une adresse MAC correspondant à l'entrée de l'adresse MAC.

La trame est ensuite transmise en dehors vers l'interface approprié.

Les routeurs modernes utilisent aussi du matériel similaire dans le plan de donnée : Un ASIC

est désigné pour la transmission logique, et des tables stockés dans le TCAM.

Pour résumer, lorsqu'un appareil reçoit un trafique de contrôle/gestion (destiné à lui même) il va utiliser le CPU. Lorsqu'un appareil reçoit le trafique de données qui devrait passer par l'appareil, il utilisé ASIC pour une meilleure rapidité.

Voyons le concept des SDN pour Software-Defined Networking. Il s'agit d'une approche des réseaux qui centralise le plan de contrôle dans une application appelé controlleur.

Ce concept est similaire à celui déjà appris qui concerne les WLAN.

Le SDN est aussi appelé Software-Defined Architecture (SDA) ou Controller-Based Networking.

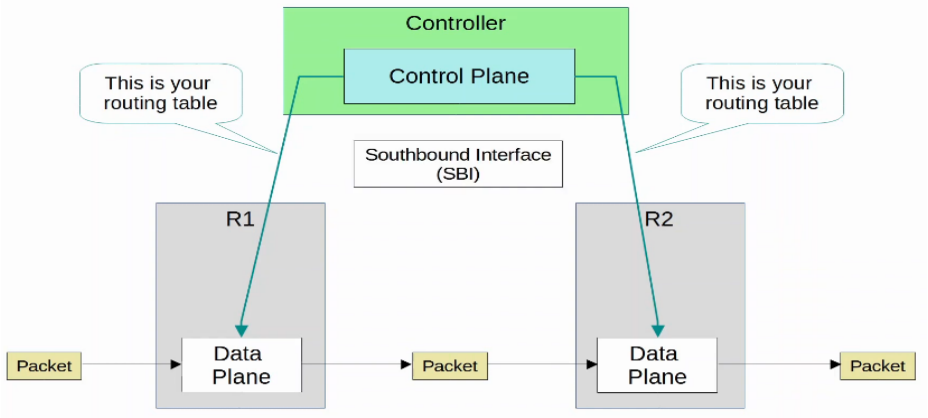
Le plan de contrôle traditionnel utilisé une architecture distribué.

Par exemple chaque routeur dans un réseau lance OSPF et les routeurs partagent leurs informations de routage et calculent ensuite leurs route préféré pour chaque destination.

Un controlleur SDN centralise les fontions du plan de contrôle comme le calcule du routage.

Cela est juste un exemple de combien le plan de contrôle varie.

Le controlleur peut intéragir de manière programmé avec les appareils du réseau en utilisant des API (Application Programming Interface).



Voici un schéma qui permet de résumer :

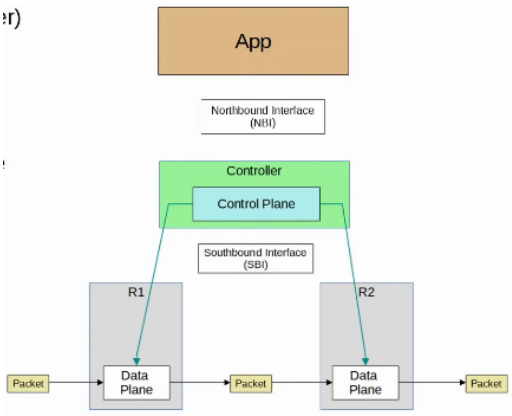
Le Southbound Interface (SBI) est utilisé pour la communication entre le controlleur et les appareils du réseau qu'il controle. Cela consiste en général de l'utilisation d'un protocole de communication et d'un API (Application Programming Interface). Les API facilitent l'échange de données entre des programmes. Les données sont échangés entre le controlleur et les appareils du réseau.

Un API sur les appareils du réseau permettent au controlleur d'accéder aux informations sur un appareil, il contrôle leur table de plan de données, etc.

En utilisant le SBI, le controlleur communique avec les appareils géré et recueille des informations à propos d'eux, par exemple les appareils présent sur le réseau, la topologie (Comment les appareils sont connectés entre eux), les interfaces disponibles sur chaque appareil, leurs configurations.

Le Northbound Interface (NBI) est ce qui permet d'intéragir avec le controleur, d'accéder aux données qu'il recueille à propos du réseau, il le programmen et fait des changements dans le réseau via SBI.

Cela peut se résumer par le schéma suivant :

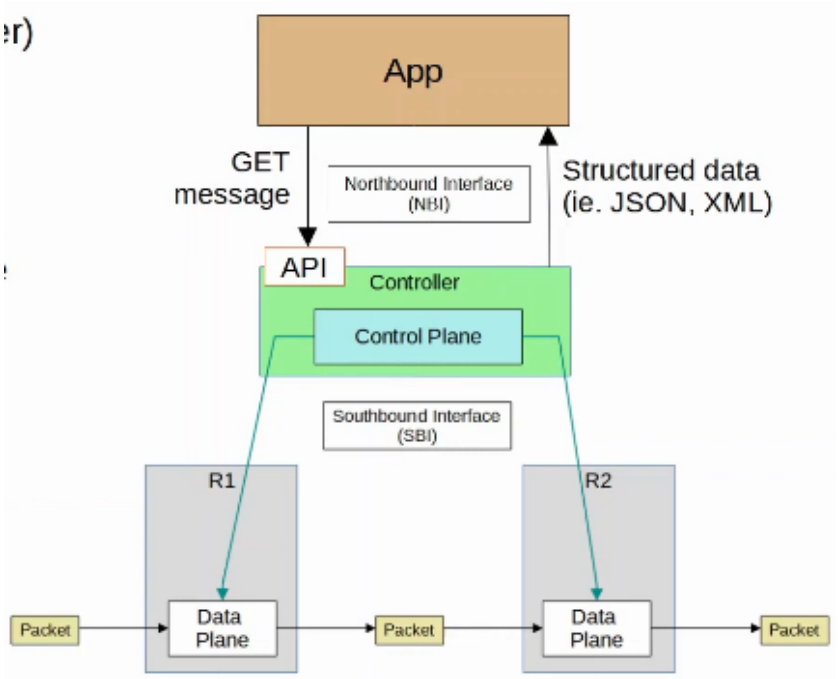


Le REST API est utilisé sur le controlleur comme interface pour que les applis interagisse avec lui.

REST est l'acronyme de Representational State Transfer

Les données sont envoyés dans un format structurés (Sérialisé) comme JASON ou XML.

Le schéma suivant résume cela :



Cela rend plus facile l'utilisation des données pour les programmes.

Voyons un comparatif de l'automatisation des réseaux traditionnels et des SDN.

Les tâches de réseaux peuvent être automatisés dans une architecture réseau traditionnel aussi :

Avec des script pouvant être écris (en utilisant Python par exemple) pour envoyer des commandes à plusieurs appareils en même temps.

La bonne utilisation de Python pour des Expression Régulière peut analyser par la commande « show » pour recueillir des informations à propos des appareils du réseau.

Le résultat de la commande « show » permet de comprendre facilement :

```
Switch#show interfaces
GigabitEthernet0/0 is up, line protocol is up (connected)
  Hardware is iGbE, address is 0cb0.28f6.5500 (bia 0cb0.28f6.5500)
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Unknown, Unknown, link type is auto, media type is unknown media type
  output flow-control is unsupported, input flow-control is unsupported
  Auto-duplex, Auto-speed, link type is auto, media type is unknown
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:09, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     83 packets input, 8576 bytes, 0 no buffer
     Received 82 broadcasts (82 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 82 multicast, 0 pause input
     54 packets output, 7221 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

La centralisation robuste des données collectés par les controlleurs SDN facilite grandement ces fonctions. Le controlleur collecte l'information à propos de tous les appareil du réseau.

Les API Northbound permettent aux applications d'accéder aux informations dans un format facile à comprendre pour les programmes (Comme par exemple JSON, XML).

La centralisation des données facilite l'analyse large du réseau.

Les outils SDN peuvent fournir des bénéfices d'automatisation sans que cela requière des scripts ou applications.

Il n'est plus nécessaire d'être expert dans l'automatisation pour utiliser des outils SDN.

Les API permettent tout de même aux application tiers d'intéragir avec le contrôlleur, qui peut être très puissant. SDN et l'automatisation ne sont pas les même choses, les architectures SDN facilitent grandement l'automatisation de tâches variés dans le réseau par des controlleurs SDN et des API.

Cours 60 : JSON, XML & YAML

Dans ce cours nous verrons ce qu'est JSON, XML et YAML. Ce sont des langages connus pour la sérialisation des données, ou le formatage de sérialisation des données. Ceci nous permet de formater ou structurer les données de manière standard pour que plusieurs applications puissent communiquer entre elles. Nous verrons tout d'abord ce qu'est la sérialisation des données, puis JSON (JavaScript Object Notation) comment l'interpréter et l'utiliser, nous verrons aussi XML (Extensible Markup Language) et YAML (YAML Ain't Markup Language).

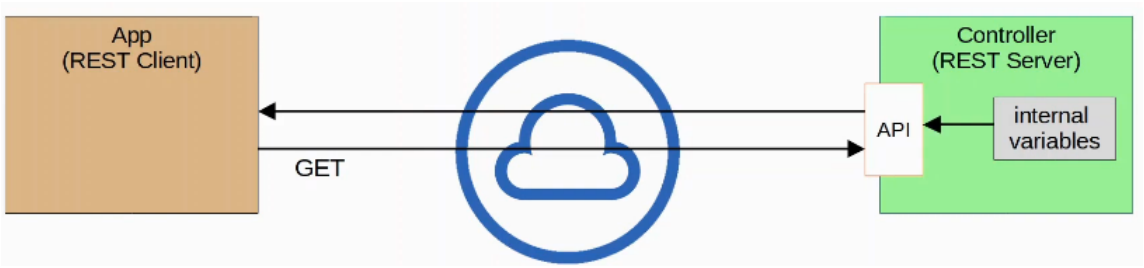
La sérialisation des données est le processus de conversion de données dans un format ou une structure standard qui peut être stocké (dans un fichier) ou transmis (par le réseau) et reconstruit plus tard (par exemple par une différente application). Cela est utile car cela permet aux données d'être communiqués entre plusieurs applications d'une manière à ce que les deux applications comprennent. Le langage de sérialisation des données permet de représenter des variables avec du texte.

Voici un exemple de variable :

```
{"interface_name": "GigabitEthernet1/1",
 "status": "up",
 "ip_address": "192.168.1.1",
 "netmask": "255.255.255.0"}
```

Il y a ici 4 variables, des variables contiennent et stockent des valeurs. Par exemple « Interface_Name » est un conteneur qui contient la valeur « gigabitethernet1/1 ».

Voyons comment l'échange de données devrait marcher sans utiliser la sérialisation standard des données.



Ci dessous l'application envoie une requête GET vers le controller, le controller réceptionne la demande par l'API et répond à sa requête.

Pour que les deux puissent communiquer (l'appli et le contrôleur) et se comprendre, le langage avec la sérialisation des données est utilisé. Le client envoie sa requête et est convertie par l'API en un format standard JSON puis est envoyé sur le réseau du contrôleur.

JSON (JavaScript Object Notation) est un standard open en format de fichier et un format de données interchangeable qui utilise un texte lisible par l'humain qui stocke et transmet les données des objets. JSON a été standardisé dans le RFC 8259. JSON a été dérivé de JavaScript mais est un langage indépendant et plusieurs langages de programmation modernes sont capables de générer et lire des données JSON. REST API utilise souvent JSON. Les espaces ne sont pas significatifs dans JSON ils ne changent pas le sens des données. Il y a quatre types de données primitif : string, nombre, booléen, nul. Il y a aussi deux type de données structurés : objets et tableau.

- Un string est une valeur dans un texte, il est entouré par des double guillemets « ».

Par exemple « Hello » est un string.

- Un nombre est une valeur numérique et n'est par entouré par des doubles guillemets.

Par exemple 5, 10 sont des nombres.

- Un booléen est un type de données qui a seulement deux valeurs possibles, et non entouré par des guillemets : true et false (Vrai ou Faux) et écris en minuscule.

- Une valeur nul est l'absence intentionnel de n'importe quelle objet de valeur et n'est pas entouré de guillemets.

Le type structuré des données JSON est ainsi :

Un objet est une liste non ordonnée par une pair de valeur clés (Les variables).

Les objets sont entourés par des accolades ({}). La clé est un string. La valeur est n'importe quelle type de données valide en JSON (string, nombre, booléen, nul, objet, tableau).

La clé et la valeur sont séparés par des deux point « : »

S'il y a plusieurs pairs de valeurs clés, chaque pair est séparé par une virgule.

Voici un exemple en bleue la variable et en rouge la valeur associé :

```
{
 "interface": "GigabitEthernet1/1",
 "is_up": true,
 "ipaddress": "192.168.1.1",
 "netmask": "255.255.255.0",
 "speed": 1000
}
```

Il est aussi possible d'écrire de la manière suivante car les espaces ne comptent pas :

```
{"interface":"GigabitEthernet1/1","is_
up":true,"ipaddress":"192.168.1.1","ne
tmask":"255.255.255.0","speed":1000}
```

Comme on peut le voir dans l'exemple suivant, les objets sont des types de données valides d'un pair de valeur clé :

```
{
    "device": {
       "name": "R1",
       "vendor": "Cisco",
       "model": "1101"
    },

    "interface_config": {
       "interface_name": "GigabitEthernet1/1",
       "is_up": true,
       "ipaddress": "192.168.1.1",
       "netmask": "255.255.255.0",
       "speed": 1000
    }
}
```

Ici la clé est « device » et « interface_config ». La valeur ou objet est contenu dans les accolades.

Un tableau ou « array » est une série de valeurs séparés par une virgule.

Il n'y a pas de valeur clé, c'est seulement une série de valeurs. Il n'est pas nécessaire que les valeurs soient du même type de données.

En voici un exemple :

```
{
    "interfaces": [
        "GigabitEthernet1/1",
        "GigabitEthernet1/2",
        "GigabitEthernet1/3"
    ],

    "random_values": [
        "Hi",
        5
    ]
}
```

Voici un résultat de la commande : « show interface brief » sur un Routeur :

```
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         192.168.1.1     YES manual up                    up
GigabitEthernet0/1         unassigned      YES unset  administratively down down
```

Voici le résultat de la même commande au format JSON :


```
{
    "ip_interfaces": [
            {
            "Interface":"GigabitEthernet0/0",
            "IP-Address": "192.168.1.1",
            "OK?": "YES",
            "Method": "manual",
            "Status": "up",
            "Protocol": "up"
            },
            {
            "Interface":"GigabitEthernet0/1",
            "IP-Address": "unassigned",
            "OK?": "YES",
            "Method": "unset",
            "Status": "administratively down",
            "Protocol": "down"
            }
    ]
}
```


Certaines fois un objet est appelé un dictionnaire.

XML est l'acronyme de (Extensible Markup Language) il a été développé comme langage de Markup, mais est maintenant utilisé comme langage général de sérialisation de données.

Les langages Markup (Par exemple HTML) sont utilisé pour formater le texte (Le font, la taille, la couleur, l'entête, etc...) XML est en général moins facilement lisible pour les humain par rapport à JSON. Les espaces ne sont pas significatifs et ne changent pas le sens du texte. XML est souvent utilisé par des REST API. Le format général qui est utilisé est : <key>valeur</key> par exemple :


```
R1#show ip interface brief | format
<?xml version="1.0" encoding="UTF-8"?>
  <ShowIpInterfaceBrief xmlns="ODM://built-in//show_ip_interface_brief">
    <SpecVersion>built-in</SpecVersion>
    <IPInterfaces>
      <entry>
        <Interface>GigabitEthernet0/0</Interface>
        <IP-Address>192.168.1.1</IP-Address>
        <OK>YES</OK>
        <Method>manual</Method>
        <Status>up</Status>
        <Protocol>up</Protocol>
      </entry>
      <entry>
        <Interface>GigabitEthernet0/1</Interface>
        <OK>YES</OK>
        <Method>unset</Method>
        <Status>administratively down</Status>
        <Protocol>down</Protocol>
      </entry>
    </IPInterfaces>
  </ShowIpInterfaceBrief>
```


on peut comparer les deux format de langage directement depuis la ligne de commandes :

```
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         192.168.1.1     YES manual up                    up
GigabitEthernet0/1         unassigned      YES unset  administratively down down


R1#show ip interface brief | format
<?xml version="1.0" encoding="UTF-8"?>
  <ShowIpInterfaceBrief xmlns="ODM://built-in//show_ip_interface_brief">
    <SpecVersion>built-in</SpecVersion>
    <IPInterfaces>
      <entry>
        <Interface>GigabitEthernet0/0</Interface>
        <IP-Address>192.168.1.1</IP-Address>
        <OK>YES</OK>
        <Method>manual</Method>
        <Status>up</Status>
        <Protocol>up</Protocol>
      </entry>
      <entry>
        <Interface>GigabitEthernet0/1</Interface>
        <OK>YES</OK>
        <Method>unset</Method>
        <Status>administratively down</Status>
        <Protocol>down</Protocol>
      </entry>
    </IPInterfaces>
  </ShowIpInterfaceBrief>
```

YAML signifie à l'origine « Yet Another Markup Language », mais n'est pas différents du langage de sérialisation plutôt qu'un langage de Markup, il a été renommé : YAML Ain't Markup Language.

YAML est utilisé par le réseau pour l'automatisation du réseau avec l'outil Ansible.

YAML est facilement lisible par l'humain. Les espace sont signifiant, c'est à dire qu'ils changent l'interprétation du langage, l'indentation est donc très importante.

Les fichier YAML commencent par des ---

Un seule – est utilisé pour indiquer une liste.

Les valeurs et clefs sont représentés au format key :valeur

```yaml
---
ip_interfaces:
- Interface: GigabitEthernet0/0
  IP-Address: 192.168.1.1
  OK?: 'YES'
  Method: manual
  Status: up
  Protocol: up
- Interface: GigabitEthernet0/1
  IP-Address: unassigned
  OK?: 'YES'
  Method: unset
  Status: administratively down
  Protocol: down
```

Voici une comparaison de texte entre JSON et YAML :

JSON

```json
{
    "ip_interfaces": [
            {
            "Interface":"GigabitEthernet0/0",
            "IP-Address": "192.168.1.1",
            "OK?": "YES",
            "Method": "manual",
            "Status": "up",
            "Protocol": "up"
            },
            {
            "Interface":"GigabitEthernet0/1",
            "IP-Address": "unassigned",
            "OK?": "YES",
            "Method": "unset",
            "Status": "administratively down",
            "Protocol": "down"
            }
    ]
}
```

YAML

```yaml
---
ip_interfaces:
- Interface: GigabitEthernet0/0
  IP-Address: 192.168.1.1
  OK?: 'YES'
  Method: manual
  Status: up
  Protocol: up
- Interface: GigabitEthernet0/1
  IP-Address: unassigned
  OK?: 'YES'
  Method: unset
  Status: administratively down
  Protocol: down
```

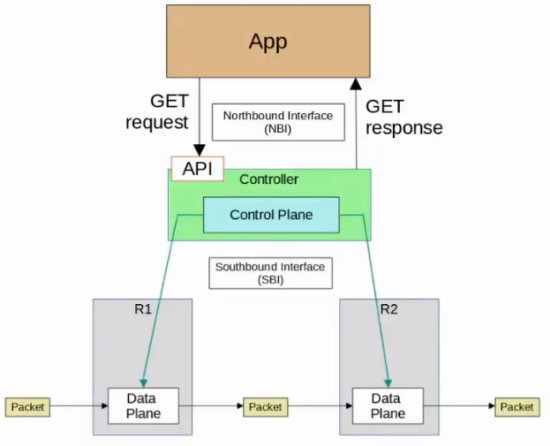
Cours 61 : REST APIs

Dans ce cours nous verrons REST APIs. Les APIs sont utilisés pour permettre aux programmes d'échanger des informations entre eux, ils sont important pour l'automatisation du réseau car ils permettent aux programmes de facilement interagir avec les appareils et le contrôleur.

Les REST APIs sont spécifiquement utilisé pour le Northbound Interface dans une architecture SDN. Nous ferons d'abord un aperçu des API. Les opérations CRUD et les verbes HTTP.

Nous verrons ce qu'est un REST API et ses caractéristiques et en dernier temps nous utiliserons un Cisco Devnet.

Une API (Application Programming Interface) est une interface logiciel qui permet à deux applications de communiquer entre elles. Les APIs sont essentiels pas seulement pour l'automatisation mais pour tout type d'applications.



Dans une architecture SDN, les APIs sont utilisés pour communiquer entre les applications et le contrôleur SDN (Par le NBI), et entre le contrôleur SDN et les appareils du réseau (par le SBI).

Le NBI utilise les REST APIs. NETCONF et RESTCONF sont des APIs populaire southbound.

CRUD (Create, Read, Update, Delete) se réfère aux opérations qui fonctionnent en utilisant REST APIs.

- Les opérations Create sont utilisés pour créer de nouvelles variables et afin de placer leur valeur initial. Par exemple créer une variable « ip_address » et placer la valeur à « 10.1.1.1 »

- Les opération Read sont utilisés pour obtenir la valeur de la variable. Par exemple quelle est la valeur de la variable « ip_address » ?

- Les opérations Update sont utilisés pour changer la valeur d'une variable. Par exemple changer la valeur de la variable « ip_address » pour « 10.2.3.4 »

- Les opérations Delete sont utilise pour supprimer des variables. Par exemple supprimer la variable « ip_address »

HTTP utilise des verbes ou méthodes pour cartographier les opérations de CRUD.

REST APIs utilise HTTP. Voici un tableau qui permet de résumer les verbes HTTP :

Intérêt	Opération CRUD	Verbe HTTP
Créer une nouvelle variable	Create	POST
Obtenir la valeur d'une variable	Read	GET
Changer la valeur d'une variable	Update	PUT, PATCH
Supprimer une variable	Delete	DELETE

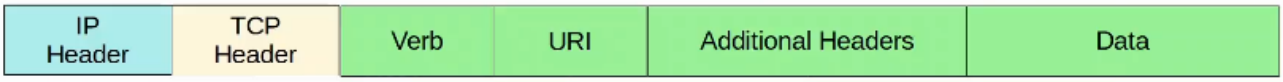
Voyons plus en détail comment ces verbes sont utilisés. Lorsqu'un client HTTP envoie une requête vers un serveur HTTP, l'entête HTTP inclus des informations comme :

- Le verbe HTTP (par exemple GET)

- Une URI (Uniform Resource Identifier) qui indique la ressource auquel il essaie d'accéder.

Par exemple le client envoie un message GET afin de demander au serveur de récupérer l'information de la variable contenu dans URI A.

La requête HTTP peut inclure des entêtes supplémentaires qui passent des informations supplémentaire au serveur.



Un exemple pourrait être une entête Accept qui informe le serveur à propos du type de donnée qui peut être envoyé en retour au client.

Par exemple le lien peut accepter des applications json ou applications XML avec cette écriture :

Accept : application/json ou *Accept : application/xml*

Lorsque le client REST fait un appelle API (requête) vers un serveur REST, il envoie une requête HTTP tout comme l'écriture précédente.

Les REST APIs n'ont pas à utiliser HTTP pour la communication, HTTP est le choix le plus commun.

La réponse serveur inclus un code de statut qui indique si la requête à réussi ou a échoué, avec d'autres détails.

Le premier chiffre indique la classe de la réponse :

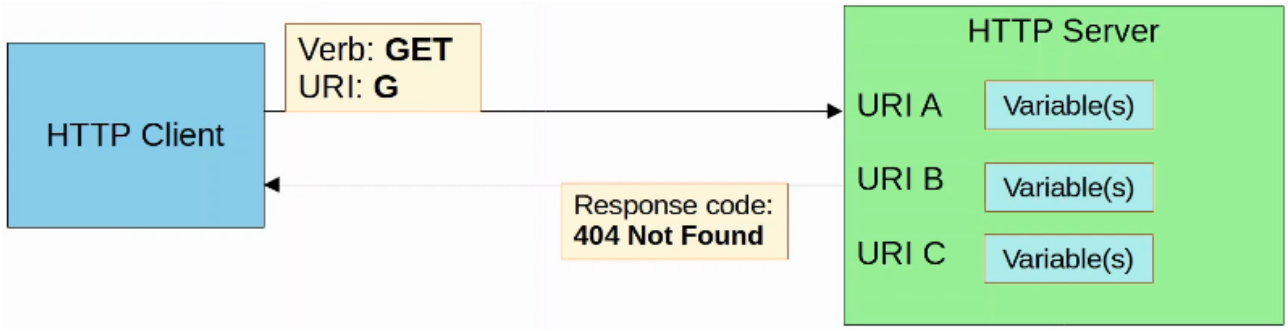
- 1xx informationnel – la requête est reçu, et continue de progresser.

- 2xx succès – la requête a été réussi et bien reçu, comprise et accepté.

- 3xx redirection – une action est nécessaire afin de compléter la requête

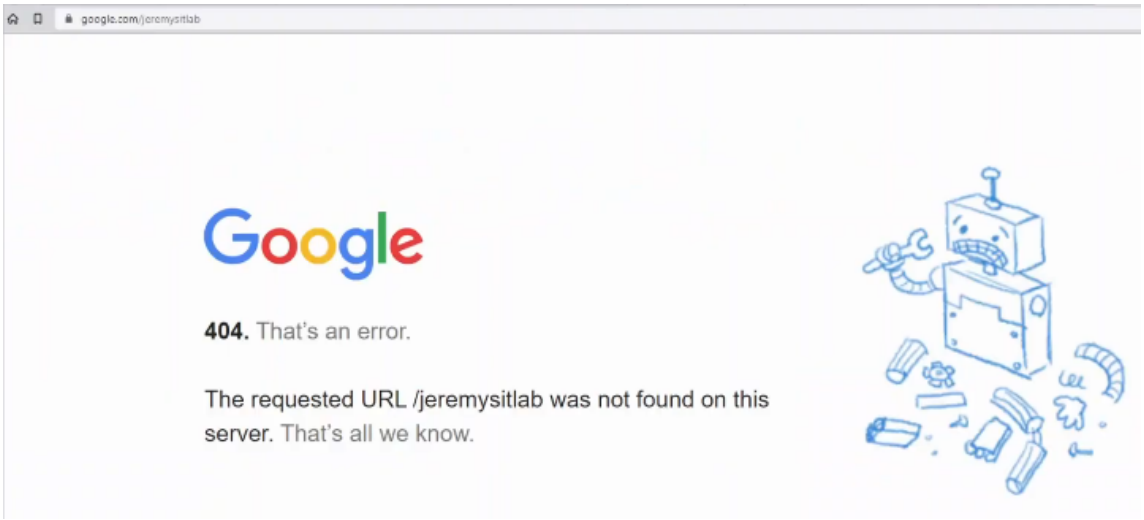
- 4xx erreur client – la requête contient une mauvaise syntaxe ou ne peut pas être satisfaite

- 5xx erreur serveur – le serveur n'a pas réussi à répondre pour terminer une requête valide



Voici un exemple dans lequel le client HTTP envoie une requête GET et où le serveur HTTP rpond avec un code d'erreur 404 signifiant qu'il y a erreur de syntaxe.

Un exemple de page 404 peut être affiché sur Google lorsque une erreur dans la syntaxe du site :



Voici quelques exemple de chaque classe de réponse HTTP :

- 1xx informationnel

102 Processing : indique que le serveur a reçu la requête et entrain de la traiter, mais la réponse n'est pour le moment pas disponible.

- 2xx Succès

200 OK : indique que la requête a réussi

201 Created : indique que la requête a réussi et que la nouvelle ressource a été crée (en réponse au POST)

- 3xx Redirection

- 301 Moved Permanently indique que la ressource de la requête a été déplacé, et que le serveur indique sa nouvelle localisation.

- 4xx erreur client

403 unauthorized signifie que le client doit s'authentifier pour avoir une réponse.

404 Not Found signifie que la ressource de la requête n'a pas été trouvé.

- 5xx erreur Serveur

500 Erreur serveur Interne signifie que le serveur à rencontrer quelque chose d'imprévu qui ne sais pas comment gérer

REST est l'acronyme de Representational State Transfer. REST APIs sont aussi connu comme REST-based APIs ou RESTful APIs. REST n'est pas spécifique à une API. Il peut décrire plusieurs règles à propos de comment l'API devrait fonctionner.

Il y a 6 contraintes des architecture RESTful :

- Uniform Interface

- Client server

- Stateless

- Cacheable ou non-cacheable

- Layered system

- Code on demand (optionnel)

Nous allons en voir quelques une en détail :

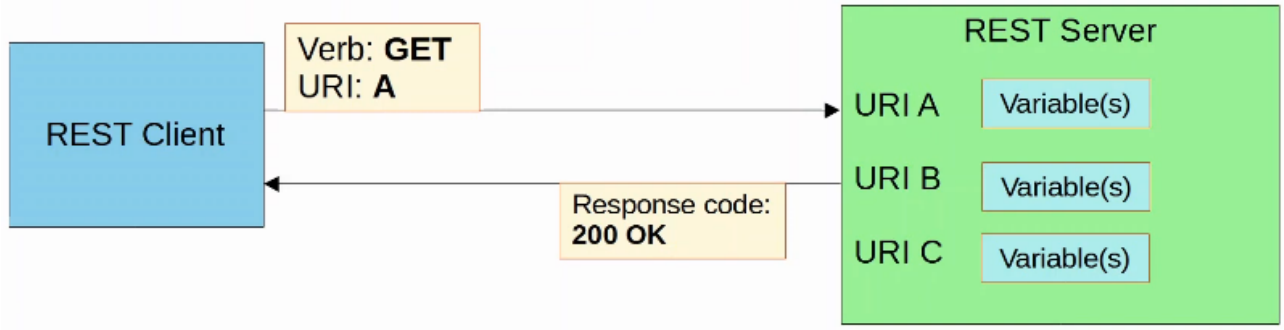
- Client-Server :

Les REST APIs utilisent des architectures client-server

Le client utilise des appelles API (requêtes HTTP) pour accéder aux ressources sur le serveur.

La séparation entre le client et le serveur signifie qu'ils peuvent tout deux changer et évoluer indépendamment l'un de l'autre. Lorsque l'application client change ou que le serveur de l'application change, l'interface entre eux ne doit pas être cassé.

Voici un exemple de client-serveur :



- Stateless :

Les échanges REST APIs neutres.

Cela signifie que chaque échange API est un évènement séparé, indépendant de tout les échanges passés entre le client et le serveur.

Le serveur ne stocke pas d'information à propos des requêtes depuis le client pour déterminer comment il devrait répondre aux nouvelles requêtes.

Si une authentification est requise cela signifie que le client doit s'authentifier avec le serveur pour chaque requête faite. TCP est un exemple de protocole stateful. UDP est un exemple de protocole Stateless. Bien que REST API utilise HTTP, qui utilise TCP (stateful) avec la couche 4 du protocole, HTTP et REST APIs eux même ne sont pas stateful.

Les fonctions de chaque couche sont séparés.

- Cacheable ou non-cacheable

REST API doit supporter le chache de données

Le cache se réfère au stockage de données pour une utilisation future.

Par exemple, l'ordinateur peut cacher plusieurs éléments d'une page web donc il n'a pas à récupérer la page entière chaque fois que l'on la visite.

Cela permet d'améliorer les performance du client et de réduire le chargement du serveur.

Pas toutes les ressources doivent être en cache, mais les ressources cachés doivent être déclarés comme cachés.

Pour plus d'informations sur les autres type d'architecture RESTful il est possible de visiter ce site :

`https://restfulapi.net/rest-architectural-constraints/`

Pour que les applications puissent communiquer à travers le réseau, les protocoles réseau doivent être facilement utilisable pour faciliter leurs communications.
Pour REST APIs, HTTP(S) est le choix le plus commun.

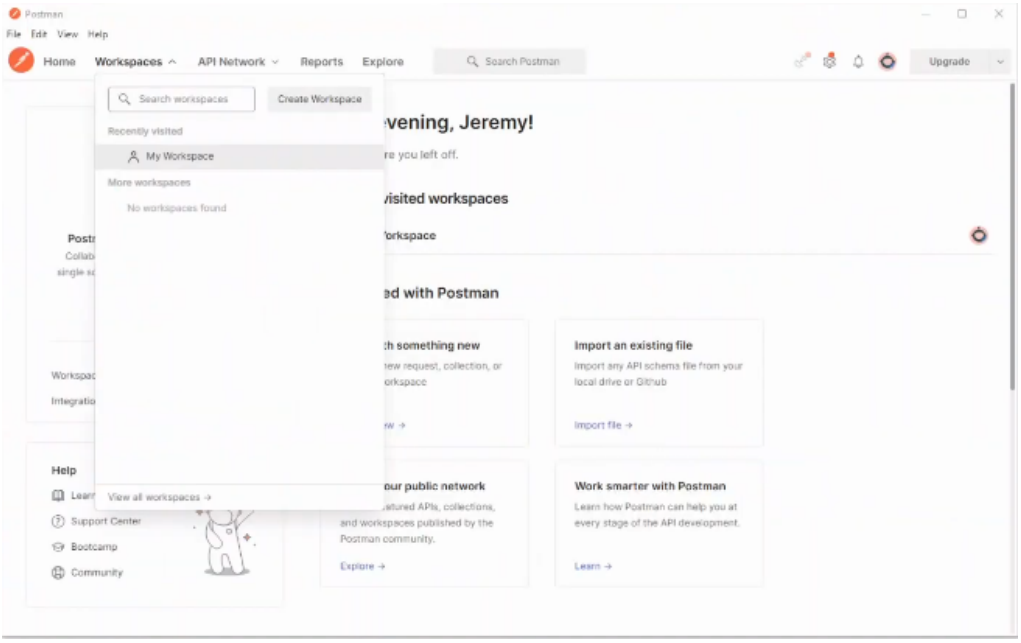
Cisco Devnet est un programme développé par Cisco pour aider les développeurs et professionnels en IT qui veulent écrire leurs applications et développer leur intégration avec des produits Cisco, des plateformes et des APIs.

DevNet offre un tas de ressources gratuites comme des cours, des tutoriels, labs, sandboxes, documentation, etc. pour apprendre à propos de l'automatisation et développer ses compétences.

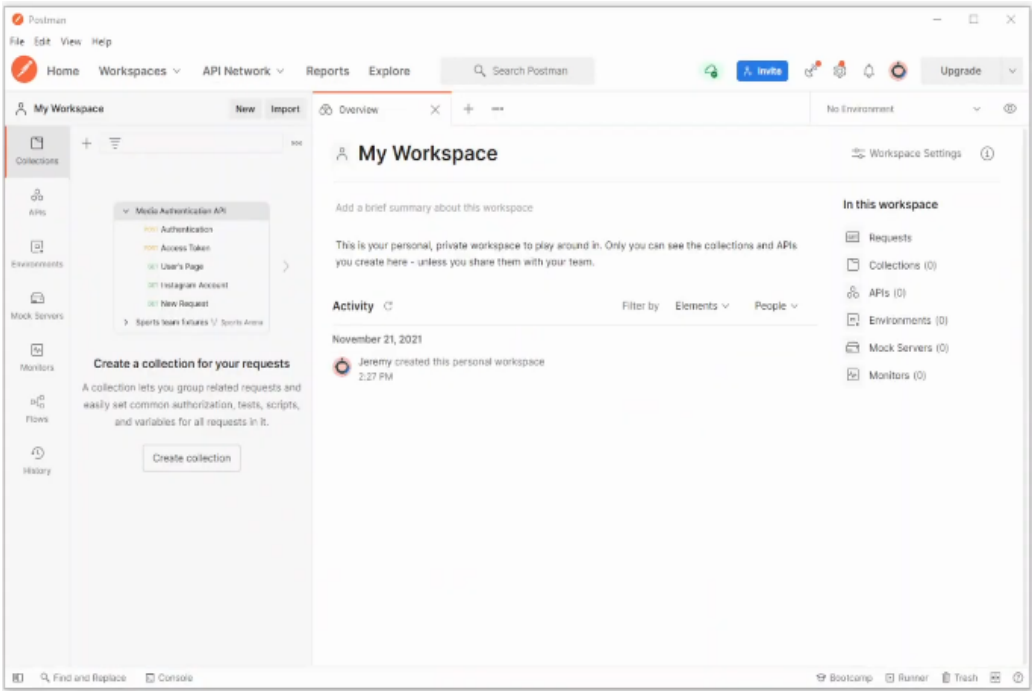
Il existe aussi des certifications DevNet que l'on peut suivre si l'on est intéressé par l'automatisation. Nous utiliserons Cisco DNA Center Sandbox pour envoyer un appelle REST API en utilisant Postman. Un centre DNA est l'un des contrôleur Cisco SDN.

Postman est une plateforme pour construire et utiliser des APIs.

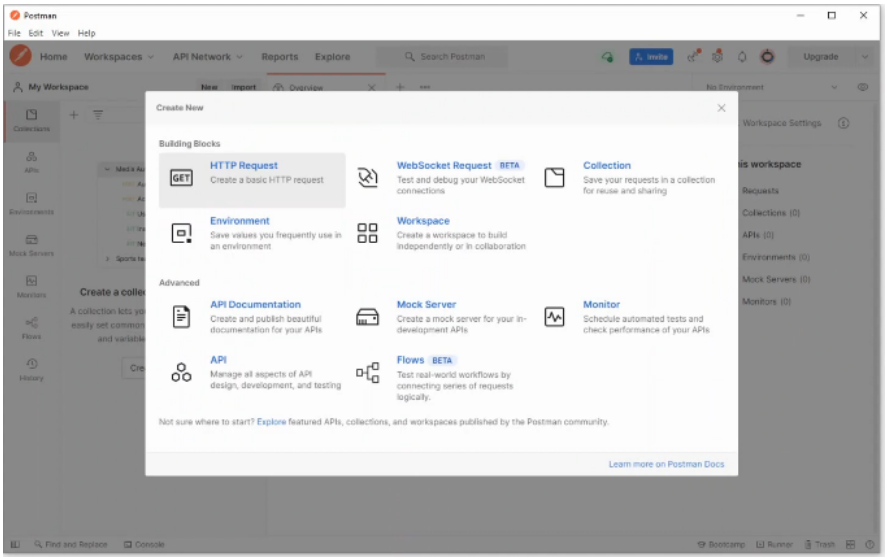
Voici l'interface :



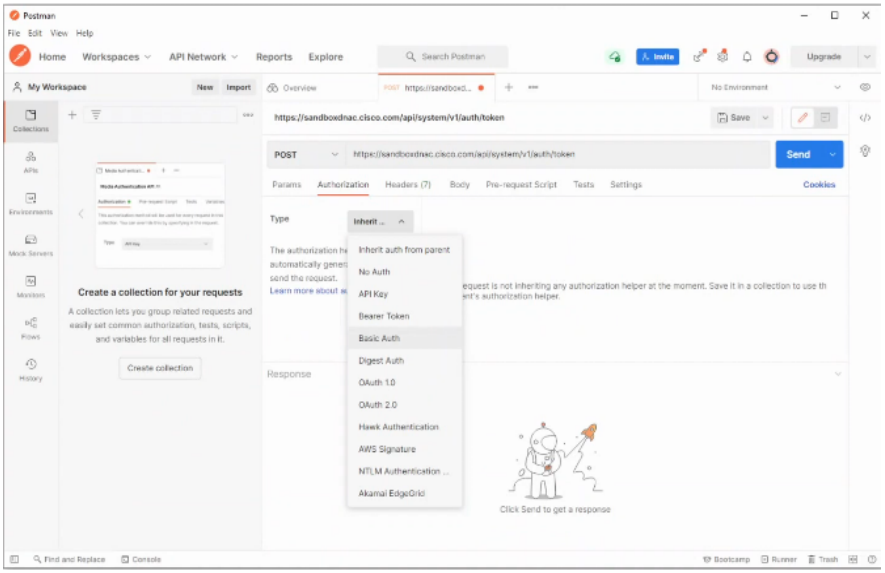
Postman se présente comme suit :



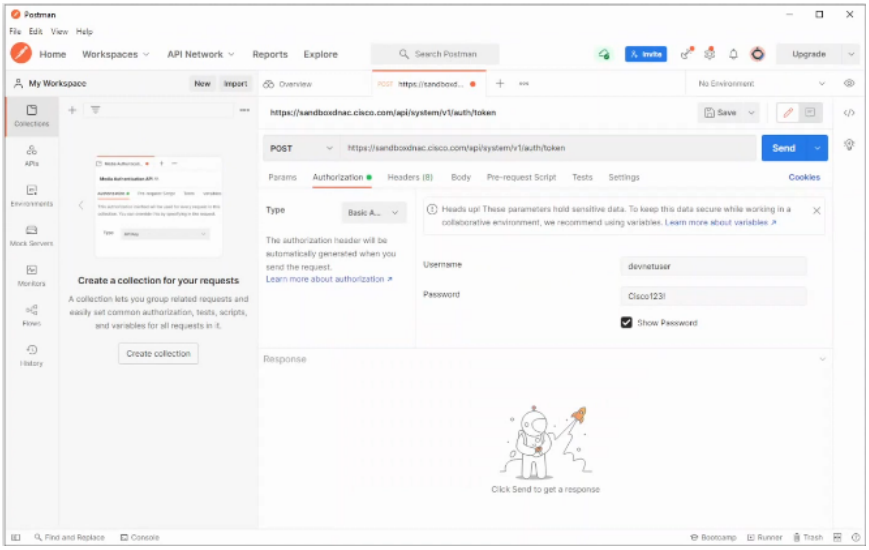
On peut créer différents éléments :



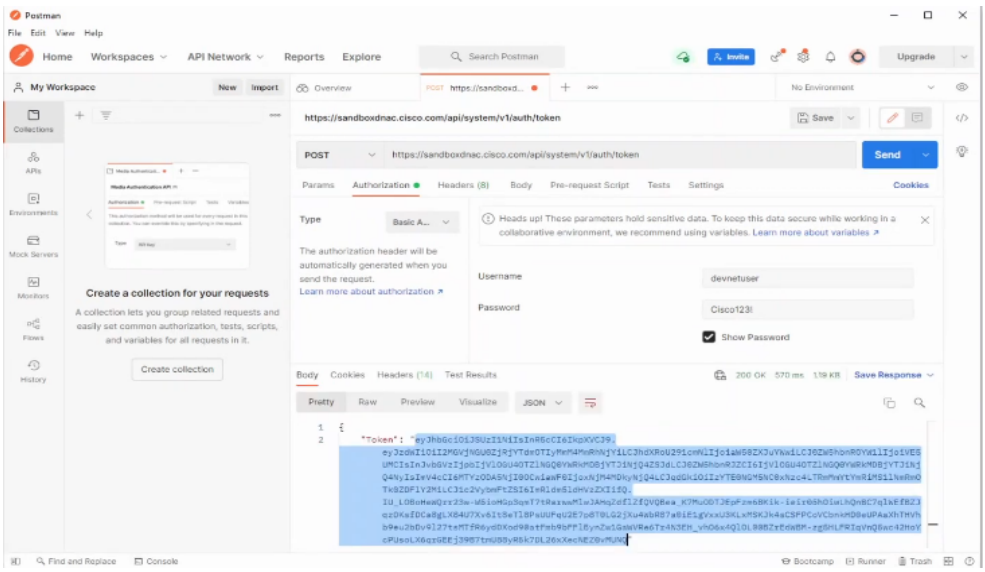
On crée une requête HTTP :



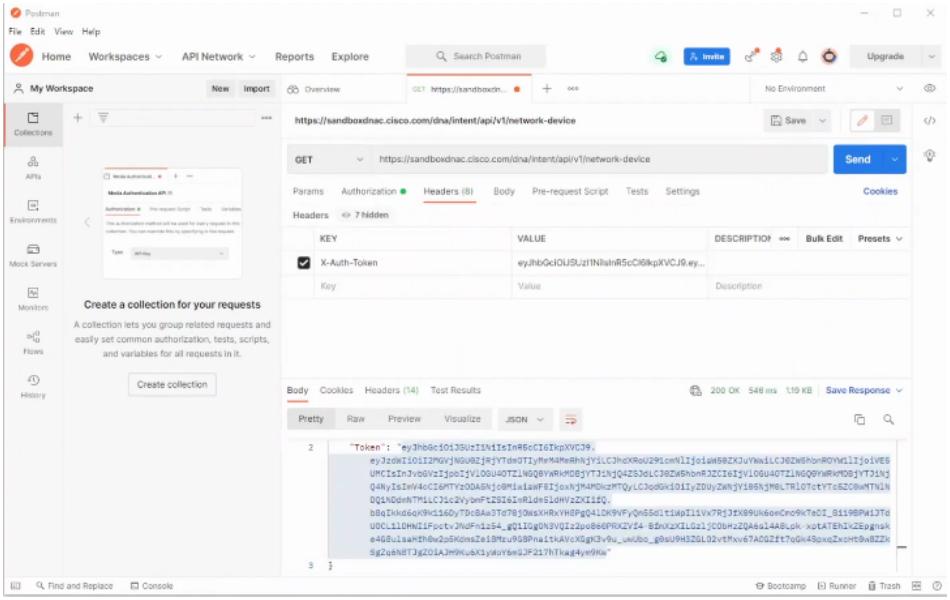
On crée le nom d'utilisateur et le mot de passe et on voie la requête avec « Send » :



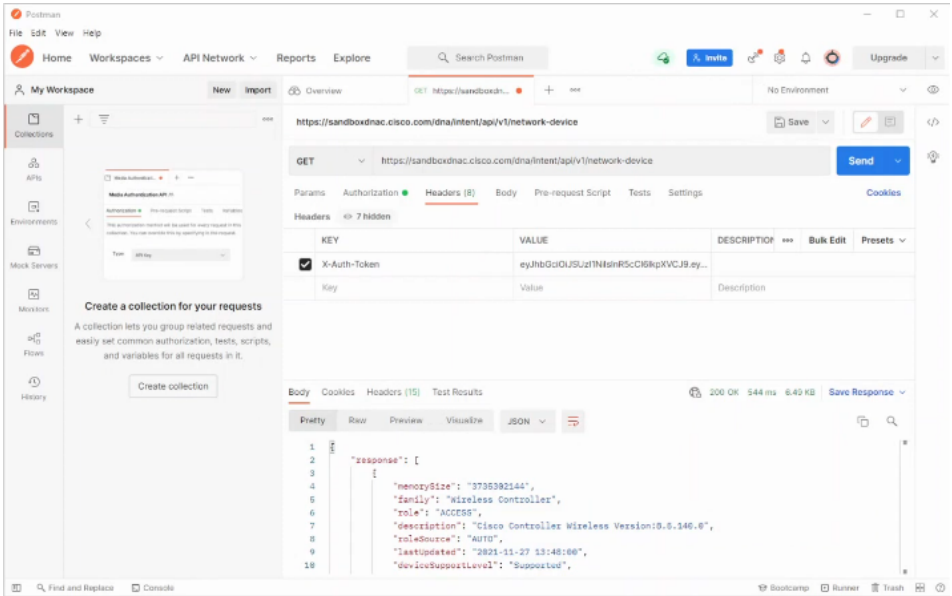
La requête a ici réussi puisqu'il y a un code de réponse « OK » :



Pour envoyer une requête GET on modifie la bannière à envoyer :



La requête a réussi puisqu'il y a un code de réponse OK :



Voici la réponse plus détaillé :

```
{
  "memorySize": "NA",
  "family": "Switches and Hubs",
  "role": "ACCESS",
  "roleSource": "AUTO",
  "lastUpdated": "2021-11-27 13:50:20",
  "deviceSupportLevel": "Supported",
  "softwareType": "IOS-XE",
  "softwareVersion": "17.3.3",
  "macAddress": "84:8a:8d:05:76:00",
  "collectionInterval": "Global Default",
  "inventoryStatusDetail": "<status><general
code=\"SUCCESS\"/></status>",
  "serialNumber": "FCW2220G09V",
  "lastUpdateTime": 1638021020343,
  "hostname": "leaf1.abc.inc",
  "tagCount": "0",
  "tunnelUdpPort": null,
  "uptimeSeconds": 2648950,
  "waasDeviceMode": null,
  "apManagerInterfaceIp": "",
  "bootDateTime": "2021-10-28 18:10:20",
  "collectionStatus": "Managed",
  "locationName": null,
  "managementIpAddress": "10.10.20.81",
  "platformId": "C9300-24U",
  "reachabilityFailureReason": "",
  "reachabilityStatus": "Reachable",
  "series": "Cisco Catalyst 9300 Series Switches",
  "snmpContact": "",
  "snmpLocation": "",
  "upTime": "29 days, 19:40:48.91",
  "apEthernetMacAddress": null,
  "associatedWlcIp": "",
  "errorCode": null,
  "errorDescription": null,
  "interfaceCount": "0",
  "lineCardCount": "0",
  "lineCardId": "",
  "managedAtleastOnce": true,
  "location": null,
  "type": "Cisco Catalyst 9300 Switch",
  "managementState": "Managed",
  "instanceUuid": "aa0a5258-3e6f-422f-9c4e-9c196db115ae",
  "instanceTenantId": "5e8e896e4d4add00ca2b6487",
  "id": "aa0a5258-3e6f-422f-9c4e-9c196db115ae"
}
```


On peut voir différentes informations comme la « famille » d'appareil est Switch et Hubs.

Le nom d'hôte est « leaf1.abc.inc »

Le nom de modèle est avec platformID : « C9300-24U »

Cours 62 : Software-Defined Networking

Dans ce cours nous verrons Software-Define Networking (SDN). Nous verrons plus en détail les offres que Cisco propose comme SD-Access ou Software Defined Access.

Nous ferons en premier temps rappel du fonctionnement de SDN, puis verrons Cisco SD-Acces, le Cisco DNA Center, Une comparaison entre Gestion DNA Center Network et une gestion réseau traditionnel.

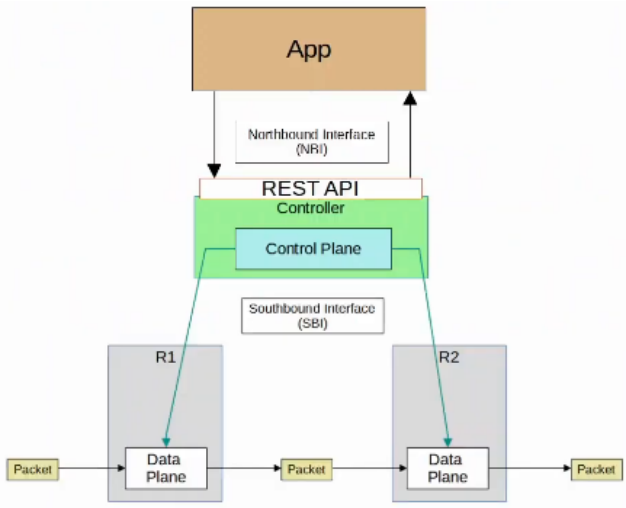
SDN est l'acronyme de Software-Define Networking il s'agit d'une approche du réseau pour centraliser le plan de contrôle dans une application appelé Contrôleur.

Les plans de contrôle traditionnel utilisent une architecture distribué c'est à dire que chaque appareil a son propre plan de contrôle. Le plan de contrôle de chaque appareil du réseau utilise des protocoles comme OSPF pour communiquer entre eux partager des informations de routage. Chaque appareil a ses propres ACL et règles de sécurité, etc...

Un contrôleur SDN centralise les fonctions du plan de contrôle comme le calcul des routes.

Le contrôleur peut interagir de manière programmé avec les appareils du réseau en utilisant des APIs. Le SBI est utilisé pour communiquer entre le contrôleur et les appareils du réseau qu'il contrôle. Le NBI est ce qui permet d'intéragir avec le contrôleur avec un script et l'application.

Voici à quoi ressemble une architecture SDN :



Ces 3 couches de l'architecture ont un nom, au dessus la couche application qui contient les scripts/applications qui disent au contrôleur SDN quelle comportement du réseau est désiré.

En deuxième niveau la couche de contrôle avec le REST API qui contient le contrôleur SDN qui reçoit et traite les instructions reçu par la couche application.

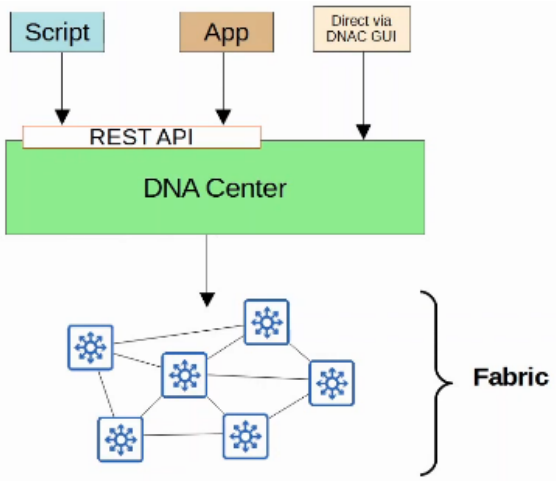
La dernière couche est la couche Infrastructure qui contient les appareils du réseau responsables de transmettre les messages sur le réseau Sur le schéma R1 et R2).

Cisco SD-Acces est une solution Cisco pour automatiser des LAN dans un campus.

ACI (Application Centric Infrastructure) est la solution SDN pour automatiser les centre de données d'un réseau. SD-WAN est la solution SDN pour automatiser des WAN.

Cisco DNA (Digital Network Architecture) Center est le contrôleur au centre du SD-Access.

Voyons une architecture SD-Access basique :



Pour comprendre la « fabrique » ou appareil du réseau, il faut tout d'abord comprendre les dessous du réseau (underlay) physique des appareil et de leurs connexions (Incluant connexion câblé et sans fil) qui fournit une connectivité IP (par exemple en utilisant IS-IS).

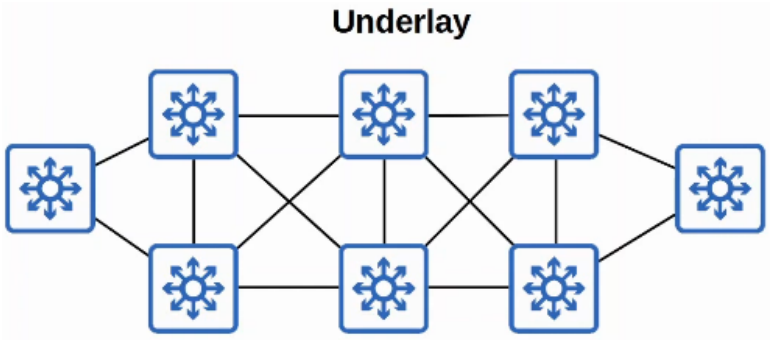
Le Multilayer est constitué des Switchs et de leurs connexions.

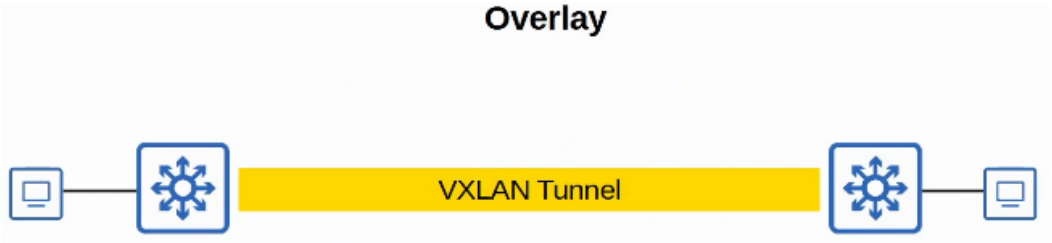
Le Overlay est le réseau virtuel construit au dessus du réseau physique underlay.

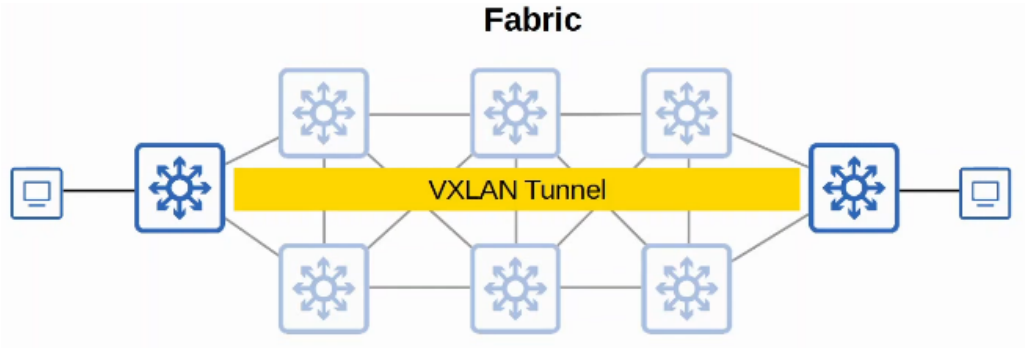
SD-Access utilise VXLAN (Virtual Extensible LAN) pour construire des tunnels.

La « fabrique » est la combinaison du overlay et du underlay, le réseau physique et virtuel tout entier.

Voici des exemples de ces concepts schématisé :







L'utilisation du underlay est conçu pour supporter les tunnels VXLAN du overlay.

Il y a trois différents rôles pour les Switchs dans le SD-Access :

- Edge nodes : connecte aux hôte finaux

- Border nodes : connecte aux appareils en dehors du domaine SD-Access par exemple le routeur WAN.

- Control nodes : utilise LISP (Locator ID Separation Protocol) pour faire fonctionner plusieurs fonctionnalités variés du plan de contrôle.

Il est possible d'ajouter SD-Access au dessus d'un réseau existant (brownfield deployment) si le matériel réseau et logiciel le supporte.

Dans ce cas DNA Center ne configure pas le underlay.

Un nouveau déploiement (greenfiel deployment) sera configuré par le DNA Center pour utiliser le SD-Access underlay :

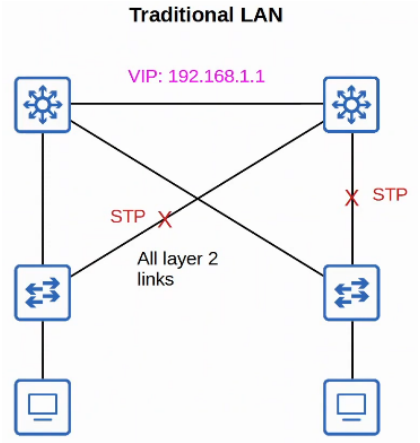
- Tous les Switchs sont de couche 3 et utilise IS-IS pour protocole de routage.

- Tout est lié entre les Switchs et les ports de routage. Cela signifie que STP n'est pas nécessaire.

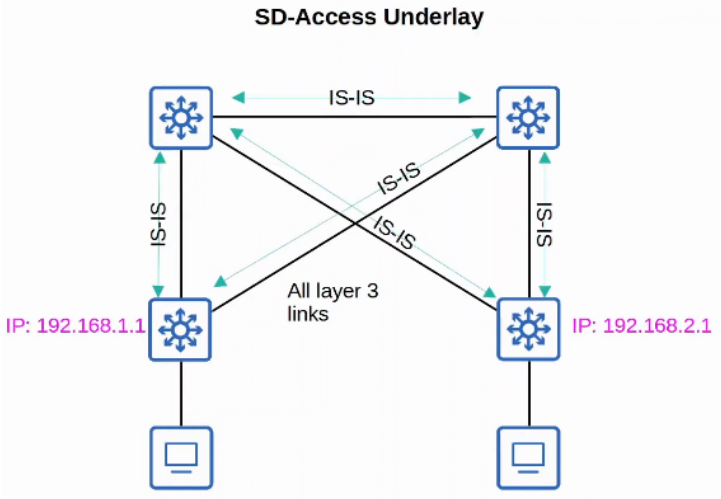
- Edge Nodes (Access Switch) fonctionnent comme passerelle par défaut de l'hôte final (couche de routage d'accès).

Un LAN Traditionnel se présente ainsi :

Pour envoyer des messages ils enverront vers le Virtual IP fournit par le FHRP (192.168.1.1) :



Dans un réseau SD-Access le fonctionnement est différent :



Les connexions entre Switch sont de couche 3 et IS-IS est utilisé pour échanger les informations.

Le SD-Access Overlay est un autre concept différent.

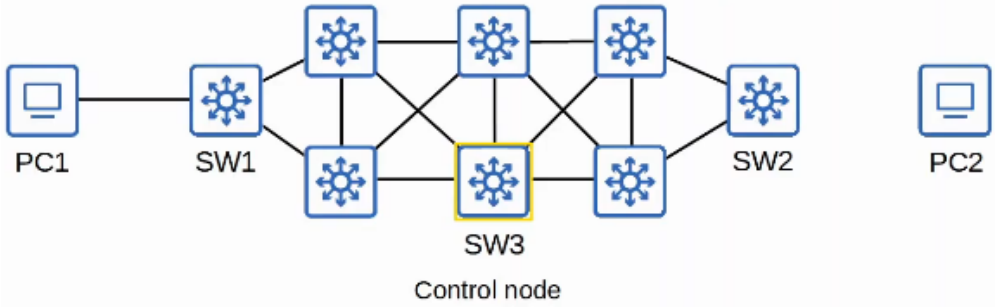
LISP fournit le plan de contrôle au SD-Access, une liste de cartographie de EID (Endpoint Identifiers) vers le RLOC (localisation du routage) est conservé.

Les EID identifient les hôtes finaux connectés aux Switchs pont, et RLOC identifie le Switch pont qui sera utilisé pour joindre l'hôte final.

Il y a bien d'autres détails à propos de LISP, mais il possible de voir en quoi il diffère d'un plan de contrôle traditionnel.

Cisco TrustSec (CTS) fournit un contrôle des politiques (QoS, politique de sécurité, etc.)

VXLAN fournit le plan de données du SD-Access.

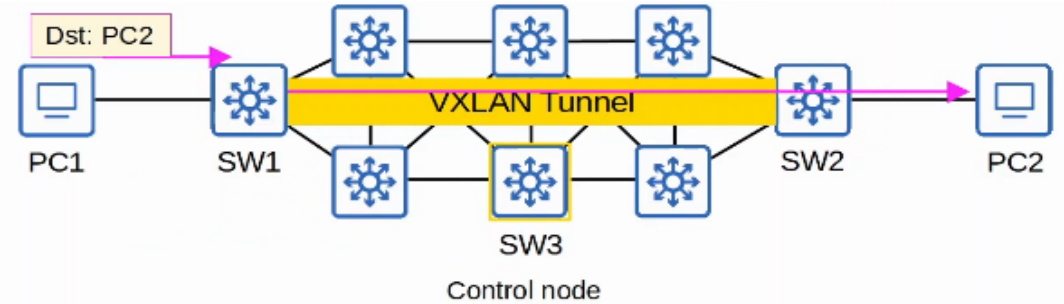


SW3 est le Control Node.

SW3 signal que PC2 est joignable par SW2.

PC1 souhaite joindre PC2, il envoie donc le message à sa passerelle par défaut (SW1), SW1 interroge SW3 de comment joindre PC2. Le SW3 lui répond que PC2 est joignable par SW2.

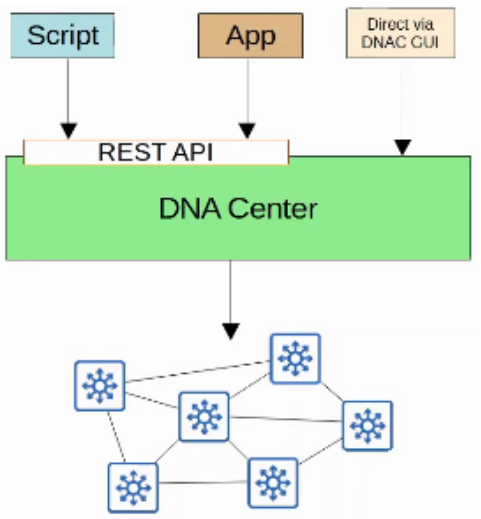
Le message de PC2 est donc transmis par un tunnel VXLAN entre SW1 et SW2.



Le DNA Center a deux rôles principaux :

- C'est le SDN controller utilisé dans le SD-Access

- Un gestionnaire réseau dans un réseau traditionnel (Non SD-Access)



Un DNA Center est une application installé sur du matériel de serveur Cisco UCS

- DNA Center a un REST API qui peut être utilisé pour interagir avec le DNA Center

- Le SBI supporte des protocoles comme NETCONF et RESTCONF (Tout comme des protocoles traditionnels comme Telnet, SSH, SNMP)

- Un DNA Center active Intent-Based Networking (IBN)

Le but est de permettre à l'ingénieur de communiquer avec leur intention du comportement au DNA Center, le DNA Center fait attention aux détails de la configuration actuelle et des politiques des appareils.

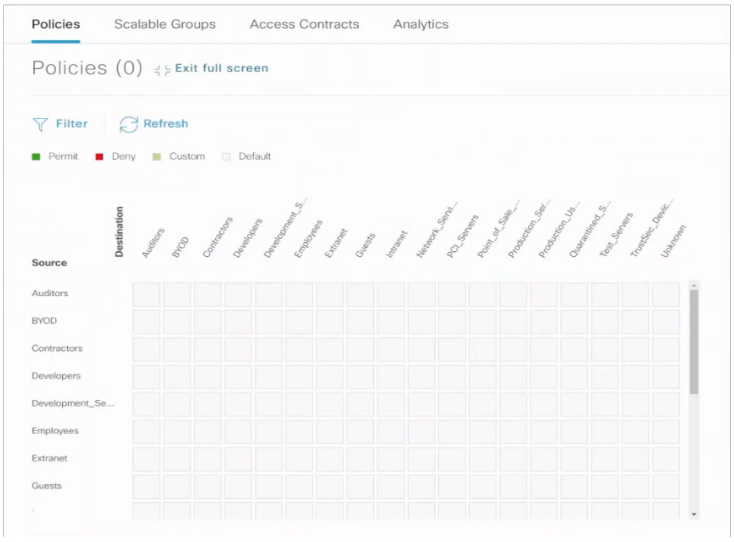
Les politique de sécurité traditionnel utilisant ACL peuvent devenir très encombrantes.

Les ACL peuvent avoir des milliers d'entrées. L'objectif des entrées est oublié avec le temps et un ingénieur qui part d'une entreprise est remplacé par un autre ne connaissant par les ACL établit par l'ancien ingénieur.

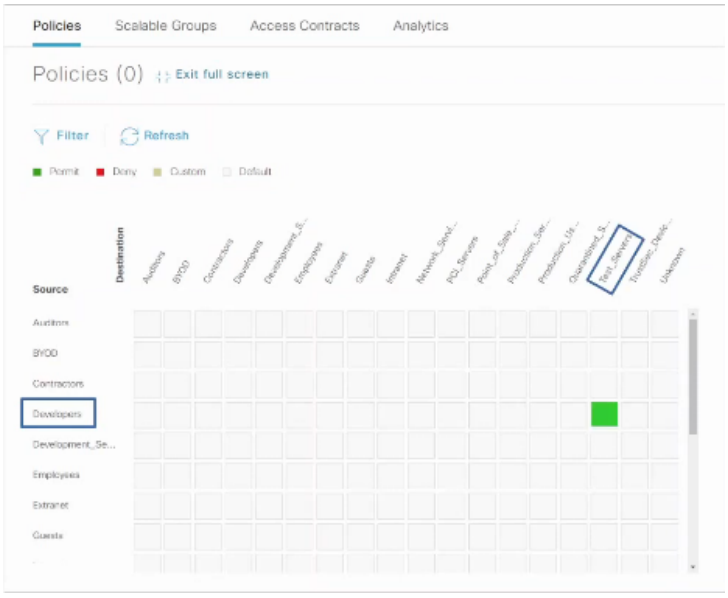
Configurer et appliquer des ACL correctement à travers le réseau est encombrant et peut poser problème en cas d'erreur.

Le DNA Center permet à l'ingénieur de spécifier l'objectif de la politique (Ce groupe d'utilisateurs ne peuvent pas communiquer avec ce groupe, ce groupe peut accéder à ce serveur mais pas à tous les serveurs, etc.) Le DNA Center fera attention aux détails exact de l'implémentation de la politique.

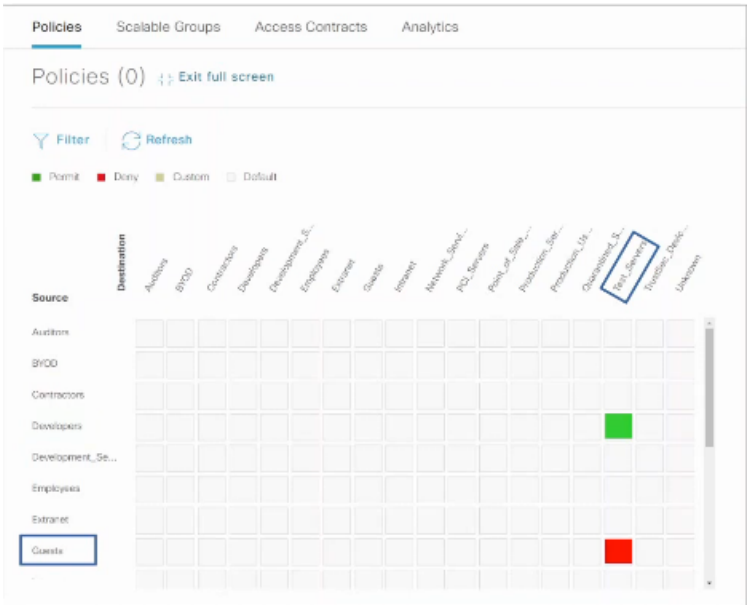
Voici à quoi ressemble de configurer des politiques dans un serveur DNA :



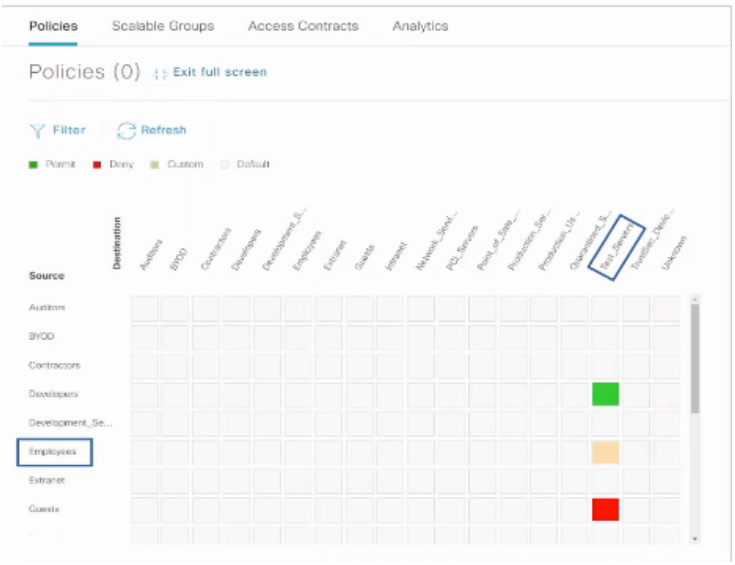
Si l'on veut configurer afin que les développeurs aient accès au Test_Servers on configure ainsi :



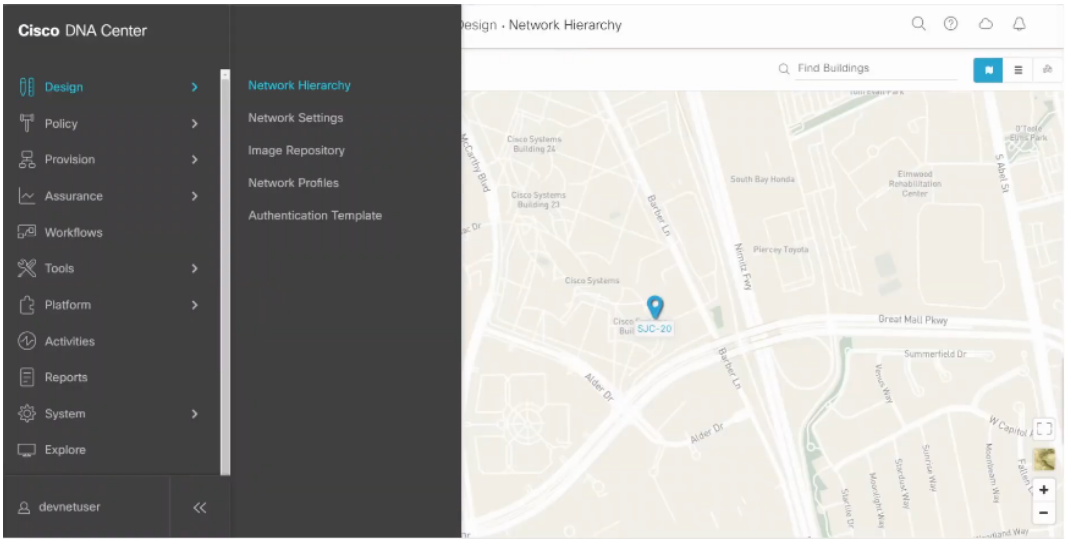
Si l'on veut bloquer le trafique des Guest pour Test_Servers on configure ainsi :



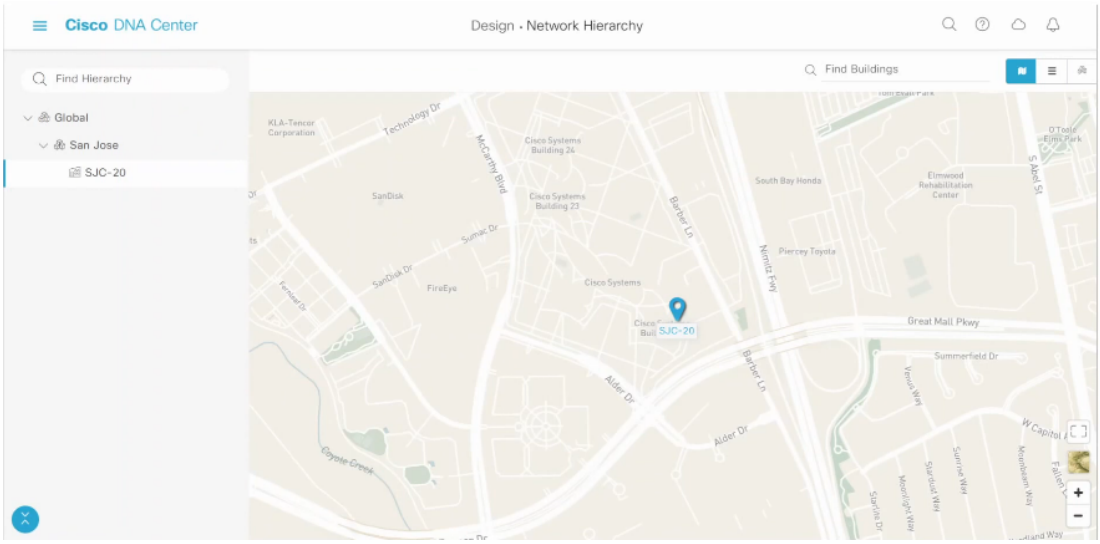
Pour personnaliser l'ACL employees et Test_Servers on configure ainsi :



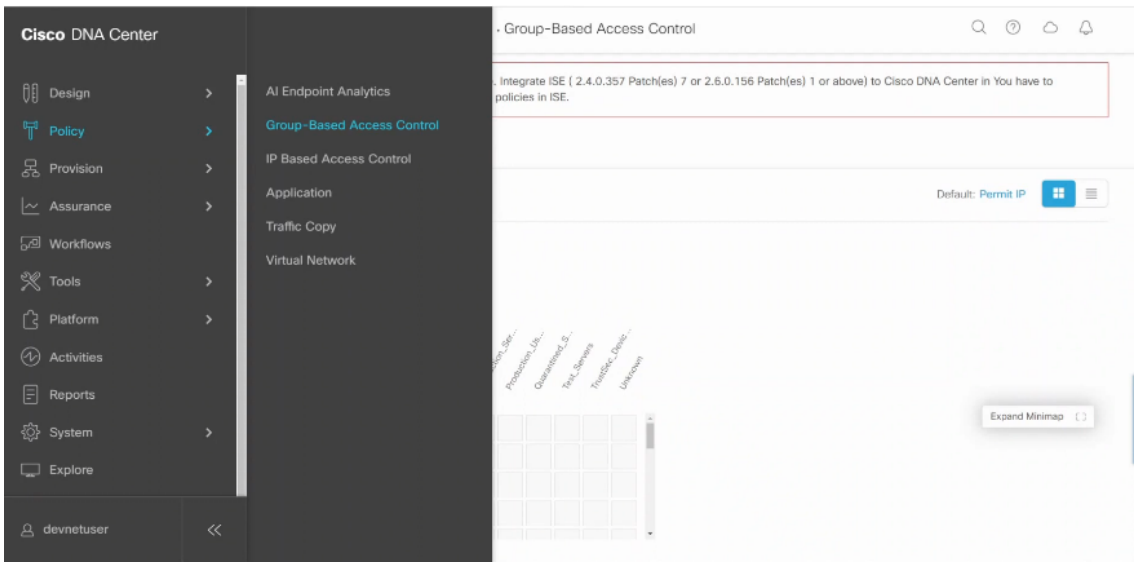
Voici à quoi ressemble le panel de connexion DNA Center :



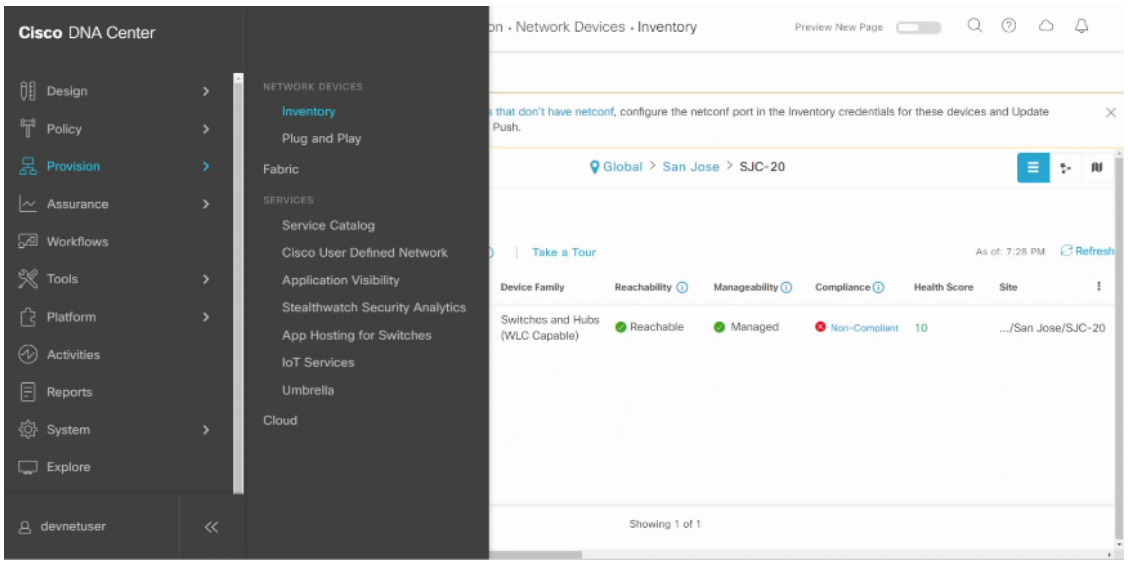
On peut cartographier le site de l'entreprise avec Cisco DNA Center :



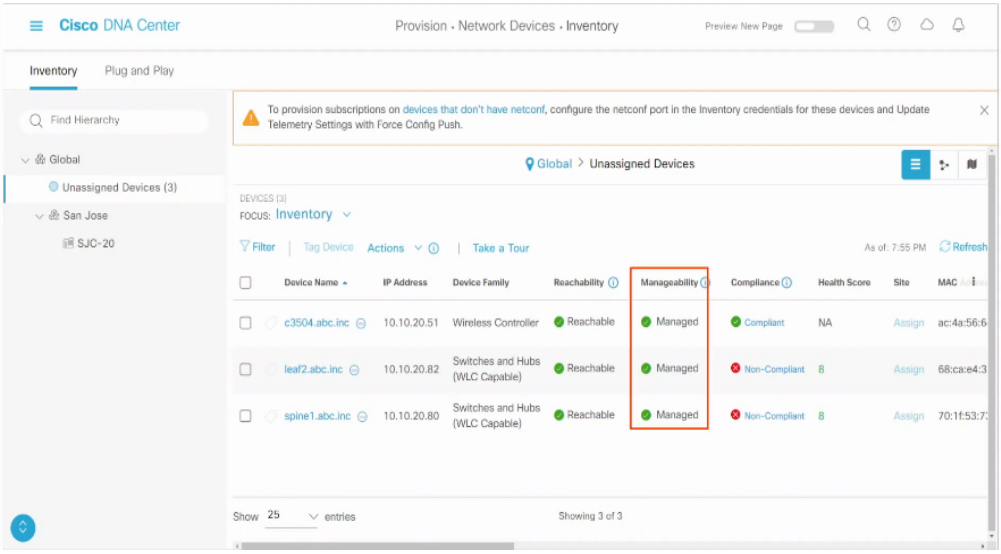
On configure la politique en accédant à Group-Based Access Control :



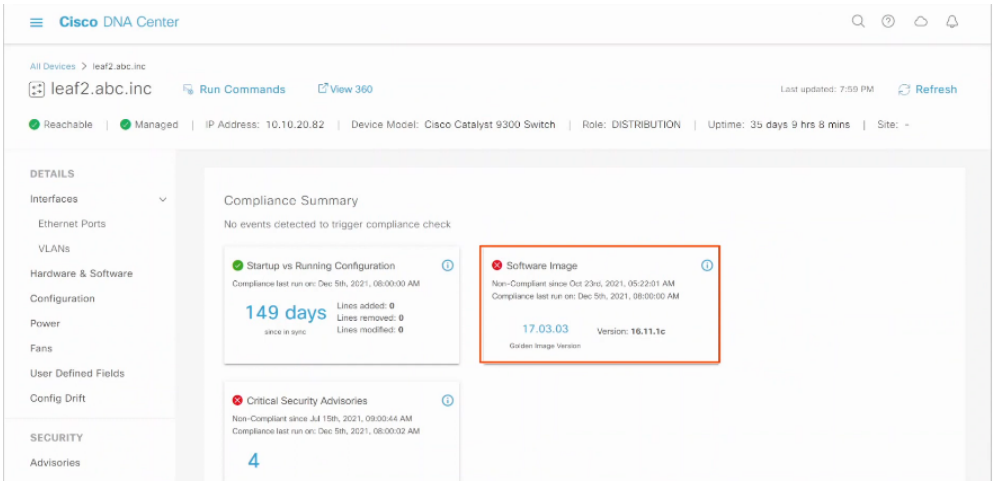
On peut accéder à l'inventaire des différents sites :



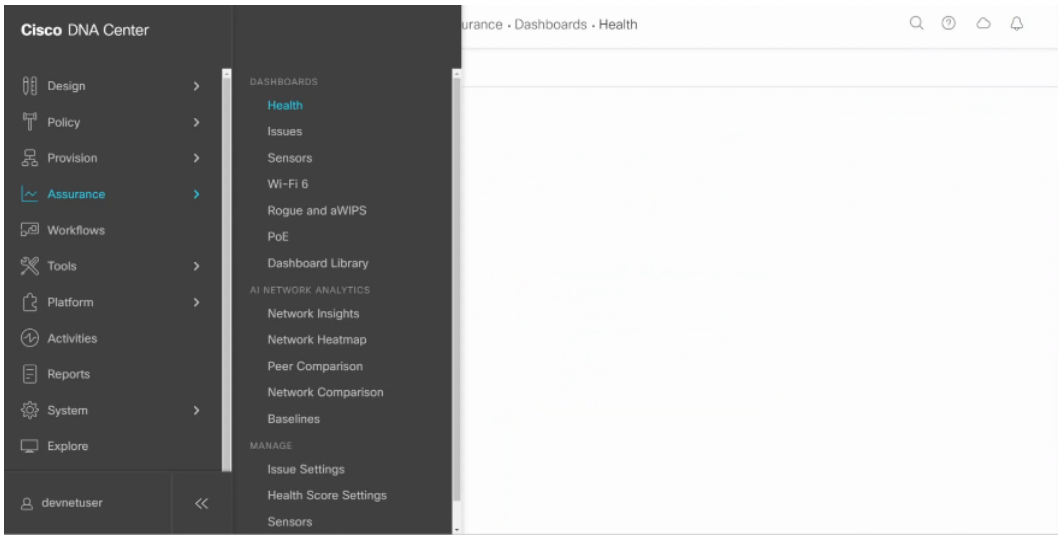
On peut par exemple voir les appareils qui ne sont pas assignés et adapter leur configuration pour qu'elle soit conforme :



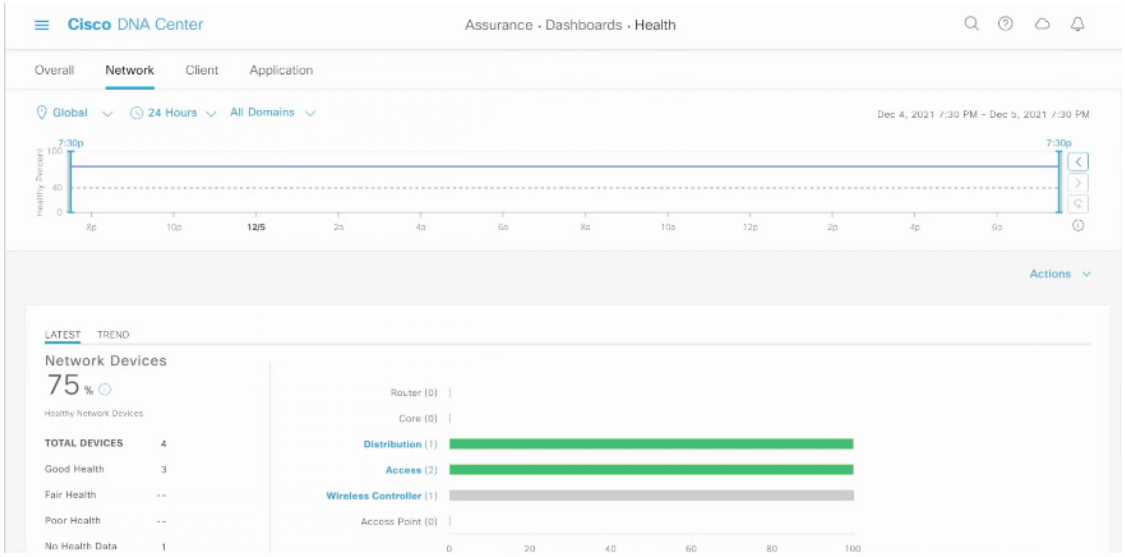
L'appareil par exemple n'a pas la dernière version de logiciel à jour :



Dans la section Health du menu on peut gérer le statut du réseau :



Voici un réseau considéré « en bonne santé » :



Faisons un comparatif entre un réseau réseau DNA Center et une gestion de réseau traditionnel :

- Dans la gestion d'un réseau traditionnel :

Les appareils sont configurés un à un par SSH ou une connexion console, Les appareils sont configurés par une connexion à la console avant d'être déployés.

La configuration et les politiques sont géré par appareil (distribué).

Le déploiement de nouveaux réseau peut prendre un long moment à cause du laborieux manuel requis. Les erreurs et défaillances sont plus dû à l'effort manuel qui augmente.

- Dans la gestion d'un réseau DNA Center-Based :

Les appareils sont géré de manière centralisé et sont monitorés depuis un DNA Center GUI ou autres applications utilisant son REST API.

L'administrateur communique le comportement du réseau prévu au DNA Center, qui change leurs comportement avec les configurations et la gestion des appareils du réseau.

Les configurations et politiques sont géré de manière centralisé.

Les versions logiciel sont aussi géré de manière centralisés. Le DNA Center peut gérer le serveur Cloud pour de nouvelles versions et mettre à jour les appareils gérés.

Le déploiement de nouveaux réseaux est bien plus rapide, les nouveaux appareils peuvent automatiquement recevoir leurs configuration depuis le DNA Center sans que cela nécessite de configuration manuelle.

Cours 63 : Ansible, Puppet, Chef

Dans ce cours nous verrons ce qu'est Ansible, Puppet et Chef ainsi que leur fonctionnement.

Nous ferons tout d'abord une introduction dans la configuration d'outils de gestion, puis nous verrons le fonctionnement de Ansible, Puppet, Chef.

Commençons tout d'abord par expliquer le concept de la configuration de drift.

Configuration drift se passe lorsque un changement individuel fait au fil du temps à cause de la configuration d'un appareil pour dévier depuis une configuration standard/correct comme définit par la compagnie.

Chaque appareil aura une partie unique de sa configuration (Les adresses IP, les nom d'hôtes, etc.)

la plupart de la configuration des appareils est définit dans un template standard désigné par le réseau d'architecte/ingénieurs de l'entreprise.

Lorsqu'un ingénieur individuel fait des changement sur l'appareil (par exemple pour résoudre et fixer un problème réseau, tester des configurations, etc.), la configuration d'un appareil peut dévier de son standard.

Les enregistrements de ces changements individuels et leurs raisons sont conservés, cela peut conduire à des problèmes futures.

Même sans outils d'automatisation, il est meilleur d'avoir des pratiques de gestion de configuration.

Par exemple lorsqu'un changement est fait, sauvegarder la configuration comme fichier texte et le placer dans un dossier partagé.

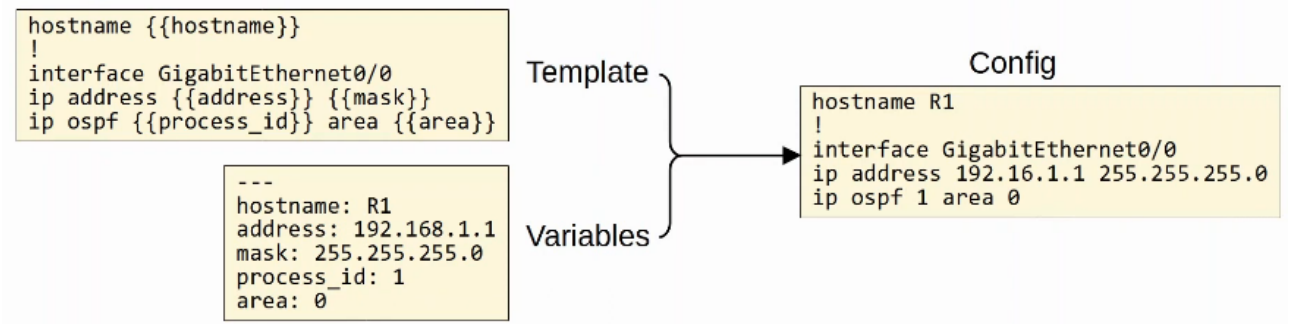
Un système de standard de nomage peut être utilisé par exemple : nom_prénom_date

Cela permet de suivre la trace des anciennes versions de la configuration et de voir quelles changements ont été fait. Il y a tout de même des défauts à ce système, comme lorsqu'un ingénieur peut oublier de placer la nouvelle configuration dans le fichier après avoir fait des changements, comment savoir lequel peut être considéré comme étant le bon fichier.

Même si la configuration est bien sauvegardé, cela ne garantie pas que la configuration corresponde bien au standard.

Le provisionnement de configuration se réfère à comment les changements de configuration sont appliqués aux appareils. Cela inclus configurer de nouveaux appareils aussi.

De manière traditionnel, la configuration fournis est appliqué en connectant aux appareils un par un via SSH. Ce n'est pas pratique pour de grands réseaux. Des outils de gestion de configuration comme Ansible, Puppet et Chef permettent de faire des changements aux appareils à grande échelle en très peu de temps/effort. Il y a deux composants essentiels : templates et les variables pour former une configuration :



Voyons les différents outils de gestion de configuration. Les outils de gestion de configuration sont des outils d'automatisation qui facilitent un contrôle centralisé d'un grand nombre d'appareils.

Ces outils ont été originellement développés après l'augmentation des Vms, pour permettre à un administrateur serveur système d'automatiser le processus de création, configuration et de suppression des Vms. Elles sont aussi largement utilisés pour gérer des appareils d'un réseau.

Ces outils peuvent être utilisés pour faire des tâches comme par exemple :

Générer des configurations pour de nouveaux appareils à grande échelle.

Faire fonctionner des changement de configuration sur des appareils (tous les appareils ou une partie d'entre eux). Vérifier que la configuration des appareils est en accord avec les standards définis. Il est aussi possible

de comparer des configurations entre plusieurs appareils, et entre différentes versions de configurations sur le même appareil.

Ansible est un outil de gestion de configuration appartenant à Red Hat.

Ansible est écrit en Python. Ansible est « agentless » cela signifie que cela ne requière pas de logiciel spécial pour être lancé dans des appareils de gestion.

Ansible utilise SSH pour connecter aux appareils, faire des changements de configuration, extraire des informations, etc.

Ansible utilise des modèle de push. Le serveur Ansible (Noeud de contrôle) utilise SSH pour gérer des appareils et pousser leurs changements de configurations.

Puppet et Chef utilise un modèle de pull.

Après avoir installé Ansible on peut créer plusieurs fichiers textes :

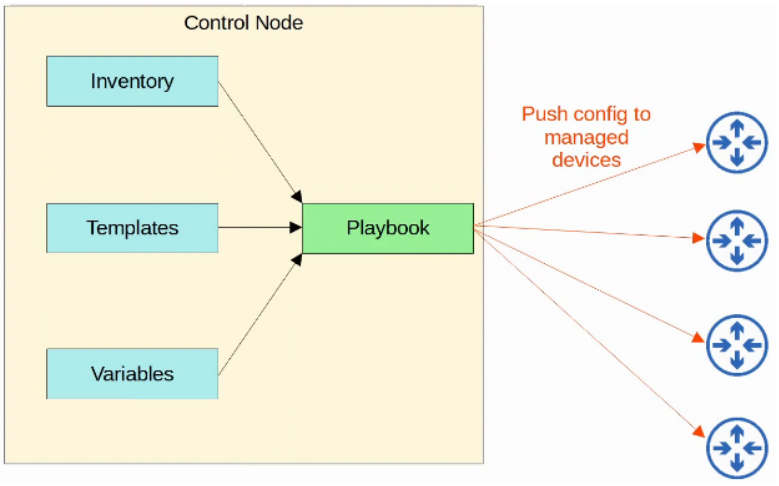
- Playbooks : ces fichiers sont des « blueprint d'automatisation de tâche ». Ils ont en dehors de la logique et des actions de tâches que Ansible devrait faire. Ecrite en YAML.

- Inventaire : Ces fichiers listent les appareils qui peuvent être gérés par Ansible, comme les caractéristiques de chaque appareil comme leur rôle d'appareil (Accéder à un Switch, Switch Core, Routeur WAN, Firewall, etc..) écrit en INI, YAML et dans d'autres formats.

- Templates : Ces fichiers représentent le fichier de configuration d'appareils, mais pour des valeurs spécifique pour des variables non fournis. Ecrite en format Jinja2.

- Variables : Ces fichiers listent les variables et leurs valeurs. Ces valeurs sont substitués en des templates pour créer des fichiers de configuration complets. Ecrit en YAML.

Voici un exemple d'utilisation d'Ansible sur un réseau dans lequel la configuration est poussé :



Puppet est un outil de gestion de configuration écrit en Ruby. Puppet est basé sur des agents.

Des logiciels spécifiques doivent être installés sur des appareils gérés.

Pas tous les appareils Cisco supportent les agent Puppet.

Puppet peut être lancé sans agent, dans lequel un agent proxy lance un hôte externe, et l'agent proxy utilisé SSH pour se connecter à l'appareil géré et communiquer avec eux.

Le serveur Puppet est appelé « Puppet Master ».

Puppet utilise un modèle de « pull » (les clients « pull » la configuration depuis le Puppet Master).

Les clients utilise TCP 8140 pour communiquer avec le Puppet Master.

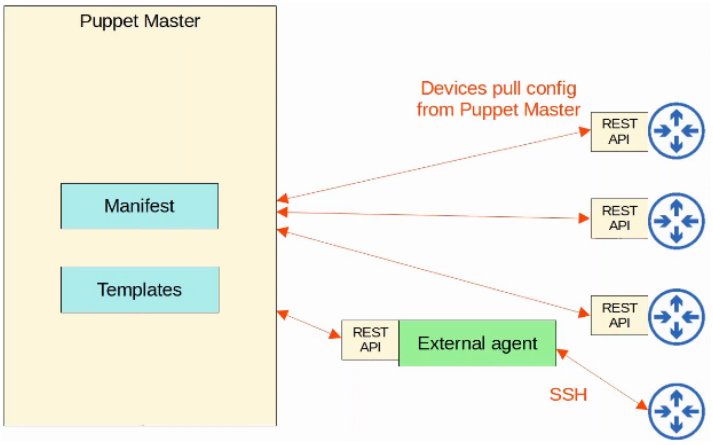
Au lieu de YAML, il utilise un langages propriétaire pour les fichiers.

Les fichiers textes requis dans le Puppet Master incluent :

- Manifest : Ce fichier définit l'état de la configuration voulue d'un appareil réseau.

- Templates : De manière similaire à des Templates Ansible. Utilisé pour générer des Manifests.

Voici un réseau utilisant Puppet :



Chef est un outil de gestion de configuration écrit en Ruby.

Chef est basé sur des agent. Des logiciels doivent être installés dans les appareils gérés. Pas tous les appareils Cisco supportent des agent Chef.

Chef utilise un modèle Pull. Le serveur utilise TCP 10002 pour envoyer des configurations aux clients. Les fichiers utilisent un DSL (Domain-Specific Language) basé sur Ruby.

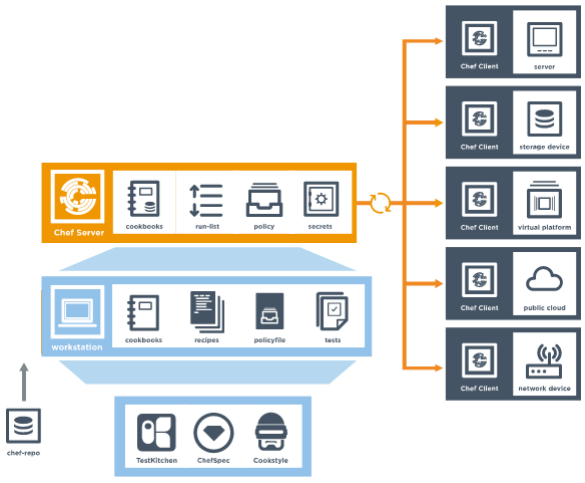
Les fichiers textes utilisés par Chef incluent :

- Ressources : les « ingrédients » dans une recette. L'objet de configuration est géré par Chef.

Les recettes dans un livre de recette. En dehors de la logique et des actions des tâches fonctionnant sur les ressources.

- Cookbook : ou livre de recette en Français est plusieurs recettes regroupés ensemble.

- Run-list : Est une liste ordonné de recettes qui sont lancés pour apporter l'appareil vers l'état de configuration voulue.



Voici un réseau utilisant Chef :

Voici un tableau comparant ces différents outils :

	Ansible	Puppet	Chef
Fichiers Clefs définis par des actions	Playbook	Manifest	Recipe, Run-list
Protocole de communication	SSH	HTTPS (Par REST API)	HTTPS (par REST API)
Port clef	22 (Port SSH)	8140	10002
Basé sur agent/sans agent	Sans Agent	Basé sur agent (Ou sans agent)	Basé sur agent
Push/Pull	Push	Pull	Pull