Cours et prises de notes

Tunui Franken

Dernière compilation le 24 mars 2021 à 18:56

Table des matières

Première partie

Réseau

Chapitre 1

Introduction

1.1 Présentation

Un réseau est un moyen de fournir des ressources. Par exemple réseau routier (transport de marchandises, de personnes), réseau de l'eau, réseau électrique...

1.2 Réseaux convergents

- Avant : Réseaux séparés taditionnels
  - les réseaux informatiques, téléphoniques et de diffusion sont séparés
  - les services tournent sur plusieurs réseaux en même temps
  - les différents réseaux ne communiquent pas (technos et protocoles differents)
- Aujourd'hui : Réseaux convergents
  - les réseaux distincts de données, de téléphonie et de vidéo convergent
  - les réseaux convergents sont capables de transmettre des données, de la voix et de la vidéo entre differents types d'appareils sur la même infrastructure de réseau
  - plusieurs services tournent sur un même réseau

1.3 Fiabilité des réseaux

- Tolérance aux pannes
  Redondance : offrir plusieurs chemins possibles pour les paquets. Les messages sont découpés en paquets qui peuvent emprunter des chemins différents.
- Évolutivité
  - possibilité d'agrandir le réseau sans affecter les services existants
  - normes et protocoles reconnus
- Qualité de service (Quality of Service (QoS))
  - les transmissions vocales et vidéo par exemple augmentent le niveau d'attente des utilisateurs sur les réseaux
  - la fusion des services de données, voix, et vidéo nécessite de gérer l'encombrement
  - si volume du trafic > bande passante : appareils gardent en mémoire les paquets pour les retransmettre plus tard

— dans une politique QoS un routeur peut donner la priorité à certains types de connexion (VoIP > web)

— Sécurité
Deux préoccupations : sécurité des infrastructures et sécurité des informations.
Trois exigences principales :

1. confidentialité
   seuls les destinataires prévus peuvent accéder aux données
2. intégrité
   garantit que les données n'ont pas été altérées lors de la transmission
3. disponibilité
   accès rapide et fiable

1.4 Types de réseaux

— Local Area Network (LAN)
— Wide Area Network (WAN)
— Internet
— Intranet
— Extranet

1.5 Différents moyens de connexion à Internet

La connection se fait par l'intermediaire d'un Fournisseur d'Accès Internet (FAI) (Internet Service Provider (ISP)).

— Petite entreprise ou travail à domicile

1. Câble
   Proposé par les fournisseurs de télévision par câble, le signal des données Internet passe par le même câble que pour la télé. Large bande passante, grande disponibilité, connexion permanente à l'Internet.
2. Digital Subscriber Line (DSL)
   Large bande passante, grande disponibilité, connexion permanente à l'Internet. Utilise une ligne téléphonique.
   En général dans un bureau ou à domicile on se connecte avec une ligne Asymmetric DSL (ADSL). En ADSL le débit descendant est supérieur au débit ascendant.
3. Cellulaire
   Utilise le réseau de téléphonie mobile. Performances limitées par les capacités du téléphone et de la tour à laquelle il est connecté.
4. Satellite
   Grande disponibilité donc avantage dans les régions qui autrement n'auraient aucune connectivité. Les antennes paraboliques ont besoin d'une ligne de vue claire vers le satellite.

5. Ligne commutée
   Peu couteux. Utilise n'importe quelle ligne téléphonique et un modem. Faible bande passante mais utile lors d'un déplacement par exemple.

— Connexion d'entreprise
Les entreprises peuvent nécessiter une bande passante plus élevée, specialisée et des services gérés.

1. Ligne louée dédiée
   Circuit réservé au sein du FAI et qui relie des bureaux géographiquement séparés mais qui ont besoin d'un réseau privé. Loué sur une base mensuelle ou annuelle.
2. Metro Ethernet
   Parfois connu sous le nom Eternet WAN. Étend la technologie d'accès au LAN au WAN.
3. Business DSL
   Disponible dans différents formats. Ligne d'abonné numérique symétrique (Symmetric DSL (SDSL)) similaire à la DSL grand public (ADSL) mais permet des téléchargements en amont et en aval aux mêmes débits élevés.
4. Satellite
   Lorsqu'une solution câblée n'est pas disponible.

1.6 Cloud computing

Le cloud est un moyen d'accéder aux données et de les stocker. Pour les entreprises : nouvelles fonctionnalités sans devoir investir dans une nouvelle infrastructure, former de personnel, ou acheter de nouveaux logiciels en licence. Fonctionne grâce aux centres de données. En général les fournisseurs de cloud stockent les données dans plusieurs centres de données situés à différents emplacements.

Il y a 4 types de cloud :

1. Cloud public
   — applications et services dispos pour le grand public
   — gratuit ou payant
   — utilise internet pour fournir les services
2. Cloud privé
   — destiné à une organisation ou une entité comme un gouvernement
   — peut utiliser le réseau privé de l'organisation mais c'est coûteux
   — ou géré par une société externe avec une sécurité d'accès stricte
3. Cloud hybride
   — deux ou plusieurs clouds (partie privée et partie publique)
   — chaque partie est un objet distinct, mais les deux sont reliés par une architecture unique
   — degrés d'accès différents sur la base des droits d'accès des utilisateurs
4. Cloud communautaire
   — similaire au cloud public mais offre des niveaux de sécurité, de confidentialité et de conformité réglementaire d'un cloud privé
   — pour des entreprises qui ont les mêmes besoins et les mêmes problèmes

1.7 Architectures de réseaux

1.7.1 Réseaux commutés sans frontière

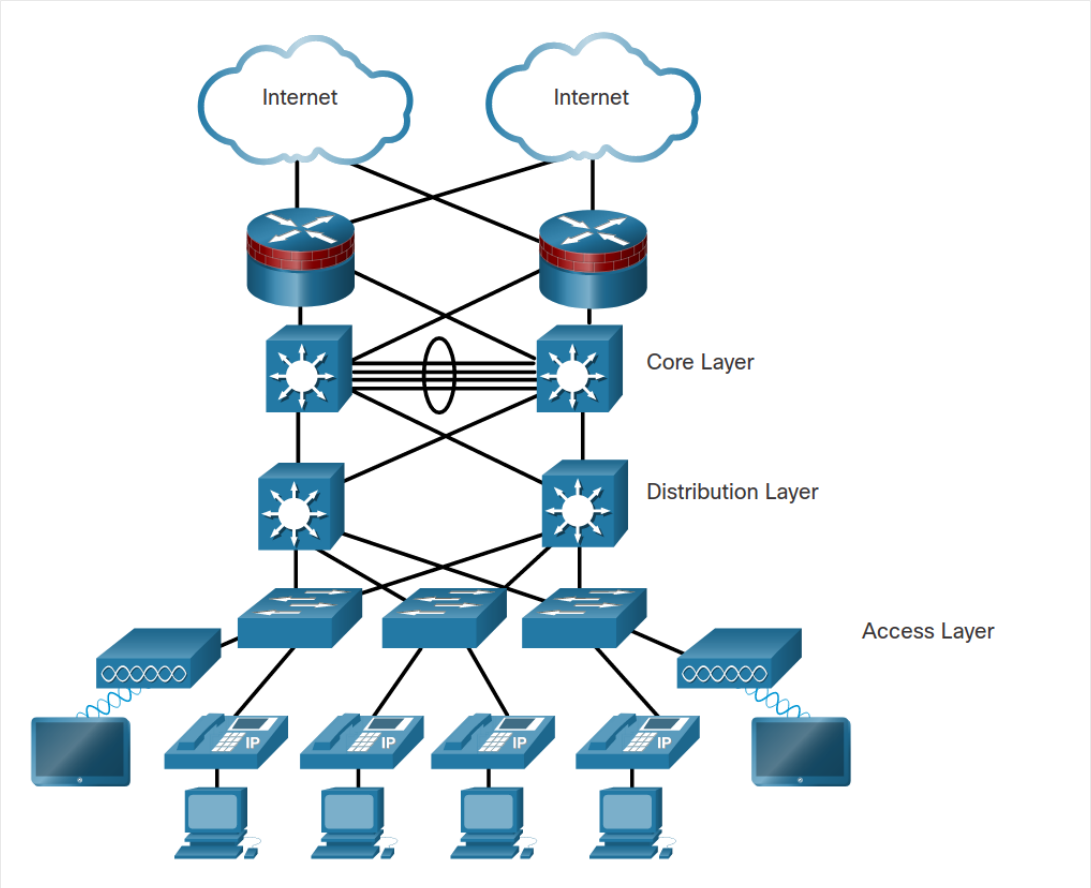
L'architecture de réseau commuté sans frontière suit les principes suivants :

— **Hiérarchique** — Chaque périphérique a un rôle défini et compréhensible. La gestion et le déploiement sont simples.
— **Modulaire** — Le design permet l'expansion du réseau de manière triviale et à la demande.
— **Résilient** — Le réseau est toujours disponible.
— **Flexible** — Le réseau permet l'équilibrage de charge en utilisant toutes les ressources du réseau.

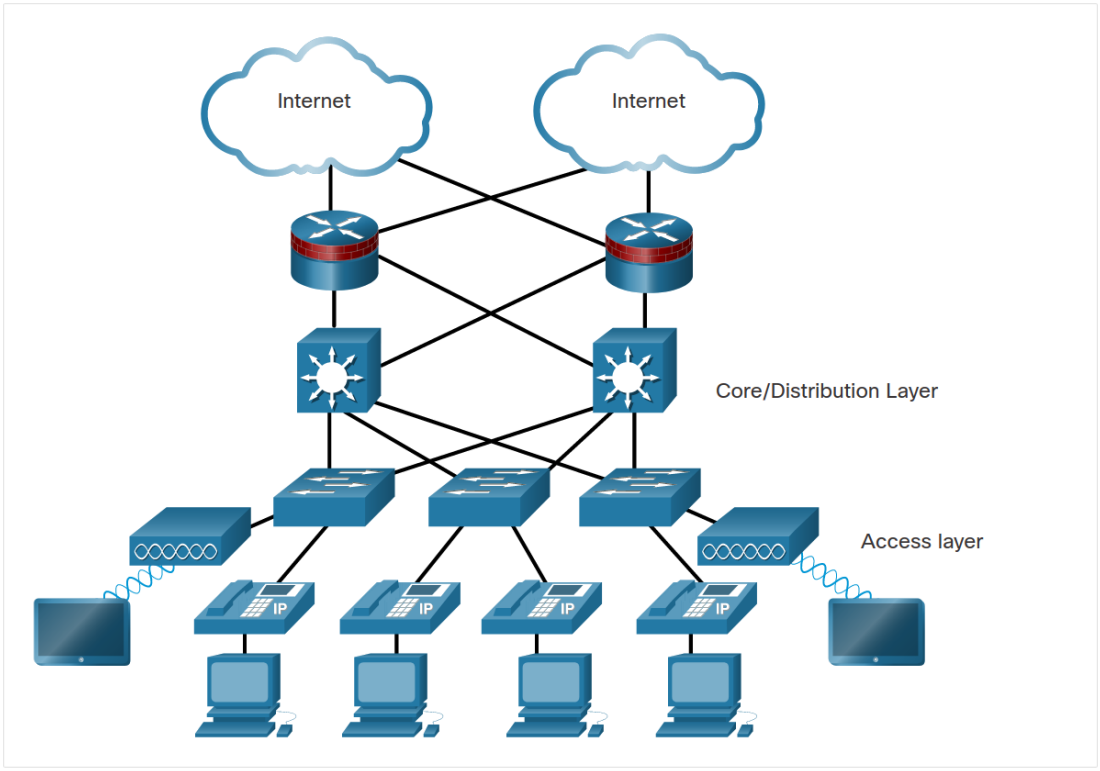
Ces principes ne sont pas indépendants.

Deux manières de créer un design hiérarchique sont les modèles à deux niveaux et les modèles à trois niveaux.

1. Modèle à 3 niveaux



2. Modèle à 2 niveaux



Ces designs prennent en compte trois couches de réseau. Chaque couche peut être vue comme un module défini avec des fonctions spécifiques :

1. **Couche d'accès** — Représente le bord du réseau, où le trafic entre et sort. Traditionnellement, la principale fonction d'un communateur de couche d'accès est de fournir l'accès réseau à l'utilisateur. Les switchs de couche d'accès sont connectés aux switchs de couche de distribution.
2. **Couche de distribution** — Relie les couches d'accès et de cœur.
3. **Couche de cœur** — Son but principal est d'assurer l'isolation des défaillances. En anglais on parle de *backbone*.

Chapitre 2

OSI vs. TCP/IP

2.1 Le modèle OSI

7. Couche Application (Application Layer)
   - HTTP, FTP, SMTP, TFTP...
   - application utilisateur
   - transfert de fichiers, messagerie...
6. Couche Présentation (Presentation Layer)
   - plus trop utilisé
   - mise en forme des données (encodage, ascii, utf...)
   - éventuellement chiffrement, compression
5. Couche Session (Session Layer)
   - plus trop utilisé
   - gestion de l'échange des données entre applications distantes
   - syncho des échanges
   - définition des points de reprise
   - avec une meilleure fiabilité des canaux, on se permet de ne plus l'utiliser
4. Couche Transport (Transport Layer)
   - gère les flux de données (TCP, UDP)
   - transmission des segments entre les deux systèmes d'extrémité
   - première couche de bout en bout
   - denière couche de contrôle des infos
   - assure aux couches supérieures un transfert fiable
3. Couche Réseau (Network Layer)
   - adresses logiques (IP)
   - transmission des paquets sur un réseau
   - acheminement
   - routage
   - contrôle de congestion
   - adaptation des blocs de données aux capacités du réseau physique (Maximum Transmission Unit (MTU))
2. Couche Liaison (Data Link Layer)
   - adresses MAC

— transmission des trames sur un canal de communication
— contrôle

1. Couche Physique (Physical Layer)
   — câbles, infrastructures, hubs...
   — transmission des bits sur un canal de communication (électrique, optique, radio)
   — garantir la transmission (1 bit envoyé = 1 bit reçu)
   — synchronisation

Quand les données a transmettre traversent les couches, chaque couche ajoute son en-tête, ce qui ralonge la longueur de la donnée à transmettre.

2.2 Le modèle TCP/IP

Le modèle TCP/IP est développé parallèlement au modèle OSI. Développé par des boîtes privées, il s'est imposé avant que le modèle OSI ne soit prêt. OSI est un standard plus complet, de référence.

4. Application (OSI #7)
   fusionne si besoin les couches Session, Présentation et Application du modèle OSI
3. Transport (OSI #4)
   TCP et UDP
2. Réseau (OSI #3)
   IPv4 et IPv6
1. Accès Réseau (OSI #1)
   regroupe les couches Physique et Liaison

La couche Transport effectue une liaison directe, c'est-à-dire qu'elle communique avec la couche Transport de son destinataire.

Les couches Réseau et Accès Réseau quant à elles sont en liaison indirecte : elles communiquent avec le routeur de leur réseau.

Chapitre 3

Couche Physique

Il y a trois types de connection :

1. câble en cuivre : les signaux sont des variations d'impulsions électriques.
2. câble à fibre optique : les signaux sont des variations lumineuses.
3. sans fil : les signaux sont des variations de transmission de fréquences radio.

3.1 Normes

Les normes TCP/IP (des couches OSI 2 et supérieures) sont implémentées dans le logiciel et régies par l'IETF. Les normes de couche Physique sont mises en œuvre dans le matériel et sont régies notamment par :

— International Organization for Standardization (ISO)
— Electronic Industries Association (EIA)/Telecommunications Industry Association (TIA)
— International Telecommunication Union (ITU)-T
— American National Standards Institute (ANSI)
— Institute of Electrical and Electronics Engineers (IEEE)

3.2 Perturbations

Il y a deux types de perturbations qu'un support physique câblé en cuivre peut subir :

1. les interférences (Electro-Magnetic Interference (EMI) et Radio-Frequency Interference (RFI))
2. la diaphonie

Pour contrer les effets des interférences, ont peut entourer le câble d'un blindage métallique et faire une mise à la terre. Pour contrer les effets de la diaphonie, les paires de fils opposés sont torsadées pour annuler la perturbation.

On peut également lors de la conception de l'infrastructure de câblage éviter les sources d'interférences potentielles.

3.3 Câble en cuivre

3.3.1 Types de câbles

Voici différentes dénominations en fonction du type de câblage :

- Unshielded Twisted Pair (UTP cable) : câble torsadé simple pas protégé du tout.
- Foiled Twisted Pair (FTP cable) : câble torsadé simple seulement entouré d'aluminium, c'est-à-dire écranté.
- Shielded Twisted Pair (STP cable) : câble torsadé blindé mais non écranté.
- Shielded and Foiled Twisted Pair (SFTP cable) : câble torsadé blindé et écranté.
- Foiled Foiled Twisted Pair (FFTP cable) : câble torsadé non blindé mais doublement écranté : chaque paire est entourée d'aluminium et l'ensemble du câble est écranté également.

Les câbles à 100 Mbps utilisent deux paires (une pour l'expédition et une pour la réception). Les câbles à 1 Gbps utilisent quatre paires (deux pour l'expédition et deux pour la réception).

Il existe aussi le câble coaxial, qui contient deux conducteurs qui partagent le même axe. Contrairement aux câbles précédents qui utilisent le RJ45, il peut utiliser différents types de connecteur :

- Bayonet Neill-Concelman (BNC)
- type N
- type F

Les câbles UTP cable ont pratiquement remplacé les câbles coaxiaux mais ceux-ci sont encore utilisés dans les installations sans fil et les installations de câbles internet.

3.3.2 Normes

Parmi la myriade d'organismes de normalisation, l'ISO et l'Association Française de Normalisation (AFNOR) sont à connaître.

Pour les normes de câbles il y a deux nomenclatures :

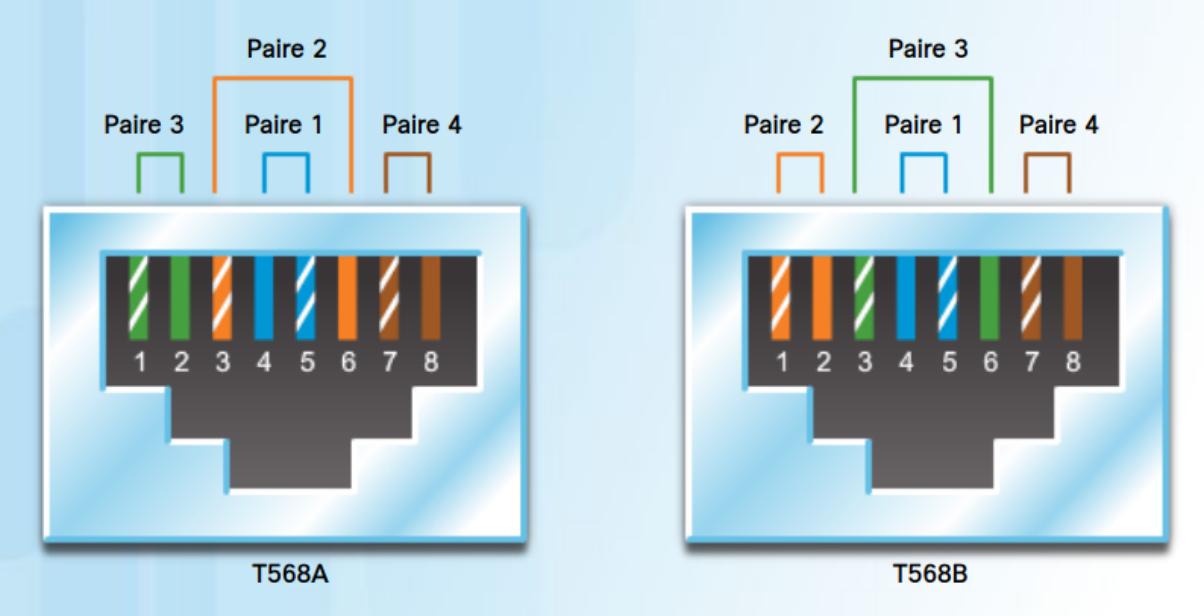
- *Catégories* : 3, 4, 5, 5e, 6 et 7
  Les catégories 3 et 4 sont obsolètes et ont normalement disparu. La norme est faite par la TIA et les tests sont réalisés en laboratoire.
- *Classes* : A, B, C, D, E et F
  Les correspondances attendues entre *catégorie* et *classe* sont les suivantes :
  - 3 → C
  - 5e → D
  - 6 → E
  - 7 → F

  La norme est faite par l'ISO et l'International Electrotechnical Commission (IEC) et les tests sont à réaliser sur le site pour vérifier la correspondance avec la catégorie.

3.3.3 Câble droit, câble croisé, câble inversé

Deux normes pour le sens des paires au niveau de la prise :

1. T568A : pour les câbles croisés seulement
2. T568B : pour les câbles droits et les câbles croisés



Pour vérifier si un câble est droit ou croisé, on met les deux bouts côte à côte. Si les couleurs sont dans le même sens c'est un câble droit, sinon, il s'agit d'un câble croisé.

Généralement, pour relier deux équipements de même type, on utilise un câble croisé.

Il existe également le câble inversé, propriétaire Cisco permettant d'obtenir une connexion avec un routeur ou un port console de switch.

3.4 Fibre optique

C'est un fil en verre très pur, transparent, flexible est très fin. N'est pas soumis aux perturbations EMI et RFI. Peut transmettre les signaux avec moins d'atténuation avec une bande passante plus large que n'importe quel autre support réseau, sur de plus longues distances.

On les utilise principalement dans quatre domaines d'application :

1. les réseaux d'entreprise
2. la technologie Fiber To The Home (FTTH)
3. les réseaux longue distance
4. les réseaux sous-marins

3.4.1 Modes de fibre

1. Single-Mode optical Fiber (SMF) : Produit un seul chemin direct pour la lumière. Utilise le laser comme source du signal lumineux. Son cœur présente un très faible diamètre. Répandue dans les réseaux longue distance (plusieurs centaines de kilomètres).

2. Multi-Mode optical Fiber (MMF) : Utilise plusieurs chemins lumineux possible. Utilise habituellement des LED comme source du signal lumineux. La taille de son cœur est supérieure. La lumière d'une LED entre dans la fibre sous différents angles. Généralement utilisée dans les réseaux locaux.

3.4.2 Connecteurs

— connecteur Straight-Tip (ST)
— connecteur Subscriber Connector (SC)
— connecteur Lucent Connector (LC)
— connecteurs LC bidirectionnels

3.5 WiFi

IEEE norme 802.11

Wireless Personal Area Network (WPAN) : par ex. bluetooth

Wireless LAN (WLAN) : wireless LAN, pour un bâtiment

Wireless WAN (WWAN) : portée de plusieurs kilomètres

1. **connexion en mode ad hoc** = directe entre deux équipements.
2. **connexion en mode infrastructure** = plusieurs équipements se connectent à un Access Point (AP), équivalent à un switch.

Chapitre 4

Couche Liaison

4.1 Sous-couches

La couche Liaison (couche 2) consiste en deux sous-couches :

1. Logical Link Control (LLC)
2. Media Access Control (MAC)

La sous-couche MAC procède à l'encapsulation après que la sous-couche LLC ait ajouté des informations de contrôle :

— **Délimitation des trames**
Procure d'importants délimiteurs pour identifier les champs à l'intérieur d'une trame. Ces bits de délimitation permettent la synchronisation entre l'émetteur et le récepteur.
— **Adressage**
Procure l'adresse source et destination pour transporter la trame de couche 2 entre périphériques partageant le même support.
— **Détection d'erreurs**
Inclut une en-queue pour détecter les erreurs de transmission (voir Frame Check Sequence (FCS)).

Cette sous-couche MAC procure également le contrôle d'accès au support, permettant à de multiples périphériques de communiquer via un support partagé (half-duplex). Les communications full-duplex ne nécessitent pas de contrôle d'accès.

4.2 Accès aux supports

Chaque fois qu'une trame passe par un routeur, le routeur effectue les actions suivantes :

1. accepte la trame d'un support
2. désencapsule la trame
3. réencapsule le paquet de couche 3 dans une nouvelle trame
4. achemine la nouvelle trame jusqu'au support du segment du réseau physique correspondant à ce qu'elle a lu dans le paquet de couche 3.

4.3 Protocoles de couche 2

Il y a une différence de protocoles utilisés sur les LAN et sur les WAN. En effet, un LAN couvre une zone géographique restreinte avec une grande bande passante, alors que pour le WAN c'est le contraire.

Comme les WAN relient entre eux des LAN, ils utilisent des protocoles variés en fonction du contexte physique. On peut notamment lister les protocoles WAN suivants :

— Point to Point Protocol (PPP)
— High-level Data Link Control (HDLC)
— Frame Relay
— Asynchronous Transfer Mode (ATM)
— X.25

Ces protocoles de couche 2 ont tendance à être remplacés dans le WAN par ethernet.

Sur le LAN on pourra trouver les protocoles suivants :

— Ethernet
— 802.11 (sans fil)
— PPP
— HDLC
— Frame Relay

4.4 Ethernet

4.4.1 Historique

— en câble (voir 3.3) : CSMA/CD
— en WiFi (voir 3.5) : CSMA/CA

Au début, prises vampire, puis coaxial fin : prises en T, ce qui est limité pour ajouter des postes...

Puis on a mis le câble dans une boîte : boîtier de couche 1 avec des prises RJ45 : le hub.

1. regénère le signal et le rebalance sur tous les ports
2. le débit de connexion est partagé sur le bus du hub : si y a 10Mbits sur le bus, et deux postes, alors chacun n'a que la moitié du débit
3. plus on ajoute de postes, plus on baisse le débit

Collisions : deux signaux ne peuvent pas cohabiter : les postes ne peuvent émettre que si personne d'autre ne parle.

En cas de collision, on attend un temps aléatoire avant de retransmettre. Pour peu de postes ça marche bien, mais dès qu'on a plus de machines on ne fait que gérer les collisions.

Puis on est passé au commutateur (switch).

1. ressemble au hub mais de couche 2
2. ce n'est plus un bus, c'est une matrice : réseau en étoile

3. le switch ne divise plus le débit sur ses ports
4. plus besoin d'éviter les collisions : normalement on peut être partout en full-duplex : les signaux peuvent se croiser sur le même câble

4.4.2 Ethernet et le modèle OSI

Ethernet et utilisé dans les couches OSI 1 et 2. Les standards sont :

— **802.2** pour la sous-couche LLC
— **802.3** pour la sous-couche MAC et la couche Physique

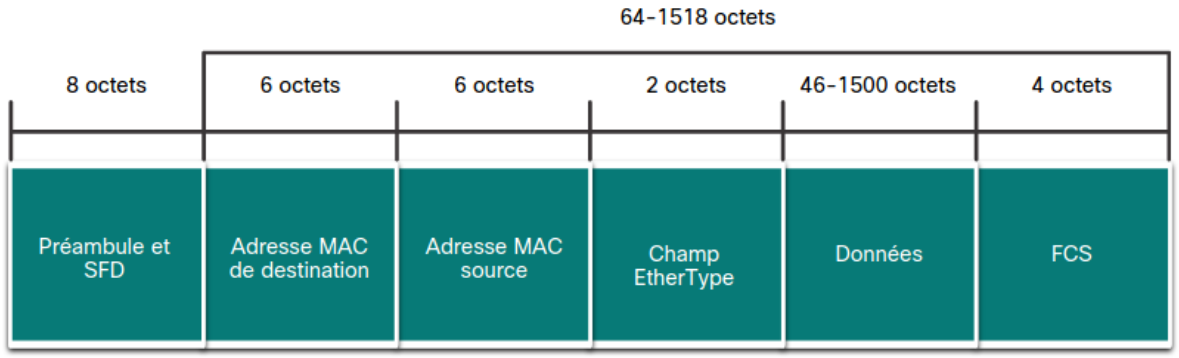
4.5 La trame

4.5.1 Champs de trame

La trame est le PDU de la couche 2. C'est la seule à posséder une en-tête ainsi qu'une en-queue. La structure de l'en-tête et de l'en-queue dépendent du protocole utilisé.

Les trames ethernet doivent avoir une longueur comprise entre 64 et 1518 octets. Cette longueur prend en compte toute la trame sauf le premier champ, le préambule.

Les champs sont les suivants :



— **Préambule (7 octets) et Start Frame Delimiter (SFD) (1 octet)** — 8 octets
Utilisés pour indentifier le début et la fin de la trame. Cela sert à la synchronisation. Le préambule indique aux destinations de se préparer à recevoir une nouvelle trame.
— **Adresse MAC de destination** — 6 octets
Identifie le destinataire sur ce réseau.
— **Adresse MAC de source** — 6 octets
Identifie la carte réseau ou l'interface d'origine de la trame.
— **EtherType** — 2 octets
Identifie le protocole de couche 3 utilisé. Les valeurs hexadécimales les plus fréquentes sont :
— `0x800` pour IPv4
— `0x86DD` pour IPv6
— `0x806` pour ARP
Ce champ peut aussi parfois s'appeler **Type** ou **Longueur**
— **Données** — entre 46 et 1500 octets

Contient toute la PDU encapsulée de la couche supérieure. Comme la longueur minimale de la trame est de 64 octets, si un petit paquet est encapsulé, d'autres bits appelés `remplissage` sont utilisés pour augmenter la taille de la trame pour atteindre la longueur minimale.

— **FCS** — 4 octets
Détecte les erreurs de transmission de la trame. Le périphérique expéditeur utilise le Cyclic Redundancy Check (CRC) et inclut les résultats dans ce champ. Le périphérique récepteur va pouvoir lui aussi calculer le CRC pour le comparer à la valeur dans le champ FCS. Si les valeurs correspondent, il n'y a pas d'erreur. Sinon la trame est rejetée.

4.6 Adressage MAC

Le champ correspondant à l'adressage de la couche 2 contient les adresses MAC de destination puis de source. Ces adresses sont différentes à chaque réseau qu'un paquet va traverser. À chaque passage de routeur, une nouvelle trame est constituée avec les adresses des nouvelles extrémités du réseau en cours.

Ces adresses de couche 2 ne sont utilisées que pour la délivrance locale. Elles n'ont pas de sens au-delà du réseau local. En revanche, les adresses de couche 3 vont de l'hôte source à l'hôte destinataire, peu importe le nombre de tronçons de réseau traversés.

Les adresses MAC sont codés sur 48 bits et sont représentés sous forme de 12 chiffres hexadécimaux (4 bits par chiffre hexadécimal).

Pour créer une adresse MAC, l'IEEE demande aux constructeurs de respecter deux règles :

— Les trois premiers octets forment un identifiant Organizationally Unique Identifier (OUI) attribué au revendeur correspondant.
— Toutes les adresses MAC ayant le même identifiant OUI doivent utiliser une valeur unique dans les trois derniers octets.

L'adresse MAC est gravée dans la ROM de la carte réseau. Mais il est aujourd'hui possible de la changer dans le logiciel. Le filtrage ou le contrôle du trafic sur la base de l'adresse MAC n'est donc plus aussi sécurisé.

4.7 ARP

Le protocole ARP convertit les adresses IPv4 en adresses MAC. Pour cela il maintient une table ARP.

C'est le périphérique expéditeur qui regarde sa table ARP pour encapsuler un paquet à envoyer.

— Si l'adresse IPv4 de destination appartient au même réseau, le périphérique recherche l'adresse IPv4 dans sa table ARP.
— Si l'adresse IPv4 de destination appartient à un réseau différent, le périphérique va chercher l'adresse de la passerelle par défaut.

Si l'adresse MAC n'est pas trouvée, le périphérique envoie une requête ARP.

Une requête ARP est un broadcast. Ainsi, tous les hôtes recevant la trame devront la traiter.

Le message de la requête ARP contient les éléments suivants :

— l'adresse IPv4 cible, donc celle pour laquelle le périphérique envoie la requête ARP
— l'adresse MAC cible, qui n'est pas connue et n'est donc pas renseignée dans le message

La requête ARP est ensuite encapsulée dans une trame ethernet avec une adresse de broadcast en destination. L'hôte recevant la requête et se reconnaissant dans la demande va envoyer une réponse ARP.

Le message de la réponse ARP contient les éléments suivants :

— l'adresse IPv4 de l'expéditeur, donc la même que la requête
— l'adresse MAC de l'expéditeur, qui manquait donc dans la requête

Les entrées ARP qui n'ont pas été utilisées pendant une durrée correspondant à un Time To Live (TTL) sont supprimées de la table.

4.8 Commutation

4.8.1 Table MAC

Quand un switch reçoit des trames, il regarde d'abord si l'adresse source est dans sa table MAC.

— Si elle s'y trouve il va remettre à 0 le TTL.
— Sinon il l'ajoute.

Puis il regarde l'adresse de destination.

— Si elle est dans sa table MAC il remet à 0 le TTL puis transfère la trame par ce port.
— Si elle ne s'y trouve pas il va renvoyer la trame à tous ses ports connectés, sauf celui d'où est arrivé la trame.

4.8.2 Méthodes de transmission de switch

Grâce aux Application Specific Integrated Circuits (ASIC), les switchs prennent des décisions très rapidement. Ils utilisent l'une des deux méthodes suivantes pour transmettre une trame :

1. Commutation par stockage et retransmission (store-and-forward)
   C'est la méthode principale de commutation de LAN des switchs Cisco.
   Avant de commencer le processus de transmission, reçoit la trame entière et vérifie la présence d'erreurs en utilisant le CRC.

   (a) Vérification d'erreurs :
   Compare le FCS de la dernière plage du datagramme avec le FCS issu de ses propres calculs.

(b) Mise en mémoire tampon automatique :

Supporte toute combination de débits ethernet. Une trame arrivant à 100 Mbps à envoyer par un port à 1 Gbps aura besoin de la méthode de stockage et de retransmission. Dès que les débits ingress et egress ne correspondent pas, le switch effectue les actions suivantes :

— enregistre la trame complète dans une mémoire tampon
— calcule le contrôle FCS
— transmet la trame au buffer du port egress
— l'envoie

2. Commutation par coupure (cut-through)

Débute le processus de transmission dès que l'adresse MAC d'une trame arrivant et son port egress correspondant ont été déterminés.

Contrairement à la méthode par stockage et retransmission qui supprime les trames ne validant pas le FCS, la méthode par coupure peut transmettre des trames invalides. Mais elle est plus rapide.

4.8.3 Domaines de collision

Des segments de réseau qui partagent la même bande passante sont appelés des *domaines de collision*. Quand au moins deux appareils sont sur le même domaine de collision et essayent de communiquer en même temps, il y aura une collision.

— half-duplex : Chaque segment est dans son propre domaine de collision.
— full-duplex : Pas de domaine de collision.

Par défaut, le port d'un switch autonégotie : il sera en full-duplex si l'appareil adjacent peut-être en full-duplex, sinon il sera en half-duplex.

4.8.4 Domaines de diffusion (broadcast)

Un ensemble de switchs interconnectés forme un domaine de diffusion unique. Seul un périphérique de couche Réseau comme un routeur peut diviser un domaine de diffusion.

Quand deux switchs sont connectés l'un à l'autre, ils agissent comme un seul switch. Ils sont sur le même domaine de diffusion.

Trop de broadcasts peuvent donner une congestion.

4.8.5 Congestion de réseau

— **Débits de port rapides** : Coûte plus cher mais réduisent la congestion.
— **Commutation interne rapide** : Meilleure performance.
— **Grandes mémoires tampon de trames** : Permet au trafic d'un port ingress plus rapide (par ex. 1 Gbps) d'être transmis à un port egress plus lent (par ex. 100 Mbps) sans perte de trame.
— **Haute densité de port** : Permet d'utiliser moins de switchs.

4.9 VLAN

Les paquets unicast, broadcast et multicast ne sont transférés qu'au sein d'un même VLAN. Un switch ne fait pas routage. Pour changer de VLAN, il faut un périphérique qui prend en charge le routage.

Au niveau de la couche Réseau, plusieurs sous-réseaux Internet Protocol (IP) peuvent exister sur un même réseau commuté sans l'utilisation de VLAN. Mais dans ce cas, les broadcasts de niveau 2 (comme les demandes ARP), seront reçus par tous les hôtes du réseau commuté.

Un VLAN crée un domaine de broadcast logique qui peut s'étendre sur plusieurs segments de réseau local physique. Cela améliore les performances réseau. Si un périphérique d'un VLAN envoie une trame ethernet de broadcast, tous les périphériques du VLAN la recevront, mais pas les périphériques d'autres VLAN.

Chaque port du switch peut être attribué à un seul VLAN à l'exception des ports connectés à un téléphone IP ou à un autre switch.

4.9.1 Avantage des VLAN

Bénéfice	**Description**
Domaines de broadcast plus petits	réduction du nombre d'hôtes dans le domaine
Sécurité optimisée	seuls les ordinateurs du même VLAN peuvent communiquer ensemble
Amélioration de l'efficacité des ressources IT	simplifie la gestion du réseau, les VLAN peuvent être nommés pour les identifier
Coût réduit	moins de besoins matériels donc coûte moins cher
Meilleures performances	les domaines de broadcast plus petits réduisent le traffic inutile
Gestion simplifiée des projets et applications	fonctions distinctes donc gestion plus simple d'un projet spécialisé

4.9.2 Types de VLAN

- VLAN par défaut
  - tous les ports sont attribués à VLAN 1 par défaut
  - le VLAN natif est le VLAN 1 par défaut
  - le VLAN de gestion est le VLAN 1 par défaut
  - le VLAN 1 ne peut pas être renommé ou supprimé
- VLAN de données
  VLAN configurés pour diviser un réseau en groupes d'utilisateurs. Le trafic de gestion vocale et réseau ne doit pas être autorisé sur les VLAN de données.
- VLAN natif

Normalement le trafic d'un VLAN est marqué avec une balise de 4 octets dans l'en-tête de la trame ethernet pour identifier le VLAN auquel appartient la trame. Mais parfois le trafic est non balisé (quand il est généré par le switch ou des périphériques hérités). Le port interurbain 802.1Q le place alors sur le VLAN natif. Il est recommandé de configurer le VLAN natif en tant que VLAN inutilisé, distinct du VLAN 1. Souvent on dédie un VLAN fixe pour le VLAN natif.

— VLAN de gestion
VLAN de données configuré pour le trafic de gestion réseau, y compris SSH, Telnet, HTTP Secure (HTTPS), HyperText Transfer Protocol (HTTP), SNMP.

— VLAN voix
VLAN distinct pour la VoIP.
Le trafic VoIP requiert :
— bande passant consolidée pour garantir la qualité de la voix
— priorité de transmission
— possibilité de routage autour des zones encombrées du réseau
— délai inférieur à 150ms sur tout le réseau

4.9.3 Plages de VLAN

Une trame destinée à un VLAN se voit insérer un champ dans son en-tête ethernet pour identifier le VLAN auquel il appartient. L'ID de VLAN étant codé sur 12 bits, on peut créer jusqu'à $2^{12} = 4096$ VLAN.

Il y a 2 plages de VLAN :

1. Plage VLAN normale : de 1 à 1005.
   Les VLAN de cette plage sont utilisés dans les entreprises de petite et moyenne taille. Les VLAN 1002 à 1005 sont réservés aux anciennes technologies de réseau comme Token Ring et Fiber Distributed Data Interface (FDDI). Les VLAN 1, et de 1002 à 1005 sont créés automatiquement et ne peuvent pas être supprimés. Les configurations sont enregistrées dans la mémoire flash dans une base de données appelée `vlan.dat`
2. Plage VLAN étendue : de 1006 à 4096.
   Les VLAN de cette plage sont utilisés par les FAI et les entreprises suffisamment grandes pour avoir besoin de la plage étendue. Ces VLAN ne sont pas pris en compte par le protocole VTP. Les configurations sont enregistrées dans la `running-config`.

4.9.4 Trunks de VLAN

Tous seuls, les VLAN ne font que découper un switch comme s'il y avait des sous-réseaux, ce qui n'est pas très utile. Les trunks permettent à tout le trafic VLAN de se propager entre les switchs. On peut donc faire communiquer plusieurs périphériques du même VLAN à travers plusieurs switchs sans passer par un routeur.

La norme la plus répandue pour la prise en charge des trunks est la norme IEEE `802.1Q`.

Les liaisons de trunk relient des switchs ou routeurs.

Un trunk n'appartient pas à un VLAN en particulier. Disons plutôt qu'il laisse passer (ou pas) les communications de plusieurs VLAN.

4.9.5 Modes d'interfaces d'un switch

On peut configurer les ports d'un switch en deux modes distincts :

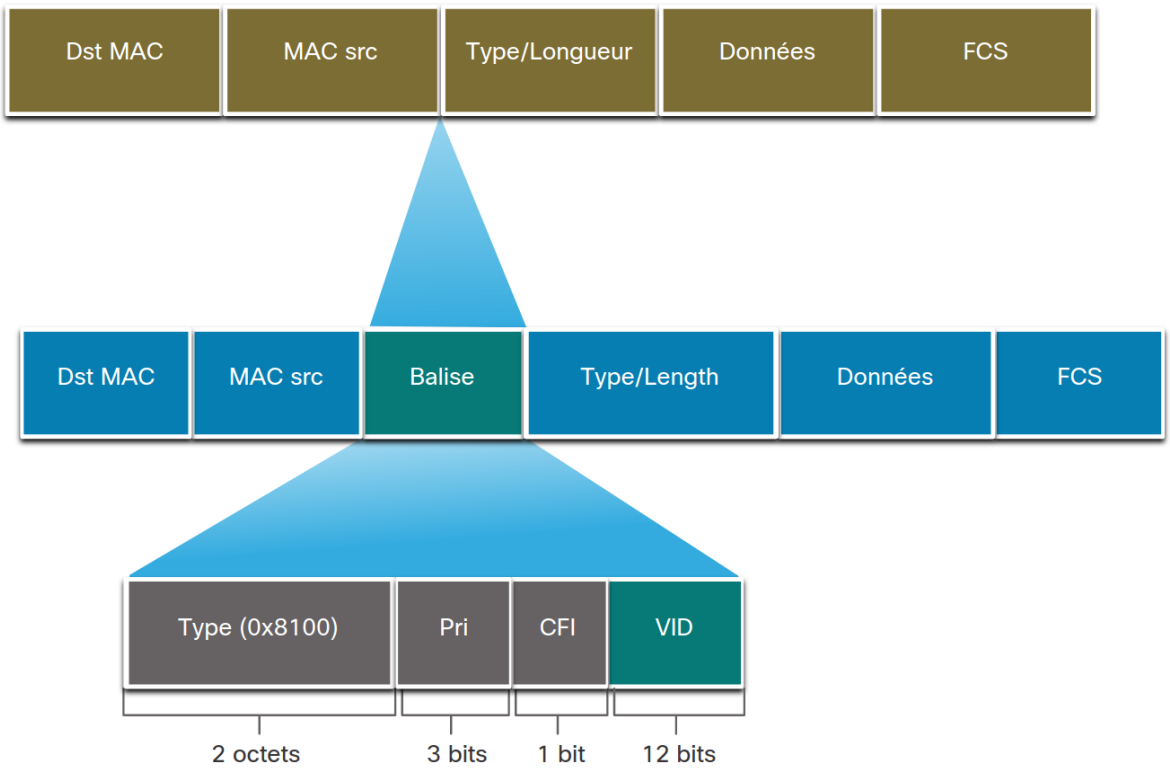
1. **Mode `access`** — sur un port connecté à un hôte. Les interfaces en mode `access` sont attribués à un VLAN.
2. **Mode `trunk`** — sur un port connecté à un autre switch. Les interfaces en mode `trunk` laissent passer ou refusent le passage à certains VLAN. En configurant deux interfaces sur deux switchs (un sur chaque switch) en mode `trunk`, cela crée un lien trunk.

Après avoir créé des VLAN, on configure les ports, soit en mode `access`, soit en mode `trunk`. En mode `access` on attribue le port à un VLAN, en mode `trunk` on définit quels VLAN seront pris en charge par ce port.

4.9.6 Balises d'identification de VLAN

L'en-tête d'une trame ethernet ne contient pas de champ pour identifier son VLAN. Du coup, quand une trame est placée sur un trunk, des informations pour identifier son VLAN doivent être ajoutées. Cela s'appelle l'*étiquetage* et se fait avec l'en-tête IEEE `802.1Q`. Il inclut un étiquette de 4 octets dans l'en-tête de la trame.

Lorsqu'une trame arrive sur un port en mode `access`, elle ne contient pas d'information sur son VLAN. C'est le port qui est assigné à un VLAN et donc c'est le commutateur qui va ajouter l'étiquette `802.1Q` dans l'en-tête de la trame à sa réception. Il envoie ensuite la trame par un port `trunk`.



— **Type** — Une valeur de 2 octets défini sur `0x8100` pour ethernet.

— **Priorité utilisateur** — Une valeur de 3 bits pour le QoS.
— **Canonical Format Identifier (CFI)** — Identificateur de 1 bit qui permet aux trames Token Ring d'être transportées sur les liaisons ethernet.
— **VLAN ID (VID)** — Numéro d'identification du VLAN sur 12 bits, qui permet donc jusqu'à 4096 VLAN différents.

4.9.7 VLAN natif

On a vu que lorsqu'une trame arrive sur un port `access`, il n'est pas balisé, et reçoit son balisage en fonction du VLAN du port, ce qui est normal. Mais lorsqu'une trame non balisée arrive sur un port `trunk`, c'est qu'elle a été envoyée sur cette liaison par un switch. La trame est alors attribuée au VLAN natif.

Les trames de gestion envoyées entre commutateurs sont un exemple de trafic non balisé.

Trames marquées sur le VLAN natif

Certains périphériques ajoutent une étiquette VLAN pour le trafic VLAN natif. Mais le trafic envoyé sur le VLAN natif ne doit pas être étiqueté. Si un port `802.1Q` reçoit une trame étiqueté avec un ID correspondant au VLAN natif, la trame sera abandonnée. Il faut donc veiller à configurer ses périphériques pour qu'ils n'ajoutent pas d'étiquette VLAN au trafic VLAN natif. Ces périphériques peuvent être des téléphones IP, des serveurs, des routeurs, et des switchs non-Cisco.

Trames non marquées sur le VLAN natif

Les trames non étiquetées devraient être rares dans un réseau bien architecturé. Si un port `trunk` d'un switch Cisco reçoit une trame non étiquetée, il la transfère au VLAN natif. Si aucun périphérique n'est assigné au VLAN natif et qu'il n'y a pas d'autres ports trunk, la trame est rejetée.

Tout trafic non étiqueté est transféré par rapport à la valeur du Port VLAN ID (PVID), qui correspond à l'ID du VLAN natif.

Les ports en mode `access` ne devraient jamais être assignés au VLAN natif, ce qui implique qu'une trame étiquetée ne devrait pas être étiquetée comme VLAN natif.

4.9.8 VLAN voix

Pour utiliser la VoIP, un VLAN distinct est nécessaire. Cela permet la QoS et la sécurité.

Quand le trafic passe par un téléphone IP, il y aura deux VLAN : un VLAN de données et un VLAN voix, qui aura une priorité suppérieure.

4.9.9 Configuration des VLAN

Quand on configure des VLAN de la plage normale, les configurations sont enregistrées dans un fichier `vlan.dat` en mémoire flash. Cela veut dire qu'il n'y a pas besoin de `S1# copy running-config startup-config`. Il est cependant important de pas oublier cette commande, parce que la configuration de VLAN implique en général des commandes qui ne sont pas mises en flash.

Création de VLAN

Il est de bonne pratique de donner un nom descriptif à chaque VLAN que l'on crée.

```
Switch# configure terminal
Switch(config)# vlan <vlan-id>
Switch(config-vlan)# name <vlan-name>
Switch(config-vlan)# end
Switch#
```

On peut également créer plusieurs VLAN d'un coup avec `vlan <vlan-id>` : la commande `vlan 100,102,105-107` crée les VLAN 100, 102, 105, 106 et 107.

Assignation de ports à des VLAN

Créer un VLAN et ne rien faire d'autre ne sert à rien. Les ports connectés à des hôtes doivent êtres en mode `access` et être ajoutés à un VLAN :

```
Switch# configure terminal
Switch(config)# interface <interface-id>
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan <vlan-id>
Switch(config-if)# end
Switch#
```

Il est bien-sûr possible de configurer plusieurs ports à la fois en utilisant `interface range`.

Les VLAN sont configurés sur les ports du switch et pas sur les hôtes. Les Personal Computer (PC) qui font partie d'un VLAN n'en ont pas conscience. Ils font partie d'un sous-réseau associé à un certain VLAN. Les autres hôtes faisant partie du même VLAN doivent être configurés dans le même sous-réseau.

Un port ne peut être ajouté qu'à un seul VLAN de données à la fois, mais il peut être associé à un VLAN voix aussi. Un PC peut être connecté à un téléphone IP, et le téléphone IP au switch. Dans ce cas, le port du switch branché au téléphone recevra du trafic pour le VLAN voix du téléphone et le VLAN de données du PC.

Il faut alors que le port soit associé aux deux VLAN.

Pour assigner un port à un VLAN voix :

```
Switch(config)# vlan 150
Switch(config-vlan)# name VOICE
Switch(config-vlan)# exit
Switch(config)# exit
Switch(config)# interface fa0/18
Switch(config-if)# switchport mode access
Switch(config-if)# mls qos trust {cos|device cisco-phone|dscp|ip-precedence}
Switch(config-if)# switchport voice vlan 150
Switch(config-if)# end
Switch#
```

Pour réassigner un port au VLAN par défaut (VLAN 1) :

```
Switch(config)# interface <interface-id>
Switch(config-if)# no switchport access vlan
Switch(config-if)# end
Switch#
```

Validation de la configuration

```
Switch# show vlan [brief|id <vlan-id>|name <vlan-name>|summary]
```

Les 4 options sont donc :

1. **brief** — Affiche le nom, l'état et les ports d'un VLAN par ligne.
2. **ip** <vlan-id> — Affiche des informations sur un VLAN identifié par son numéro.
3. **name** <vlan-name> — Affiche des informations sur un VLAN identifié par son nom.
4. **summary** — Affiche un bref sommaire.

Pour vérifier si un port spécifique a bien été assigné à un VLAN :

```
Switch# show interfaces <interface-id> switchport
```

Suppression de VLAN

```
Switch(config)# no vlan <vlan-id>
```

Mais attention, il vaut mieux réassigner les ports associés à ce VLAN avant de le supprimer, sinon les ports ne pourront plus communiquer tant qu'ils ne feront pas partie d'un VLAN actif.

On peut aussi supprimer entièrement le fichier `vlan.dat`, remettant ainsi la configuration d'usine pour les VLAN :

```
Switch# delete flash:vlan.dat
```

Configuration des trunks

Pour configurer un lien trunk, il faut configurer les deux ports à chaque extrémité du lien.

```
Switch# configure terminal
Switch(config)# interface <interface-id>
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk native vlan <vlan-id>
Switch(config-if)# switchport trunk allowed vlan <vlan-list>
Switch(config-if)# end
Switch#
```

La commande `trunk native` permet de définir un VLAN autre que 1 comme natif. La commande `trunk allowed` définit une liste de VLAN qui peuvent passer par ce port.

Pour retirer des VLAN au trunk, remettant les valeurs par défaut VLAN 1 en natif et aucun VLAN associé :

```
Switch(config)# interface <interface-id>
Switch(config-if)# no switchport trunk native vlan
Switch(config-if)# no switchport trunk allowed vlan
Switch(config-if)# end
Switch#
```

Quand le trunk est dans son état par défaut, il utilise le VLAN 1 comme VLAN natif et laisse passer tous les VLAN.

Vérification des trunks

```
Switch# show interfaces <interface-id> switchport
Switch# show interface trunk
```

4.9.10 Protocole DTP

Dynamic Trunking Protocol (DTP) configure automatiquement les switchs par annonce sur les switchs Cisco.

Les modes DTP sont :

- `dynamic desirable` — fait une requête trunk
- `dynamic auto` — fait une réponse trunk
- `trunk` — se met en mode `trunk` et l'annonce à son voisin
- `access` — se met en mode `access` et l'annonce à son voisin

S'il y a une erreur de négociation, les interfaces passent en mode `access`.

Si un switch Cisco est connecté à un périphérique ne faisant pas de DTP (autre marque par exemple), il est problématique d'envoyer des trames DTP. Il vaut mieux mettre le port en question explicitement en mode trunk et retirer la négociation :

```
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport nonegociate
```

Pour réactiver le DTP :

```
Switch(config-if)# switchport mode dynamic auto
```

La commande `switchport nonegociate` désactive la négociation DTP sur ce port, mais nécessite de configurer le mode du port à la main. Il faut que le port soit en mode `access` ou `trunk`.

Un résumé :

	Dynamic Auto	**Dynamic Desirable**	**Trunk**	**Access**
Dynamic Auto	Access	Trunk	Trunk	Access
Dynamic Desirable	Trunk	Trunk	Trunk	Access
Trunk	Trunk	Trunk	Trunk	Connection limitée
Access	Access	Access	Connection limitée	Access

Pour vérifier les modes DTP :

```
Switch# show dtp interface <interface-id>
```

4.9.11 Protocole VTP

VTP est un protocole de couche 2 qui permet la cohérence des configurations de VLAN en s'occupant de l'addition, de la suppression et de l'édition de VLAN dans un domaine VTP.

Il propage la configuration d'un switch aux autres switchs de la topologie, évitant de configurer à la main trop de switchs. Mais il permet aussi d'éviter des erreurs de configuration.

C'est avant d'ajouter des VLAN qu'il faut décider si on veut utiliser VTP dans le réseau. Avec VTP, on fait les configurations de manière centralisée sur un périphérique réseau.

Domaine VTP

Un domaine VTP est composé d'un ou plusieurs périphériques interconnectés et qui partagent le même nom de domaine VTP. Un périphérique réseau ne peut être que dans un seul domaine VTP. Les changements de configuration globale pour le domaine se font soit en ligne de commande, soit avec SNMP.

Par défaut, les switchs Cisco sont en VTP mode serveur dans l'état de domaine de non-gestion. Le switch prend un domaine soit quand il reçoit une annonce sur un lien trunk, soit quand on configure un domaine de gestion.

Sur un serveur VTP, il n'est pas possible d'ajouter ou de modifier des VLAN avant que le domaine de gestion soit appris ou spécifié.

Quand un switch reçoit une annonce VTP sur un lien trunk, il hérite du domaine de gestion et du numéro de révision de la configuration VTP. Le switch ignore les annonces d'un domaine différent ou d'un numéro de révision antérieur.

Si on configure le switch en *VTP Transparent*, on peut créer et modifier des VLAN, mais les changements n'affectent que ce switch.

Quand on effectue un changement sur un serveur VTP, le changement est propagé à tous les périphériques réseau du domaine VTP. Les annonces VTP sont transmis par tous les Inter-Switch Link (ISL) et les connections trunk IEEE 802.1Q.

VTP ajoute les VLAN de manière dynamique sur plusieurs types de LAN avec des noms uniques et des associations d'index internes.

Modes VTP

Il y a 3 modes :

1. **Server** — C'est le mode par défaut. Dans ce mode, on peut créer, modifier et supprimer des VLAN et ajouter d'autres paramètres de configuration (comme la version VTP et le VTP pruning). Ces configurations valent pour le domaine VTP en entier. Les serveurs VTP annoncent leur configuration VLAN aux autres périphériques du domaine. Ils synchronisent également leur configuration avec d'autres périphériques réseau en fonction d'annonces reçues sur des liens trunk.
2. **Client** — Les clients VTP se comportent de la même manière que les serveurs VTP, sauf qu'il n'est pas possible de créer, modifier ou supprimer des VLAN.
3. **Transparent** — Les périphériques transparents ne participent pas au VTP. Un périphérique transparent n'annonce et ne synchronise pas sa configuration VLAN. Cependant, en VTP version 2, les périphériques VTP transparents transfèrent les annoncent VTP qu'ils reçoivent sur un lien trunk.

Annonces VTP

Chaque périphérique dans le domaine VTP envoie périodiquement des annonces par chaque interface trunk vers une adresse multicast réservée. Ces annonces VTP sont reçues par les périphériques voisins, qui mettent leurs configurations VTP et VLAN à jour.

Les annonces contiennent les informations suivantes :

— ID de VLAN (ISL et 802.1Q)
— Nom de LAN émulé (pour ATM LAN Emulation (LANE))
— Valeurs SAID 802.10 (FDDI)
— Nom de domaine VTP
— Numéro de révision de la configuration VTP
— Configuration VLAN, ce qui inclut la taille MTU pour chaque VLAN
— Format de trame

VTP version 2

Il faut choisir entre VTP version 1 et version 2.

Par rapport à VTP version 1, la version 2 a les caractéristiques additionnelles suivantes :

— **Support du Token Ring**
— **Support des Type-Length Value (TLV) non reconnues** — Un serveur ou client VTP propage les changements de configuration vers ses autres trunks, même pour les TLV qu'il ne peut pas déchiffrer. La TLV non reconnue est enregistrée dans la Non Volatile RAM (NVRAM).
— **Mode transparent dépendant de la version** — En VTP version 1, un périphérique en mode VTP transparent inspecte les messages VTP du nom de domaine et de la version et ne transfère le message que si la version et le nom de domaine

correspondent. En version 2, le mode transparent transfère les messages VTP sans vérifier la version.

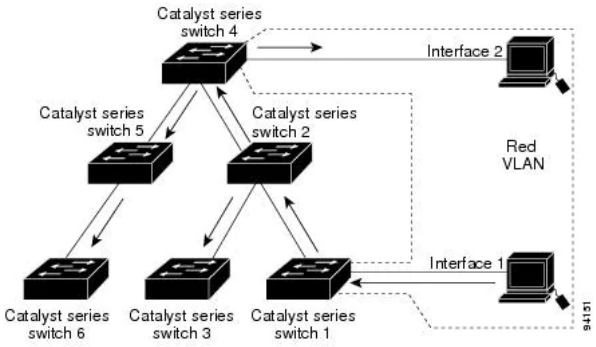
— **Vérifications de cohérence** — En VTP version 2, les vérifications de cohérence de VLAN (comme les noms de VLAN et leur valeur) ne sont effectuées que quand on ajoute des informations en ligne de commande ou par SNMP. Les vérifications de cohérence ne sont pas faites quand une nouvelle information est obtenue à partir d'un message VTP ou quand elle est lue dans la NVRAM.

VTP pruning

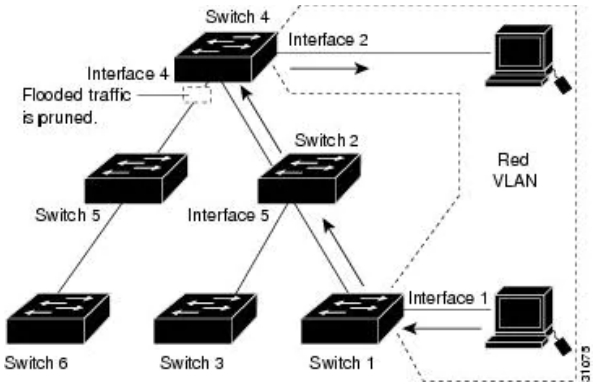
En français, élagage VTP. Il permet d'améliorer l'utilisation de la bande passante en réduisant le trafic non nécessaire, comme les paquets broadcast, multicast et unicast. Pour faire ça, le VTP pruning réserve le trafic d'inondation aux liens trunk que le trafic doit utiliser pour accéder aux périphériques réseau appropriés. Par défaut le VTP pruning est désactivé.

Pour que le VTP pruning soit efficace, tous les périphériques du domaine de gestion doivent prendre en charge le VTP pruning, ou bien sur les périphériques qui ne le supportent pas il faut configurer les VLAN autorisés sur les trunks.

Sans VTP pruning :



Avec VTP pruning :



Si on active le VTP pruning sur un serveur VTP, cela active le VTP pruning pour le domaine entier. Par défaut, les VLAN éligibles pour le VTP pruning sont les VLAN 2 à 1000. Le VLAN 1 n'est pas éligible au VTP pruning. Le trafic du VLAN 1 ne peut donc pas être élagué.

Pour activer le VTP pruning sur une interface :

```
S1(config-if)# switchport trunk pruning vlan
```

VTP Bomb

Quand on ajoute un switch à un réseau, par défaut il est configuré sans nom de domaine VTP et sans mot de passe. Il sera en mode serveur VTP. Comme aucun nom de domaine n'aura été défini, il va prendre le nom de domaine du premier paquet VTP qu'il reçoit. La révision de la configuration d'un nouveau switch est 0. Cela veut dire que le switch va considérer tout numéro de révision comme plus récent. Si les mots de passe coïncident, il effacera ses informations VTP. Par contre, si on branche un switch à un réseau avec

un nom de domaine et un mot de passe corrects mais un numéro de révision plus élevé que celui du réseau, alors le domaine VTP en entier va adopter la configuration du nouveau switch, ce qui peut causer des pertes d'informations VLAN sur tous les switchs du domaine VTP. Cela peut arriver par exemple avec un switch qui a été retiré du réseau pour maintenance et remis avec ses informations VLAN supprimées.

Configuration VTP

- Tous les périphériques d'un domaine VTP doivent tourner sur la même version VTP.
- On doit configurer un mot de passe sur chaque périphérique du domaine si le VTP est en mode sécurisé.
- Pour faire cohabiter des périphériques VTP version 1 et version 2, il faut que la version 2 soit désactivée sur les périphériques pouvant l'utiliser. Par défaut, la version 2 est désactivée.
- Il ne faut pas activer la version 2 sur un périphérique à moins que tous les périphériques du réseau sont compatibles avec la version 2. Quand on active la version 2 sur un serveur VTP, tous les périphériques compatibles du domaine activent la version 2.
- Quand on active ou désactive le VTP pruning sur un serveur VTP, cela active ou désactive le VTP pruning pour le domaine entier.
- Quand on configure des VLAN comme éligibles à l'élagage sur un switch, cela rend ces VLAN élagables sur ce switch, pas sur tous les périphériques du domaine.

Par défaut, le VTP est configuré de la manière suivante :

- Nom de domaine VTP : aucun
- Mode VTP : serveur
- Version VTP : version 1
- Mot de passe VTP : aucun
- VTP pruning : désactivé

Paramètres globaux

Configurer / retirer un mot de passe VTP : `Switch# [no] vtp password <password>`	Et pour l'afficher : `Switch# show vtp password`
Activer le VTP pruning : `Switch# [no] vtp pruning`	On vérifie la configuration avec : `Switch# show vtp status`
Activer la version 2 : `Switch# [no] vtp version {1|2}`	On vérifie la configuration avec : `Switch# show vtp status`

Switch en tant que serveur VTP

1. Configurer le switch en tant que serveur VTP :

```
Switch# configure terminal
Switch(config)# vtp mode server
```

2. Définir le nom de domaine VTP (32 caractères maximum) :

```
Switch(config)# vtp domain <domain_name>
Switch(config)# end
```

3. Vérifier la configuration :

```
Switch# show vtp status
```

Switch en tant que client VTP

```
Switch(config)# [no] vtp mode client
```

L'utilisation du mot `no` retourne au mode par défaut, serveur.

Désactiver le VTP (mode transparent)

```
Switch(config)# [no] vtp mode transparent
```

Afficher les statistiques VTP

```
Switch# show vtp counters
```

4.10 STP

4.10.1 Redondance dans les réseaux de couche 2

Il est très commun de chercher à offrir une redondance dans un réseau commuté pour contourner les défaillances et ainsi prévenir l'interruption des services. Cette redondance se fait grâce à l'ajout de chemins physiques et/ou logiques. Mais ces chemins supplémentaires augmentent le risque de créer des boucles physiques et logiques de couche 2. Or, un réseau local ethernet exige une topologie sans boucle avec un chemin unique entre deux périphériques. Une telle boucle peut provoquer la propagation des trames ethernet jusqu'à saturation et interruption de la liaison. Cela s'appelle une Tempête de broadcasts.

Contrairement aux protocoles de la couche 3 (IPv4 et IPv6), ethernet dans la couche 2 n'a pas de mécanisme pour identifier et éliminer les trames dans une boucle. Grâce aux champs `TTL` (pour l'IPv4) et `Hop limit` (pour l'IPv6), un paquet ne peut être retransmis qu'un nombre limité de fois. Les commutateurs ethernet n'ont pas de telle fonction.

Pour parer ce problème, un protocole, le STP.

4.10.2 Présentation du protocole STP

Le protocole STP permet la redondance tout en créant une topologie de couche 2 sans boucle. Il permet de désactiver une liaison qui si activée provoquerait une boucle. Si un lien devient défaillant et tombe, le protocole STP recalculera les chemins possibles et réactivera le lien désactivé, permettant ainsi la redondance.

STP est activé par défaut sur les switchs Cisco.

Sans le STP, boucles de couche 2

Les boucles peuvent se faire avec des trames broadcast, multicast ou unicast et peuvent rendre indisponible un réseau en quelques secondes. Par exemple, prenons un type de broadcast répandu : les requêtes ARP. Quand un switch reçoit une trame pour un broadcast, il le renvoie sur tous ses ports sauf le port de réception. Le switch suivant fait la même chose et ainsi de suite jusqu'à ce que la trame repasse par le premier switch puisqu'il y a plusieurs chemins possibles. Nous voilà à la case départ et une boucle se produit. Chaque switch va constamment mettre à jour sa table MAC en fonction des trames reçues, ce qui résulte en une instabilité des tables MAC. Cela augmente l'utilisation du CPU et crée de fausses entrées dans les tables MAC. Et bien évidemment, les trames repassant sans cesse sur les mêmes liens, la bande passante va vite chuter également.

La même chose peut arriver en unicast : lorsqu'un switch reçoit une trame et qu'il ne connaît pas l'adresse MAC de destination, il renvoie la trame par tous ses autres ports, créant des doublons.

STA, l'algorithme de STP

Le STA désigne l'algorithme sur lequel se base STP pour faire ses calculs. Il a été inventé par Radia Perlman et publié en 1985. Le STA sélectionne un root bridge et coupe certains liens redondants en bloquant certains ports à des endroits stratégiques.

Il calcule le coût de tous les chemins de la topologie avec la bande passante. Ensuite, chaque switch détermine le chemin qui le sépare du root bridge. Une fois que les ports stratégiques ont été bloqués, chaque switch n'a qu'un chemin pour rejoindre le root bridge. On peut imaginer un arbre où le root bridge est le tronc. Chaque switch est sur une branche, avec un seul moyen de rejoindre le tronc.

Les chemins physiques redondants existent toujours mais ne servent pas. Si un chemin désactivé doit être utilisé pour compenser la perte d'un lien, STP recalcule les chemins.

4.10.3 Mise en place de la topologie sans boucle

STP utilise 4 étapes pour créer une topologie sans boucle :

1. choisir le root bridge
2. choisir les ports `root`
3. choisir les ports `Designated`
4. choisir les ports `Alternate` (bloqués)

Les switchs utilisent les BPDU pour partager leurs informations et élire le `root bridge` ainsi que les ports `root`, `Designated` et `Alternate`. Le BPDU contient un Bridge ID (BID) pour identifer le switch l'ayant envoyé. Ce BID est central dans les calculs STA. Il contient :

— **La priorité du bridge** (4 bits) — Sur les switchs Cisco la valeur par défaut est la valeur décimale 32768. Les valeurs vont de 0 à 61440 par sauts de 4096 (0, 4096, 8192, 12288...). Une valeur plus basse est prioritaire.

— **Un ID de système étendu** (12 bits) — Valeur décimale ajoutée à la priorité du bridge pour identifier le VLAN pour ce BPDU. Les switchs anciens (d'un temps où l'on n'utilisait pas de VLAN) ne contiennent pas cette valeur.
L'ID de système étendu permet d'utiliser différents root bridges pour différents VLAN. Ainsi certains ports peuvent êtres bloquants pour un VLAN mais laisser passer d'autres VLAN.
Quand le STP opère avec plusieurs VLAN, c'est une version Per-VLAN Spanning Tree (PVST). Il y a une instance de Spanning Tree par VLAN. Si tous les ports de tous les switchs font partie du VLAN 1, alors il n'y a qu'une seule instance de Spanning Tree.

— **L'adresse MAC du switch** (48 bits) — Il peut arriver que deux switchs soient configurés avec la même priorité et aient le même ID de système étendu. Pour les différencier, le switch ayant l'adresse MAC avec la valeur la plus basse aura la valeur de BID la plus basse.

Élection du root bridge

Le root bridge est le point de référence pour tous les calculs de chemin.

Tous les switchs du domaine de broadcast envoient des BPDU toutes les 2 secondes. Ces BPDU contiennent le BID du root bridge (`Root ID`) et le BID du switch qui émet le BPDU.

Chaque switch commence par se déclarer root bridge en mettant son BID en `Root ID`. En recevant les BPDU d'autres switchs, chaque switch va ensuite apprendre lequel a le BID le plus faible. C'est le switch ayant le BID le plus faible qui devient le root bridge.

La valeur par défaut du BID est 32768 ($= 2^{15}$), donc il est possible que plusieurs switchs aient la même priorité. Ce sera alors l'adresse MAC qui déterminera le root bridge, mais cela n'est pas forcément ce qu'il y a de mieux pour la topologie. Il est donc recommandé de configurer une priorité plus basse sur un switch que l'on désire voir devenir le root bridge.

Détermination du coût du chemin vers le root bridge

Une fois que le root bridge a été élu, le STA commence les meilleurs chemins vers le root bridge à partir de toutes les destinations du domaine de broadcast. Le coût de chaque chemin est déterminé par la somme des coûts de chaque port sur le chemin. Ce coût est inclus dans le BPDU. Le coût de chaque port est lié à son débit, et a donc une valeur par défaut, mais l'administrateur peut modifier ce coût, ce que lui donne de la flexibilité pour contrôler les chemins vers le root bridge.

Élection des ports root

Chaque switch non root doit choisir un port root. Il s'agit du port le plus proche du root bridge en termes de coût global. Ce coût global s'appelle le *coût interne du chemin root*.

Les chemins ayant des coûts les plus bas sont préférés, et tous les autres chemins redondants sont bloqués.

Élection des ports désignés

Une fois que chaque switch a choisi un port root, les switchs vont choisir des ports désignés. Chaque lien entre deux switchs aura un port désigné. Le port désigné sera celui ayant le meilleur chemin vers le root bridge.

Tous les ports du root bridge sont forcément des ports *désignés*, puisque le root bridge a le chemin le plus court vers lui-même.

Sur un lien entre deux switchs, si un port est *root*, l'autre sera forcément *désigné*. Tous les ports connectés à des hôtes sont aussi des ports *désignés*.

Les liens entre deux switchs qui ne sont pas root bridge peuvent n'avoir aucun port *root*. Dans ce cas c'est le port ayant le coût le plus faible qui sera *désigné*. Si les deux ports ont les mêmes coûts, le STA fait intervenir le BID.

Élection des ports alternatifs

Tous les ports qui ne sont ni *root* ni *désignés* deviennent des ports *alternatifs* et sont bloqués.

4.10.4 Minuteurs et états de ports

Le protocole STP comprend 3 minuteurs :

1. **Hello** — Intervalle entre deux envois de BPDU. Par défaut, 2 secondes, mais peut être défini sur une valeur entre 1 et 10 secondes.
2. **Forward Delay** — Temps qui est passé à écouter et à apprendre. Par défaut, 15 secondes, mais peut être défini sur une valeur entre 4 et 30 secondes.
3. **Max Age** — Temps maximum qu'un switch attend avant d'essayer de changer la topologie STP. Par défaut, 20 secondes, mais peut être défini sur une valeur entre 6 et 40 secondes.

Les temps par défaut des minuteurs peuvent être changées sur le root bridge, ce qui va dicter les valeurs dans tout le domaine STP.

Quand STP recalcule et change les état des ports bloqués pour qu'ils soient en état de transfert (forwarding), il ne faut pas qu'un port passe directement de l'état *bloqué* vers l'état *forwarding*. En effet, cela créerait temporairement une boucle. Pour cela, STP a cinq états de ports, dont 4 sont opérationnels. L'état non opérationnel est l'état *désactivé*.

1. *Disabled* (désactivé) — Avec `S1(config-if)# shutdown` par exemple. Ne participe pas au STP et ne transfère pas de trame.
2. *Blocking* (bloqué) — Port *Alternate* qui ne transfère pas de trame. Reçoit des BPDU pour déterminer où se trouve le root bridge. Si le port n'a pas reçu de BPDU de la part d'un voisin, il se met en état *blocking*.
3. *Listening* (en écoute) — En sortant de l'état *blocking*, le port reçoit des BPDU pour déterminer le chemin vers le root bridge. Le port envoie également des BPDU pour indiquer aux autres qu'il sort de l'état *blocking*.

4. *Learning* (en apprentissage) — Après l'état *listening*, le port reçoit des BPDU et se prépare à participer à la transmission de trames. Il commence à remplir la table MAC.
5. *Forwarding* (en fonctionnement) — Le port fait maintenant partie de la topologie active. Il transmet des trames et reçoit et envoie des BPDU.

4.10.5 Un peu d'histoire

Il y a en fait différentes versions de STP. Un nouveau standard utilise le RSTP et s'appelle le `IEEE-802-D-2004`.

Le standard original du STP est le `IEEE 802.1D`. C'est ce terme qu'on utilise dans les discussions si on veut parler du STP original, pour éviter la confusion.

Voici différentes versions du Spanning Tree :

— **STP** — Version originale `802.1D` (1998 et avant). On l'appelle aussi Common Spanning Tree (CST). N'assure qu'une instance de Spanning Tree pour la topologie, peu importe le nombre de VLAN.
— **PVST** — Amélioration Cisco du STP original. Il fait une instance de STP `802.1D` pour chaque VLAN de la topologie.
— **RSTP** — `IEEE 802.1w` Plus rapide comme son nom l'indique.
— `802.1D-2004` — Version mise à jour du standard STP qui incorpore le `802.1w`.
— **Rapid PVST** — Amélioration Cisco de RSTP qui utilise PVST et propose donc une instance séparée de `802.1w` par VLAN.
— **Multiple Spanning Tree Protocol (MSTP)** — Standard de l'IEEE inspirée par le Multiple Instance Spanning Tree Protocol (MISTP) de Cisco. Mappe plusieurs VLAN dans la même instance de Spanning Tree.
— **Multiple Spanning Tree (MST)** — Instance Cisco du MSTP.

Les versions Cisco Internetwork Operating System (IOS) depuis la version 15.0 utilisent PVST par défaut.

4.10.6 RSTP

RSTP et rétro-compatible avec STP `802.1D`. La terminologie reste la même, l'algorithme est le même. La différence est surtout dans la rapidité de la convergence. Dans un réseau bien architecturé, un changement de topologie de couche 2 peut être recalculé en quelques centaines de millisecondes. Un port `Alternate` peut passer en `Forwarding` directement sans attendre que le réseau ne converge.

Sur les switchs Cisco, le RSTP qui tourne est le Rapid PVST. Il prend donc en compte les différents VLAN.

Les états et rôles de ports RSTP sont plus simples que STP :

États	
STP	RSTP
Disabled Blocking Listening	Discarding
Learning	Learning
Forwarding	Forwarding

Rôles	
STP	RSTP
Root Port	Root Port
Designated Port	Designated Port
Blocked Port	Backup Port Alternate Port

Les ports RSTP `Backup` sont rares, ils servent à offrir un lien de secours vers un hub par exemple, mais les hubs ne devraient plus exister dans une topologie.

4.10.7 PortFast et BPDU Guard

Pour passer vers un état `Forwarding`, un port passe par les états `Listening` et `Learning`. À chaque fois, le port attend que le minuteur `Forward Delay` s'écoule, c'est-à-dire 15 secondes pour chaque état, donc 30 secondes en tout. Cela peut poser des problèmes pour le DHCP, privant un hôte d'adresse IPv4. En IPv6 ce n'est pas un problème parce que le routeur continue d'envoyer des messages Router Advertisement (RA).

Un port peut être configuré en `PortFast` pour passer à l'état `Forwarding` immédiatement sans passer par `Listening` et `Learning`. Cette configuration ne doit être faite que sur des ports `access`, c'est-à-dire connectés à des hôtes. Si on configure `PortFast` sur un trunk, donc un port connecté à un autre switch, on risque de créer une boucle.

Dans une configuration valide, un port en `PortFast` ne doit jamais recevoir de BPDU. Cela indiquerait que le port est connecté à un autre switch. Pour préserver de cette mauvaise configuration, Cisco propose BPDU Guard. Quand BPDU Guard est activé, il désactive immédiatement un port qui reçoit un BPDU et le met en mode `error-disabled`. Un administrateur doit alors manuellement réactiver le port.

4.11 EtherChannel

4.11.1 Principes

Imaginons qu'on ait besoin de d'avantage de bande passante que ce que notre interface peut offrir. Par exemple avec des interfaces à 1 Gbps, s'il nous faut une bande passante de 2 Gbps, on pourrait brancher les deux liens et profiter de la redondance. Mais le STP bloquera un des deux liens redondants pour prévenir des boucles.

EtherChannel permet de parer à ce problème en faisant une agrégation de liens. Il permet des liens redondants qui ne sont pas bloqués par STP.

EtherChannel permet la tolérance aux pannes, l'équilibrage de charge, une plus grande bande passante et la redondance entre switchs, routeurs et serveurs.

Au départ, EtherChannel a été développé par Cisco comme une technique LAN de switch à switch.

Il regroupe plusieurs port ethernet en un canal logique. L'interface vituelle résultant d'un EtherChannel s'appelle un canal de port.

Voici certains avantages d'EtherChannel :

— La plupart des configurations peuvent être faites sur l'interface EtherChannel au lieu de chaque port individuel.
— Pas besoin de mettre à niveau le port pour augmenter la bande passante, ce qui coûterait plus cher.
— Deux liens faisant partie du même EtherChannel font de l'équilibrage de charge.
— EtherChannel crée une agrégation qui apparaît comme un lien logique. Si le STP bloque un lien pour éviter une boucle, il bloque le lien EtherChannel entièrement, et donc tous les ports en faisant partie. S'il n'y a qu'un lien EtherChannel, STP laisse tous les ports actifs puisqu'il ne voit qu'un seul lien logique.
— EtherChannel permet la redondance. La perte d'un lien physique de l'EtherChannel ne crée pas de changement dans la topologie. Pas besoin de nouveau calcul STP. S'il y au moins un lien physique, l'EtherChannel continue de fonctionner, même si la bande passante baisse au sein de l'EtherChannel.

Par contre, il y a des restrictions au niveau de l'implémentation :

— On ne peut pas mélanger de types d'interface (pas de FastEthernet et GigabitEthernet dans un même EtherChannel).
— Chaque EtherChannel peut compter au maximum 8 ports compatibles. Cela veut dire que les débits peuvent aller jusqu'à 800 Mbps (Fast EtherChannel, 8×100 Mbps) ou 8 Gbps (Gigabit EtherChannel, 8×1 Gbps).
— Le switch Cisco Catalyst 2960 prend en charge 6 EtherChannels maximum. Avec le temps, de nouveaux IOS permettront probablement d'avantage dEtherChannel par switch et d'avantage de ports par EtherChannel.
— Les ports de chaque côté d'un lien faisant partie d'un EtherChannel doivent être configurés ensemble. Par exemple si les ports d'un côté sont configurés en trunk, l'autre côté doit être configuré en trunk aussi. Chaque port d'un EtherChannel doit également être de couche 2.
— Chaque EtherChannel ayant une interface de canal logique, une configuration appliquée à l'interface du canal affecte toutes les interfaces physiques de cet EtherChannel.

Il est possible de configurer des EtherChannels statiques mais il existe aussi deux protocoles d'auto-négociation :

1. PAgP
2. LACP

PAgP

C'est un protocole Cisco propriétaire qui aide à créer des liens EtherChannel automatiques. PAgP identifie les liens ethernet pour former un EtherChannel puis il ajoute l'EtherChannel au Spanning Tree comme un seul port.

PAgP envoie des paquets toutes les 30 secondes. Il s'assure que quand un EtherChannel est créé, tous ses ports ont la même configuration. Il s'occupe aussi de l'ajout de liens, ou de ruptures de liens.

PAgP a 3 modes :

1. **On** — Force l'EtherChannel sans PAgP. Les interfaces configurées en mode `on` n'échangent pas de paquet PAgP.
2. **PAgP desirable** — Place l'interface dans un mode PAgP de négociation active. L'interface envoie des paquets PAgP aux autres interfaces.
3. **PAgP auto** — Place l'interface dans un mode PAgP de négociation passive. L'interface répond aux paquets PAgP qu'il reçoit mais n'initie aucune négociation.

Les modes doivent correspondre des deux côtés. Si un côté est en mode `auto`, il est passif. Si en l'interface d'en face est aussi en mode `auto`, il n'y aura jamais de négociation et donc pas de création d'EtherChannel. Si tous les modes sont désactivés ou si aucun mode n'est configuré, l'EtherChannel est désactivé.

Le mode `on` ne marche que s'il est configuré des deux côtés.

LACP

LACP fait partie de la spécification IEEE 802.3ad qui autorise plusieurs ports physiques de se grouper pour former un seul canal logique. LACP permet à un switch de négocier un groupe automatique en envoyant des paquets LACP à l'autre switch.

C'est une fonction similaire à l'EtherChannel PAgP de Cisco, mais comme LACP est un standard de l'IEEE, il permet de créer des EtherChannels dans des topologies contenant plusieurs fabriquants. Les périphériques Cisco comprennent les deux protocoles.

La spécification LACP, originellement 802.3ad, est maintenant défini dans le 802.1AX.

LACP fonctionne de manière identique à PAgP. Il détecte les configurations de chaque côté du lien pour s'assurer qu'elles sont compatibles pour activer l'EtherChannel.

Les modes LACP sont les suivants :

1. **On** — Force l'interface à créer un canal sans LACP. Les interfaces sur le mode `on` n'échangent pas de paquet LACP.
2. **LACP active** — Place le port dans un état de négociation active. Le port envoie des paquets LACP.
3. **LACP passive** — Place le port dans un état de négociation passive. Le port répond aux paquets LACP reçus mais n'initie pas de négociation LACP.

Tout comme PAgP, les modes doivent être compatibles de chaque côté du lien pour former un EtherChannel.

LACP permet 8 liens actifs et 8 liens de secours. Un lien de secours devient actif si un lien actif échoue.

4.11.2 Configuration

Avant de vouloir regrouper des liens en EtherChannel, il faut s'assurer que :

— Toutes les interfaces prennent en charge l'EtherChannel.

— Toutes les interfaces d'un EtherChannel ont la même configuration de duplex et de débit.
— Toutes les interfaces d'un EtherChannel sont dans le même VLAN ou sont configurées en trunk.
— Dans un EtherChannel de trunk, toutes les interfaces de l'EtherChannel autorisent les mêmes VLAN.

Il est important de noter qu'un changement de configuration sur l'interface du canal s'appliquera sur toutes les interfaces qui forment ce canal, mais qu'un changement de configuration sur une interface individuelle ne s'appliquera pas sur le canal, ce qui peut donc causer des configurations incompatibles.

Par défaut, l'EtherChannel est désactivé et doit être configuré.

Il y a 3 étapes pour créer un lien EtherChannel :

1. Choisir les interfaces qui doivent composer l'EtherChannel.
   On utilise `range` pour configurer plusieurs interfaces à la fois et s'assurer qu'on leur applique la même configuration.
   Avant de créer l'EtherChannel, on désactive les ports physiques pour s'assurer qu'ils ne soient pas mis dans l'état `err-disabled`.
   ```
   Switch(config)# interface range <interface-ids>
   Switch(config-if-range)# shutdown
   ```
2. Créer l'interface du canal de port, puis réactiver les ports du `range`.
   ```
   Switch(config-if-range)# channel-group <id> mode active
   Switch(config-if-range)# no shutdown
   Switch(config-if-range)# exit
   ```
   `mode active` l'identifie comme une configuration LACP. On pourrait écrire `mode on`, `passive`, `auto`, `desirable`... en fonction du protocole ou de l'état que l'on veut.
3. Entrer dans l'interface de l'EtherChannel pour configurer le canal entièrement.
   ```
   Switch(config)# interface port-channel <id>
   ```

4.11.3 Vérification

Afficher l'état général de l'intreface du canal de port :

```
Switch# show interfaces port-channel <id>
```

Si plusieurs EtherChannels sont configurés sur le même périphérique, afficher une ligne d'informations par EtherChannel :

```
Switch# show etherchannel summary
```

Afficher des informations sur une interface de canal de port spécifique.

```
Switch# show etherchannel port-channel
```

Afficher les informations d'une interface faisant partie d'un EtherChannel :

```
Switch# show interfaces f0/1 etherchannel
```

4.12 Attaques de réseau LAN

4.12.1 Attaque de table MAC

Les switchs ont des tables MAC (ou cache ARP) à taille fixe, ce qui veut dire que cette table peut être remplie.

Une attaque de table MAC (ou ARP-cache poisoning) consiste à faire en sorte que cette table soit remplie pour s'assurer que le switch ne vérifie plus sa table MAC en recevant des nouvelles trames.

Un outil pour effectuer cette attaque s'appelle `macof`. L'attaque se fait en 4 étapes :

1. L'acteur de la menace se connecte au VLAN et utilise `macof` pour générer de nombreuses adresses MAC avec des adresses IP source et destination aléatoires.
2. Cela remplit la table MAC du switch rapidement.
3. Une fois la table MAC pleine, le switch va systématiquement renvoyer toutes les trames qu'il reçoit par tous ses ports. Tant que `macof` continue d'inonder la table pour s'assurer qu'elle ne désemplisse pas, le switch continue de rediriger toute trame entrante par chaque port associé au VLAN en question.
4. L'acteur de la menace n'a plus qu'à sniffer le réseau (avec Wireshark par exemple) pour récupérer toutes les trames de tous les appareils du réseau local.

Dès que `macof` arrête d'inonder la table ARP, celle-ci se désemplit et remplace les fausses entrées par celles reçues de la manière normale. Le switch peut alors refonctionner normalement.

Pour mitiger ce genre d'attaque, il faut implémenter une sécurité des ports. Les adresses MAC qui vont remplir la table du switch sont définies à l'avance et en nombre limité.

4.12.2 Attaque par saut de VLAN

Cette attaque permet au trafic d'un VLAN d'être vu par un autre VLAN sans l'aide d'un routeur. Pour effectuer cette attaque il faut configurer un hôte pour qu'il agisse comme un switch. L'hôte a alors une fonction activée par défaut : le trunking automatique. Le switch établit une liaison de trunk avec l'hôte, lui permettant d'accéder à tous les VLAN du switch. Ensuite l'acteur de la menace peut envoyer du trafic sur n'importe quel VLAN.

4.12.3 Attaque de double marquage VLAN

Dans certaines situations un acteur de la menace pourrait ajouter un marqueur `802.1Q` dans une trame qui contient déjà un tel marqueur. Ceci permet à la trame d'accéder à un VLAN que le marqueur `802.1Q` orginal ne spécifiait pas.

Le premier marqueur indique le VLAN de l'attaquant, qui sera considéré comme VLAN natif par le premier switch rencontré. Le deuxième marqueur indique le VLAN de la victime. Le switch retire le premier marqueur avant de renvoyer la trame par tous les ports de ce VLAN. La trame n'est pas marquée de nouveau parce qu'il s'agit du VLAN natif. Le premier switch n'a donc pas conscience du deuxième marqueur injecté par l'acteur de la menace. Ainsi, quand la trame arrive au deuxième switch, celui-ci ne connaît pas le

VLAN d'origine. Le deuxième switch va donc envoyer la trame par les ports du deuxième VLAN.

Cette attaque est unidirectionnelle et ne fonctionne que quand l'attaquant est connecté à port sur le même VLAN que VLAN natif du trunk.

Pour mitiger cette attaque, ainsi que le saut de VLAN vu précédemment, il faut suivre des principes de sécurité concernant les trunks :

— désactiver le trunk sur tous les ports
— désactiver l'auto-trunking sur les liens de trunk pour que les trunks soient forcément activés manuellement
— être sûr que le VLAN natif n'est utilisé que pour les liens de trunk

4.12.4 Attaque DHCP

Il y a deux types d'attaques DHCP : la famine DHCP et l'usurpation DHCP.

1. **DHCP starvation** — Le but est de créer un Denial of Service (DoS). Cette attaque a besoin d'un outil comme `Gobbler`. Cet outil peut scanner toute l'étendue des adresses IP du serveur DHCP et tente d'établir un bail pour chacune d'entre elles. `Gobbler` crée des messages DHCP discovery avec des adresses MAC fictives.
2. **DHCP spoofing** — Cette attaque utilise un serveur DHCP non autorisé et connecté au réseau. Ce serveur fournit des mauvaises configurations IP à des clients légitimes. Il peut fournit les mauvaises informations suivantes :
   — Passerelle par défaut, pour faire une attaque Man-in-the-Middle (MitM).
   — Serveur DNS, qui peut rediriger l'utilisateur vers un site web malicieux.
   — Adresse IP, créant un DoS sur le client DHCP.

4.12.5 Attaques ARP

Quand une adresse IPv4 de destination n'est pas dans une table MAC d'un hôte, il va envoyer une requête ARP en broadcast. Tous les hôtes du réseau local reçoivent cette requête ARP. Une requête ARP non sollicitée s'appelle un ARP gratuit.

Un attaquant peut donc envoyer à un switch un ARP gratuit avec une adresse MAC usurpée. Le switch va alors mettre à jour sa table ARP en fonction. Cela veut dire que n'importe quel hôte peut mentir à propos de la combination de ses adresses MAC et IP.

Une attaque peut consister en envoyant des réponses ARP non sollicitées avec son adresse MAC et l'adresse IP de la passerelle par défaut. Des outils permettant de faire ça comprennent `dsniff`, `Cain & Abel`, `ettercap`, `Yersinia`, et d'autres.

La mitigation de ces attaques consiste à implémenter un Dynamic ARP Inspection (DAI).

4.12.6 Attaque STP

Il est possible de manipuler le protocole STP pour changer la topologie d'un réseau. Là aussi le but est une attaque MitM. L'attaquant redirige tout le trafic par son poste en se faisant passer pour un switch prioritaire dans le protocole STP. Pour cela il envoie des

BPDU en broadcast contenant des configurations forçant de nouveaux calculs STP. Ces BPDU ont une priorité plus basse dans le but d'être choisi comme le bridge root.

Pour mitiger cette attaque il faut implémenter BPDU Guard sur tous les ports d'accès.

4.12.7 Reconnaissance CDP

Cisco Discovery Protocol (CDP) est un protocole Cisco de découverte de voisin, activé par défaut sur tous les périphériques Cisco. Il implique que ces périphériques s'échangent des informations de configuration sur le réseau. Ces informations sont envoyées périodiquement sur tous les ports où CDP est activé, et cela de manière non chiffrée dans un multicast.

Ces informations contiennent :

- l'adresse IP du périphérique
- la version de l'IOS
- la plateforme
- ses capacités
- le VLAN natif

Évidemment, un attaquant peut utiliser ces informations pour découvrir des vulnérabilités d'infrastructure de réseau. Il pourrait aussi envoyer des trames CDP contenant de fausses informations de configuration, ce qui aura pour effet de reconfigurer les équippements à proximité et de changer la topologie du réseau.

Pour mitiger ces attaques, il faut limiter l'utilisation du CDP. On peut par exemple désactiver le CDP sur les ports connectés à des périphériques non fiables.

Pour désactiver CDP globalement sur un périphérique :

```
S1(config)# no cdp run
```

Pour désactiver CDP sur un port :

```
S1(config-if)# no cdp enable
```

Chapitre 5

Couche Réseau

5.1 Adressage IP

L'adressage IP permet la communication entre hôtes n'étant pas forcément présents sur le même réseau.

Aujourd'hui, deux versions du protocole IP coexistent : IPv4 et IPv6.

5.1.1 Attribution des adresses IP

Les adresses publiques devant être uniques, leur utilisation est régulée par l'IANA. L'IANA gère des blocs d'adresses IP et les attribue aux organismes d'enregistrement régionaux (RIR).

5.2 IPv4

5.2.1 Présentation

Les ordinateurs travaillent en binaire, mais nous représentons les adresses IPv4 en notation *décimale pointée*.

Une adresse IPv4 consiste en 32 bits représentés par 4 nombres représentant chacun un octet. Cette adresse est toujours accompagnée de son *masque*, lui aussi représenté sous forme de quatre octets. Un masque a toujours des bits consécutifs à 1 à gauche et à 0 à droite. Les bits à 1 recouvrent la partie réseau de l'adresse IPv4 et les bits à 0 recouvrent la partie machine.

5.2.2 Classes

Aujourd'hui, les classes sont obsolètes mais à l'origine on utilisait ces classes.

Classe IPv4	plage d'adresses	masque	CIDR
A	0.0.0.0 à 127.255.255.255	255.0.0.0	/8
B	128.0.0.0 à 191.255.255.255	255.255.0.0	/16
C	192.0.0.0 à 223.255.255.255	255.255.255.0	/24
D	224.0.0.0 à 239.255.255.255	non défini	non défini
E	240.0.0.0 à 255.255.255.255	non défini	non défini

Quelques adresses spéciales à retenir :

- Les adresses des plages 10.0.0.0 /8, 172.16.0.0 /12 et 192.168.0.0 /16 sont des adresses réservées par la RFC 1918 pour un usage privé (adresses non routables).
- Les adresses de la plage 127.0.0.0 /8 sont utilisées pour le rebouclage (loopback).
- Les adresses de la plage 169.254.0.0 /16 sont attribuées automatiquement par l'OS et sont des adresses link-local.
- Enfin, les adresses de la classe D sont utilisées pour le multicast.

Le principe des classes ayant gaspillé beaucoup d'adresses publiques, on utilise depuis les années 90 un système sans classe : le routage Classless Inter-Domain Routing (CIDR).

5.2.3 Monodiffusion, diffusion, multidiffusion

Un hôte à trois façons de communiquer en IPv4 :

1. Monodiffusion (unicast) : envoi d'un paquet d'un hôte à un autre. Les adresses unicast couvrent les classes A, B et C (bien que certaines plages soient réservées à un usage spécifique).
2. Diffusion (broadcast) envoi d'un paquet d'un hôte à tous les autres hôtes du réseau. Un routeur ne transfère pas les broadcasts.
3. Multidiffusion (multicast) envoi d'un paquet d'un hôte à un groupe d'hôtes spécifiques qui peuvent se trouver sur différents réseaux. Les adresses multicast sont dans la classe D.

5.2.4 Adresses privées et adresses publiques

La Request For Comments (RFC) 1918 a reservé trois blocs d'adresses IPv4 qui ne sont pas routées sur internet :

1. `10.0.0.0 /8` → `10.0.0.0` à `10.255.255.255`
2. `172.16.0.0 /12` → `172.16.0.0` à `172.31.255.255`
3. `192.168.0.0 /16` → `192.168.0.0` à `192.168.255.255`

Ces adresses doivent être traduites en adresses IPv4 publiques. Cette traduction s'effectue par la NAT en général sur le routeur qui connecte le réseau interne à celui du FAI.

5.2.5 Adresses spéciales

Voici d'autres adresses spéciales :

— Adresses de rebouclage (localhost) : `127.0.0.0 /8` → `127.0.0.1` à `127.255.255.254`
Utilisées par les hôtes pour rediriger le trafic vers eux-mêmes.
— Adresses locales-liens (Automatic Private IP Addressing (APIPA)) : `169.254.0.0 /16` → `169.254.0.1` à `169.254.255.254`
Utilisées par un client DHCP Windows si aucun serveur DHCP n'est disponible.
— Adresses TEST-NET : `192.0.2.0 /24` → `192.0.2.0` à `192.0.2.255`
Réservées à des fins pédagogiques et utilisées dans la documentation ou dans des exemples réseau.

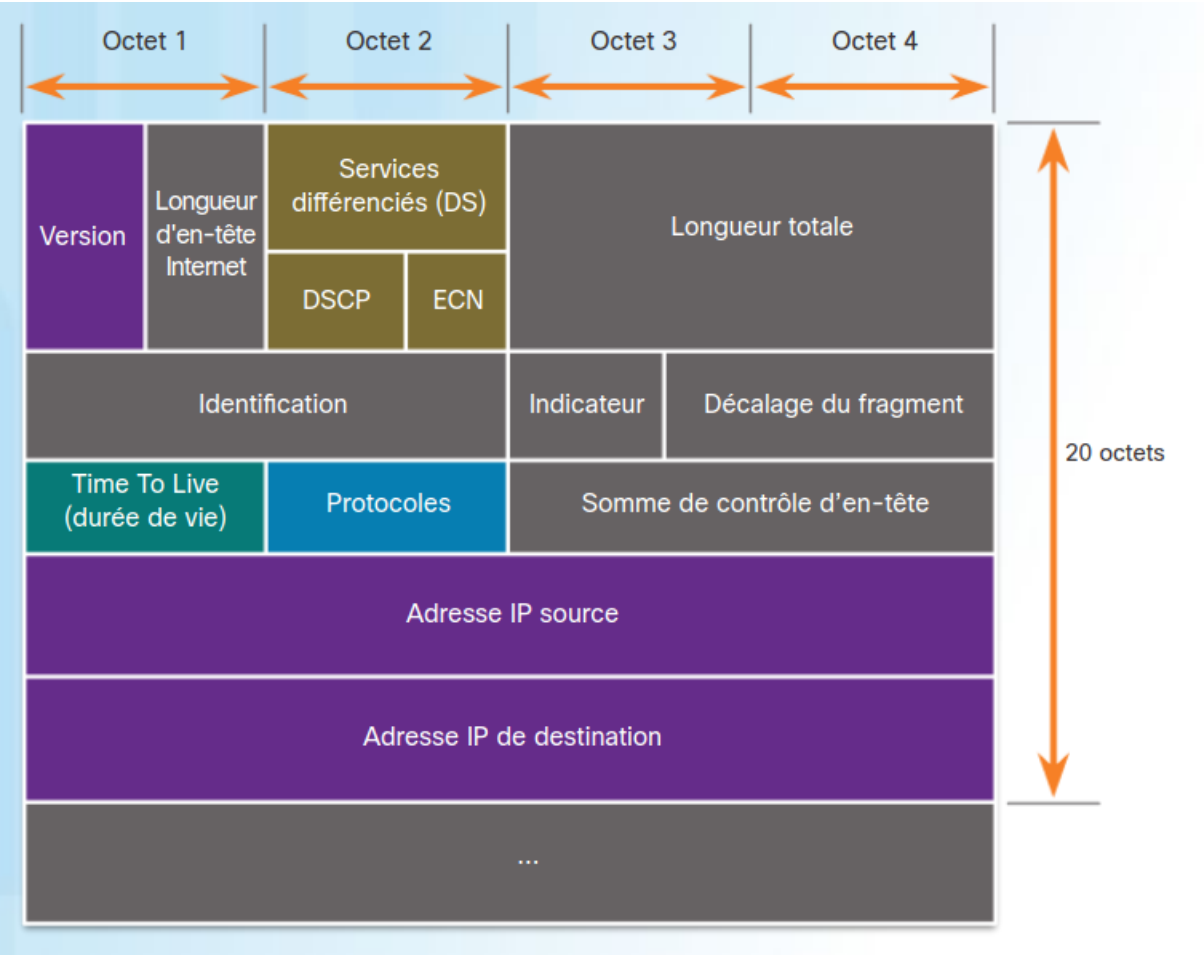
Enfin, noter que la classe D est réservée pour les multicasts, et que la classe E est réservée pour une utilisation expérimentale.

5.2.6 Calculs IPv4

— Pour trouver l'adresse de broadcast :
B = octet du réseau + (255 − octet du masque)
On effectue l'opération octet par octet donc 4 fois.
— Pour trouver le nombre d'hôtes possibes :
$= 2^{(32-CIDR)} - 2$
exemple : /26 : $2^{(32-26)} - 2 = 2^6 - 2 = 64 - 2 = 62$ hôtes
— Pour trouver le nombre de sous-réseaux :
$= 2^{(CIDR-masque\ natif)}$
exemple : 10.x.y.z /26 $= 2^{(26-8)} = 2^{18}$ sous-réseaux

5.2.7 En-tête IPv4

L'en-tête IPv4 fait 20 octets (jusqu'à 60 avec le champ `Options` facultatif).



Voici les champs les plus importants :

- Version : contient une valeur de 4 bits indiquant qu'il s'agit d'un paquet IPv4. La valeur est `0100` pour IPv4.
- Differentiated Services (DS) : anciennement appelé `type de service`. Champ de 8 bits utilisé pour définir la priorité de chaque paquet.
  Ce champ est en deux parties :
  1. Differentiated Services Code Point (DSCP) sur 6 bits
  2. Explicit Congestion Notification (ECN) sur 2 bits
- TTL : une valeur de 8 bits utilisée pour limiter la durée de vie d'un paquet. Il diminue d'un point à chaque fois que le paquet est traité par un routeur. Si la valeur arrive à zéro, le routeur rejette le paquet et envoie un message de dépassement du délai ICMP à l'adresse IP source.
- Protocole : valeur de 8 bits qui indique le type de données transportées par le paquet. Les valeurs les plus courantes sont ICMP (1), TCP (6) et UDP (17).
- Adresse IPv4 source : valeur de 32 bits. L'adresse IP source est toujours une adresse unicast.
- Adresse IPv4 de destination : valeur de 32 bits.

5.3 IPv6

5.3.1 Raison

Dès les années 80 on s'est rendu compte qu'il allait y avoir une pénurie d'adresses IPv4. L'IPv6 est prêt dans les années 90.

Au lieu de le mettre en place tout de suite on a eu plusieurs mécanismes :

— NAT
— classes privées
— agrégation d'adresses dans les tables de routage

Aujourd'hui on n'est plus en pénurie, on est carrément en épuisement. Les adresses IPv4 ont été épuisées en Asie dès 2011 et en Europe fin 2012. Mais personne n'a vraiment envie de déployer l'IPv6.

5.3.2 Quelques éléments nécessaires avec IPv4 qui ne le sont plus

L'IPv6 ne se contente pas de fournir un plus grand nombre d'adresses. Il améliore le protocole ICMP : ICMPv6 inclut la configuration automatique et la résolution d'adresse. Les protocoles ARP et NAT ne sont plus nécessaires.

5.3.3 Coexistence avec IPv4

La transition vers l'IPv6 ne se faisant pas du jour au lendemain, plusieurs méthodes permettent de faire cohabiter les deux protocoles IP.

— *double pile* : permet de faire coexister l'IPv4 et l'IPv6 sur un même segment de réseau. Les périphériques doubles pile sont capable d'exécuter les piles de protocoles IPv4 et IPv6 simultanément.
— *tunneling* : méthode de transport de paquets IPv6 via un réseau IPv4. Un paquet IPv6 est encapsulé dans un paquet IPv4 de la même manière que d'autres types de données.
— *traduction* : les périphériques IPv6 peuvent utiliser la traduction NAT64 pour communiquer avec des périphériques IPv4. La technique de traduction est similaire à la NAT pour IPv4. Un paquet IPv6 est traduit en un paquet IPv4 et inversement.

5.3.4 Représentation

L'adresse IPv6 est représentée en hexadécimal sur 16 octets (128 bits). L'IPv6 possède 340 sextillions d'adresses disponibles (340 suivi de 36 zéros).

0000	:	0000	:	0000	:	0000	:	0000	:	0000	:	0000	:	0000	
à		à		à		à		à		à		à		à	
ffff	:	ffff	:	ffff	:	ffff	:	ffff	:	ffff	:	ffff	:	ffff	

Deux chiffres hexadécimaux font $16 \times 16 = 256$ possibilités donc un octet. Chacun des huit groupes de l'adresse IPv6 peut donc s'appeler un hextet (quatre chiffres hexadécimaux).

Les lettres ne sont pas sensibles à la casse (on peut les écrire en majuscules ou en minuscules).

— On peut omettre les 0 qui commencent un groupe :
`0660` → `660`
`fe80:0000:56ad:...` → `fe80:0:56ad:...`
— Quand plusieurs groupes de 0 se suivent dans l'adresse, on peut omettre le groupe complètement en le remplaçant par " : :".
Ainsi, quand on lit une adresse IPv6 et qu'on voit " : :", cela veut dire qu'il faut remplir avec autant de zéros qu'il faut pour avoir 128 bits.
Il ne peut y avoir qu'un seul " : :" par adresse IPv6.

Enfin, pour noter le masque de sous-réseau, on utilise la longueur de préfixe avec le slash ou le pourcentage. Elle peut être comprise entre 0 et 128. La longueur standard est /64.

Cela veut dire que la partie réseau de l'adresse a une longueur de 64 bits, ce qui en 64 pour la partie hôte.

5.3.5 Types d'adresses

Comme l'IPv4, une adresse IPv6 peut être unicast ou multicast. Mais contrairement à l'IPv4, l'IPv6 n'a pas d'adresse de broadcast. Il exite une adresse multicast destinée à tous les nœuds IPv6 et qui offre globalement les mêmes résultats.

L'IPv6 possède également les adresses anycast, qui sont des adresses unicast pouvant être attribuées à plusieurs périphériques.

Une carte réseau peut avoir plusieurs adresses IPv6 qui ont chacune un usage précis.

— *Adresse unicast globale* (`2000::/3` — Global Unicast Address (GUA)) :
Similaire à une adresse IPv4 publique. Les adresses globales de monodiffusion (GUA) sont routables sur internet et doivent donc être uniques.
— *Adresse de liaison locale (link-local)* (`fe80::/10` — Link Local Address (LLA)) :
Ne peut être utilisée que pour des communications sur le réseau local. Si on doit passer par un routeur, on ne peut pas utiliser cette adresse.
Si elle n'est pas configurée manuellement, le périphérique crée automatiquement sa propre adresse *link-local* sans passer par un serveur DHCP.
— *Adresse locale unique* (`fc00::/7` — Unicast Local Address (ULA)) :
Équivalant aux adresses privées IPv4 mais il n'y a plus de NAT.
Une adresse locale unique peut être utilisée pour un périphérique qui n'aura jamais besoin d'être accessible sur un autre réseau. Contrairement aux adresses *link-local*, qui ne peuvent pas être routées du tout, les adresses *locales uniques* peuvent être routées dans un domaine de routage (elles ne pourront juste pas sortir sur internet).
— *Adresses multicast* (`ff00::/8`) :
Même principe qu'en IPv4.
— *IPv4 intégrée* :
Intègre une adresse IPv4 pour communiquer grâce au tunneling.
— *Adresses de rebouclage* (`::1/128`) :
Équivalent à `127.0.0.1` en IPv4.
— *Adresse non spécifiée* (`::/128`) :
Adresse que prend la carte réseau avant de s'attribuer une adresse.

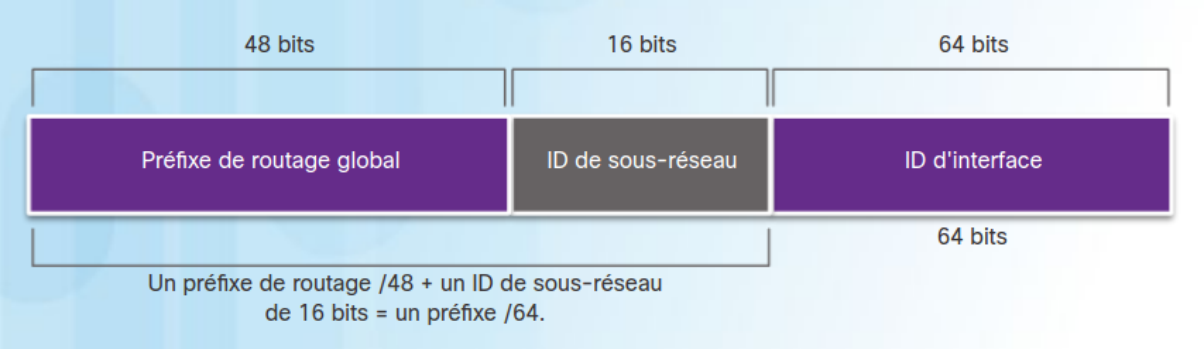
5.3.6 Structure d'une adresse globale

Actuellement, seules les adresses dont les trois premiers bits sont 001 (`2000::/3`) sont attribuées. Le premier chiffre hexadécimal d'une GUA commence donc par 2 ou 3.

Préfixe de routage global

Le préfixe de routage global est attribué par le fournisseur (comme un FAI) à un client ou un site. C'est la partie réseau de l'adresse.

Généralement, les Regional Internet Registry (RIR) attribuent un préfixe /48.



La taille du préfixe global de routage détermine la taille de l'ID de sous-réseau.

ID de sous-réseau

L'ID de sous-réseau est utilisé par une entreprise pour identifier les sous-réseaux de son site. Plus l'ID est un nombre important, plus il y a de sous-réseaux disponibles.

ID d'interface

C'est l'équivalent de la partie hôte de l'adresse IPv4. Le terme a changé parce qu'en IPv6 un hôte unique peut avoir plusieurs interfaces, chacune dotée d'une ou plusieurs adresses IPv6.

Il est recommandé d'utiliser des sous-réseaux /64, donc un ID d'interface 64 bits.

Pas d'adresse de broadcast ou de réseau

Contrairement à l'IPv4, l'IPv6 n'utilise pas d'adresses de diffusion. Ainsi les ID d'interface contenant uniquement des 1 peuvent être utilisés. Les adresses contenant uniquement des 0 sont réservées comme adresses de routeur de sous-réseau et ne doivent être attribuées qu'aux routeurs.

5.3.7 Conversion IPv4 vers IPv6

L'IPv6 étant compatible avec l'IPv4, il est possible de représenter une adresse IPv4 en IPv6.

L'adresse IPv4 sera entièrement comprise dans les deux derniers groupes de l'adresse IPv6 (deux chiffres hexadécimaux par octet). On peut donc passer une adresse IPv4 en IPv6 comme ceci :

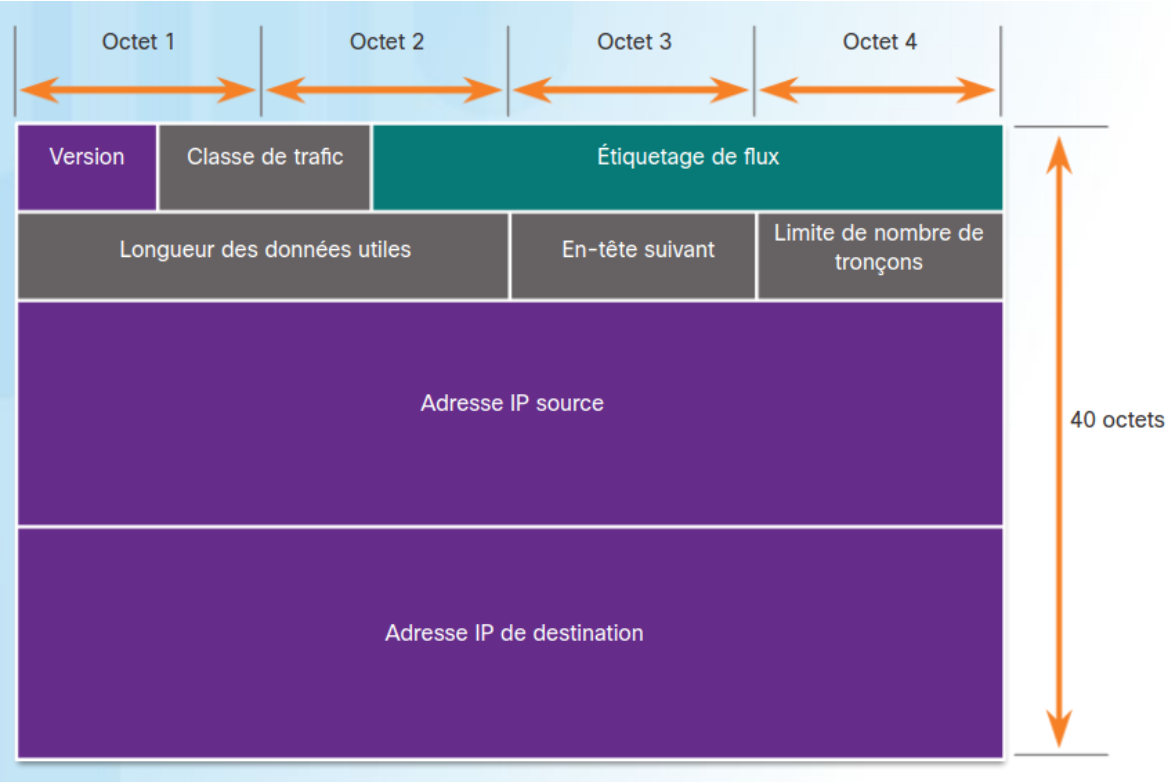
— 192.168.0.1
0000 :0000 :0000 :0000 :0000 :ffff :c0a8 :0001
192 ÷ 16 = 12 reste 0 (c0)
168 ÷ 16 = 10 reste 8 (a8)
0 ÷ 16 = 0 reste 0 (00)
1 ÷ 16 = 0 reste 1 (01)

C'est-à-dire 10 octets à 0, 2 octets à 1 (`FFFF`), et 4 octets pour l'adresse IPv4.

5.3.8 Calculs IPv6

5.3.9 En-tête IPv6

L'en-tête IPv6 fait 40 octets.



— Version : contient une valeur de 4 bits indiquant qu'il s'agit d'un paquet IPv6. La valeur est `0110` pour IPv6.
— Classe de trafic : champ de 8 bits qui est l'équivalent du champ DS pour l'IPv4.
— Étiquetage du flux : champ de 20 bits qui indique que tous les paquets portant la même étiquette de flux doivent être traités de la même manière par les routeurs.
— Longueur des données utiles : champ de 16 bits qui indique la longueur de la partie données du paquet IPv6.
— En-tête suivant : champ de 8 bits qui est l'équivalent du champ de protocole de l'IPv4. Il indique le type de données transportées par le paquet, c'est-à-dire le protocole de la couche supérieure.
— Limite du nombre de tronçons : champ de 8 bits qui remplace le champ TTL de l'IPv4. De la même façon, cette valeur est décrémentée à chaque fois que le paquet passe un routeur. Si la valeur atteint 0, le paquet est rejeté et un message ICMPv6 de délai dépassé est transféré à l'hôte émetteur.

— Adresse IPv6 source : champ de 128 bits contenant tout simplement l'adresse IPv6 de l'hôte à l'origine du paquet.
— Adresse IPv6 destination : champ de 128 bits contenant l'adresse IPv6 de l'hôte auquel le paquet est destiné.

À première vue, cet en-tête IPv6 est une simplification de l'en-tête IPv4 : moins de champs et une longueur plus courte, d'autant plus que la longueur de l'en-tête IPv6 est principalement due à la longueur des adresses IPv6.

Mais un paquet IPv6 peut également contenir des en-têtes d'extension qui fournissent des informations facultatives de couche Réseau. Ces en-têtes d'extension, lorsqu'ils sont présents, sont placés entre l'en-tête IPv6 et les données utiles. Il sont utilisés pour la fragmentation, la sécurité, la prise en charge de la mobilité, etc.

Contrairement à IPv4, les routeurs ne fragmentent pas les paquets IPv6.

5.3.10 Attribution des adresses IPv6

Il y a deux possibilités pour obtenir une adresse IPv6.

1. Il y a un serveur DHCPv6 sur le réseau.
2. Il n'y en a pas, alors c'est l'ordinateur qui la génère, par exemple grâce à l'adresse MAC.

Pour obtenir leur configuration automatiquement, les périphériques se basent sur les messages d'annonce de routeur ICMPv6 du routeur local. Ce routeur envoie des messages d'annonce de routeur ICMPv6 toutes les 200 secondes à tous les périphériques IPv6 du réseau ou en réponse à un hôte ayant envoyé un message de sollicitaion de routeur ICMPv6.

Ce message d'annonce contient les éléments suivants :

— *le préfixe de réseau et la longueur de préfixe*
— *l'adresse de la passerelle par défaut* qui est une adresse link-local
— *les adresses DNS et le nom de domaine*

Pour recevoir la configuration automatiquement, il y a trois options :

1. SLAAC uniquement
2. SLAAC avec une serveur DHCPv6 stateless
3. DHCPv6 stateful (pas de SLAAC)

Par défaut, le routeur envoie son message d'annonce en utilisant l'option 1, SLAAC uniquement. Mais l'interface du routeur peut être configurée pour envoyer une annonce de routeur à l'aide des méthodes SLAAC et DHCPv6 stateless ou DHCPv6 uniquement.

Configuration dynamique — SLAAC uniquement

Cette méthode permet à un périphérique d'obtenir sa configuration à partir d'un routeur IPv6 sans passer par un serveur DHCPv6.

Avec la SLAAC, aucun serveur central n'assure l'attribution d'adresses et la tenue à jour d'une liste des périphériques. Le périphérique client utilise les informations du message d'annonce de routeur pour créer sa propre adresse globale. Il utilise pour cela le préfixe reçu dans le message d'annonce de routeur et l'ID d'interface, qui utilse la méthode EUI-64 (son adresse MAC) ou est généré aléatoirement.

Configuration dynamique — SLAAC et DHCPv6 stateless

Avec l'option SLAAC et DHCPv6 stateless, le message d'annonce du routeur suggère aux périphériques d'utiliser :

— la SLAAC pour générer sa propre adresse globale
— l'adresse link-local du routeur comme passerelle par défaut
— un serveur DHCPv6 stateless pour obtenir d'autres informations comme le DNS et le nom de domaine

Un serveur DHCPv6 stateless ne distribue pas d'adresse globale.

Configuration dynamique — DHCPv6 stateful

Le DHCPv6 stateful est similaire au DHCP pour IPv4.

Avec cette option, le message d'annonce de routeur suggère aux périphériques d'utiliser :

— l'adresse link-local du routeur comme passerelle par défaut (il s'agit de l'adresse IPv6 source du message d'annonce)
— un serveur DHCPv6 stateful pour obtenir une adresse globale et les informations DNS

Le serveur DHCPv6 stateful tient également à jour une liste de ses attributions. Il ne fournit pas l'adresse de la passerelle par défaut.

5.3.11 Créer son adresse IPv6 globale

Avec la méthode SLAAC seule ou avec DHCPv6 stateless, le périphérique aura reçu la partie préfixe de son adresse globale. Il doit maintenant générer son ID d'interface.

Méthode EUI-64

Ce processus utilise l'adresse MAC à 48 bits et insère 16 autres bits au milieu de cette adresse MAC.

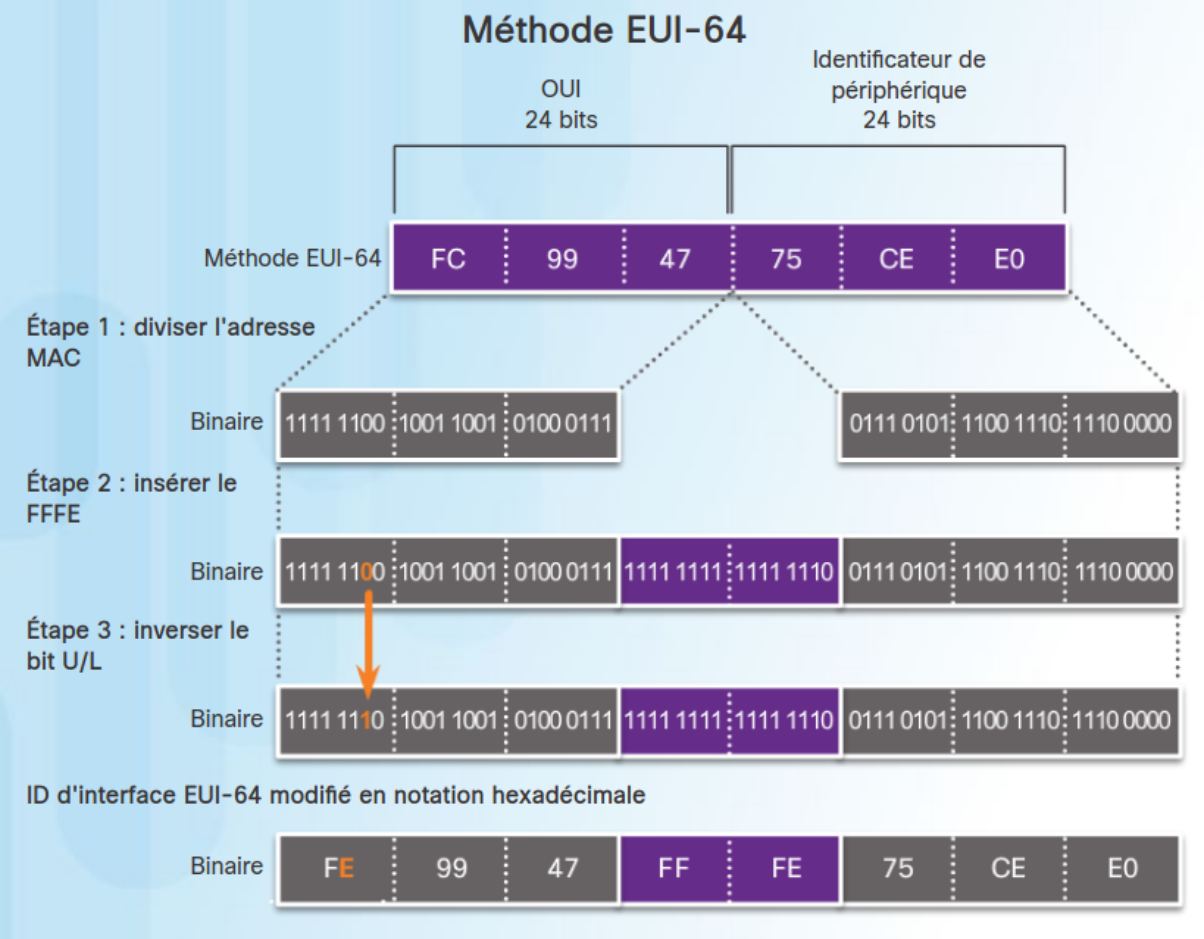
Pour rappel, l'adresse MAC est en deux parties, chacune sur 3 octets :

1. l'OUI, code de fournisseur attribué par l'IEEE
2. l'ID de périphérique, une valeur unique

L'ID d'interface de 64 bits comprend donc trois parties :

1. le code OUI provenant de l'adresse MAC mais dont le septième bit (Universal/Local (U/L)) est inversé
2. 16 bits correspondant à la valeur `FFFE` en hexadécimal

3. ID de périphérique de l'adresse MAC



Pour des raisons de confidentialité il est également possible de générer l'ID d'interface aléatoirement.

ID d'interface généré aléatoirement

Windows prévilégie cette méthode depuis la version Vista.

Pour s'assurer que l'adresse globale ainsi créée est unique, le client peut utiliser le processus de détection d'adresse dupliquée (Duplicate Address Detection (DAD)). Cela ressemble à une requête ARP pour sa propre adresse : en l'absence de réponse, l'adresse est unique.

5.3.12 Adresses link-local

Tous les périphériques IPv6 doivent avoir une adresse link-local.

Elle peut être créée dynamiquement à partir du préfixe `FE80::/10` et de l'ID d'interface à l'aide de la méthode EUI-64 ou d'un nombre aléatoire. En général les systèmes d'exploitation utilisent la même méthode pour les adresses globales et link-local.

Les routeurs Cisco créent l'adresse link-local dès qu'une adresse globale est attribuée à l'interface.

Comme les adresses link-local n'ont besoin d'être uniques que sur le réseau local, il n'est pas pratique de l'attribuer automatiquement sur un routeur en raison de la difficulté de

lecture des adresses IPv6. Il est courant de configurer les adresses link-local de manière statique sur les routeurs. Ainsi l'adresse sera reconnaissable et plus facile à mémoriser. Cela est d'autant plus pratique que ces adresses sont utilisées comme passerelles par défaut.

On peut par exemple configurer sur le routeur `R1` l'adresse `fe80::1` en link-local sur *chaque* liaison, car elle ne doit être unique que sur cette liaison. De la même façon, on mettrait `fe80::2` sur `R2`.

5.3.13 Adresses multicast

Il existe deux types d'adresses multicast IPv6 :

1. les adresses multicast attribuées
2. les adresses multicast de nœud sollicité

Adresses multicast attribuées

Réservées à des groupes ou périphériques prédéfinis. C'est une adresse unique.

Les plus courants sont les deux groupes suivants :

1. à tous les nœuds (`ff02::1`) :
   Tous les périphériques IPv6 peuvent rejoindre ce groupe. Un paquet envoyé à ce groupe est accepté par toutes les interfaces IPv6 situées sur le réseau.
   Cette opération a le même effet qu'un broadcast IPv4. Le message d'annonce de routeur (RA) ICMPv6 utilise cette adresse multicast.
2. à tous les routeurs (`ff02::2`) :
   Tous les routeurs IPv6 peuvent rejoindre ce groupe. Un routeur rejoint ce groupe lorsqu'il est activé en tant que routeur IPv6 avec la commande `R1(config)# ipv6 unicast-routing` Un paquet envoyé à ce groupe est accepté par tous les routeurs IPv6 situés sur le réseau.
   Les périphériques IPv6 envoient leur message de sollicitation (Router Solicitation (RS)) à cette adresse.

Adresses multicast de nœud sollicité

Elle est comparable à une adresse multicast à tous les nœuds. Sauf qu'elle est mappée à une adresse ethernet multicast. Cela permet à la carte réseau de filtrer la trame en examinant l'adresse MAC de destination pour accepter ou non la trame sans l'envoyer à la couche Réseau.

5.4 ICMP

Les protocoles ICMPv4 et ICMPv6 permettent d'envoyer des messages quand des erreurs se produisent.

Les messages ICMP communs à la version 4 et 6 sont notamment les suivants :

— `Host confirmation` :
Un message ICMP Echo permet de déterminer si un hôte est joignable. L'hôte local envoie un message ICMP Echo Request à un autre hôte. Si celui-ci est disponible, il répond avec une réponse d'Echo.
Cette utilisation est à la base de la commande `ping`.
— `Destination or Service Unreachable` :
Envoyé par un hôte ou une passerelle qui ne peut pas acheminer un paquet reçu. Le message comprend un code de destination. Pour l'ICMPv4 :
— 0 — Réseau inaccessible
— 1 — Hôte inaccessible
— 2 — Protocole inaccessible
— 3 — Port inaccessible
En ICMPv6 les codes sont légèrement différents.
— `Time exceeded` :
En IPv4, le champ TTL du paquet a atteint 0. En IPv6, c'est le champ de limite de nombre de tronçons qui est utilisé.
— `Route redirection`

Le protocole ICMPv6 ajoute quelques fonctionnalités à l'ICMPv4. Les messages ICMPv6 sont encapsulés dans l'IPv6.

ICMPv6 offre quatre nouveaux protocoles dans le cadre du protocole Neighbour Discovery Protocol (NDP, ND) :

1. Message de sollicitation de routeur (RS)
2. Message d'annonce de routeur (RA)
3. Message de sollicitation de voisin (Neighbour Solicitation (NS))
4. Message d'annonce de voisin (Neighbour Advertisement (NA))

Les deux derniers (entre voisins) sont utilisés pour la résolution d'adresse et la DAD.

— *Résolution d'adresse*
Équivalent du protocole ARP, utilisé quand l'hôte connaît l'adresse IPv6 de son destinataire, mais pas son adresse MAC. Le voisin qui reconnaît son adresse IPv6 répond avec son adresse MAC.
— *DAD*
Lorsqu'une adresse IPv6 globale ou link-local est attribuée, il est recommandé de faire une DAD pour s'assurer que l'adresse est unique. L'hôte envoie un message de sollicitation de voisin avec sa propre adresse IPv6. Si quelqu'un lui répond avec une adresse MAC, l'adresse n'est pas unique.
Elle n'est pas obligatoire mais recommandée par la RFC 4861.

5.5 Routage

5.5.1 Principes de routage

Quand un périphérique envoie un paquet à un autre périphérique, il consulte sa table de routage d'abord. Si la destination se trouve sur un réseau distant, le paquet est envoyé à la passerelle par défaut.

Les routeurs sont des périphériques de couche 3, qui ont donc aussi une table de routage. Ils transfèrent des paquets qui ne leur sont pas destinés, pour les amener dans un autre réseau.

5.5.2 Passerelle par défaut

La passerelle par défaut peut amener le trafic vers d'autres réseaux. Le trafic ne peut pas sortir du réseau local s'il n'y a pas de passerelle par défaut, si son adresse n'est pas configurée ou si elle est en panne.

En général la table de routage d'un hôte contient une passerelle par défaut.

Pour recevoir l'adresse IP de la passerelle par défaut :

- En IPv4, soit manuellement, soit dynamiquement par DHCP.
- En IPv6, soit manuellement, soit le routeur annonce l'adresse de la passerelle par défaut.

5.5.3 Table de routage

La table de routage contient toutes les adresses réseau (préfixes) connues et où transférer les paquets. Ces entrées s'appellent des *routes*.

Sur un hôte

Pour afficher la table de routage sur un hôte :

- Sur Windows :

```
route print
ou
netstat -r
```

- Sur Linux :

```
ip route
```

Dans la première colonne on peut trouver la route, c'est-à-dire l'adresse réseau de destination.

Dans la deuxième colonne on peut trouver la sortie, c'est-à-dire vers où transférer le paquet. Cette sortie s'appelle `next hop` ou `prochain saut`. Il peut s'agir d'une adresse IP d'un autre périphérique ou d'un identifiant d'interface duquel sortira le paquet.

L'adresse réseau de la route par défaut se note `0.0.0.0/0`, `::/0` ou `default`.

Sur un routeur

La table de routage d'un routeur contient trois types de routes :

- **Réseaux directement connectés** — Ce sont des interfaces de routeur actives. N'oublions pas que chaque interface d'un routeur est connecté à un réseau différent. Le routeur ajoute une route directement connectée quand une interface est configurée avec une adresse IP et est activée (`up/up`).

— **Réseaux distants** — Les réseaux distants ne sont pas directement connectés au routeur. Par défaut le routeur ne connaît pas les réseaux distants. Ils sont soit explicitement configurés par un administrateur (cf. 5.5.6), soit en échangeant des informations de routage avec d'autres routeurs en utilisant un protocole de routage (cf. 5.5.7).
— **Route par défaut** — Elle constitue l'alternative, quand aucune meilleure route n'a été trouvée. Comme un routeur ne pourra pas connaître tous les réseaux du monde, cette route par défaut est très souvent présente dans les tables de routage. Elle peut être entrée manuellement en tant que route statique, ou bien apprise automatiquement grâce à un protocole de routage dynamique. En IPv4 la route par défaut se note `0.0.0.0/0`. En IPv6 elle se note `::/0`. La longueur de préfixe `/0` indique qu'aucun bit n'a besoin de correspondre. On l'appelle parfois la *passerelle de dernier recours*.

Pour afficher la table de routage sur un routeur :

```
R1# show ip route
R1# show ipv6 route
```

La première partie de la table de routage contient des codes lettrés qui indiquent de quelle façon chaque route a été découverte par le routeur :

— **C** — Connected
Un réseau directement connecté. Souvent les entrées ayant **C** comme source seront suivies d'une entrée ayant **L** comme source.
Une route `C` est ajoutée automatiquement quand une interface reçoit une adresse IP et est activée.
— **L** — Local
L'adresse IP de l'interface connectée à un réseau directement connecté (visible avec **C**). Pour les routes locales IPv4 la longueur de préfixe est `/32`. Pour les routes locales IPv6 la longueur de préfixe est `/128`. Cela veut dire que pour que la route soit sélectionnée, l'entièreté de l'adresse IP du paquet doit correspondre à cette route locale. L'intérêt des routes locales est de distinguer les paquets qui lui sont destinés des paquets qui doivent être transférés.
Une route `L` est ajoutée automatiquement quand une interface reçoit une adresse IP et est activée.
— **S** — Static
Une route statique, c'est-à-dire configurée à la main. Il s'agit en général d'une route par défaut. Cette route par défaut garantit que le routeur ne rejettera aucun paquet. Voir 5.5.6
— **O** — Protocole OSPF
— **D** — Protocole EIGRP
— ***** — La route candidate pour une route par défaut

On peut aussi utiliser une route pour diriger le trafic vers un périphérique spécifique. On appelle cela une *route hôte*.

Les routes hôtes ressemblent aux routes locales : elles ont un masque à `/32` ou une longueur de préfixe à `/128`. La différence avec les routes locales est que l'adresse IP de destination de la route hôte ne sera pas celle d'une interface du routeur mais celle du périphérique où envoyer les paquets. Le masque `/32` ou la longueur de préfixe `/128` s'assureront que la

route sera utilisée seulement pour les paquets à destination du périphérique en question.

Sous les codes lettrés ou peut trouver les routes.

— **En IPv4** — Certaines lignes sont indentées, à cause du format historique de recherche par classes IPv4. Même si les routes ne sont plus recherchées avec les classes, le format n'a pas changé.
Une entrée indentée s'appelle une *route enfant*. Il s'agit du sous-réseau d'une adresse par classe (A, B ou C) (voir 5.2.2).
Les réseaux directement connectés (`C`) seront toujours indentés (routes enfant) parce que l'adresse locale (`L`) a toujours une longueur de préfixe de `/32`.
La route enfant aura un code lettré de source de route ainsi toutes les informations de routage (prochain saut, etc.). Le réseau par classe de ce sous-réseau se trouve juste au dessus de l'entrée de route, moins indentée et sans code lettré de source. Il s'agit de la *route parent*.

```
Router# show ip route
(Output omitted)
     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
O    192.168.2.0/24 [110/65] via 192.168.12.2, 00:32:33, Serial0/0/0
O    192.168.3.0/24 [110/65] via 192.168.13.2, 00:31:48, Serial0/0/1
     192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.12.0/30 is directly connected, Serial0/0/0
L       192.168.12.1/32 is directly connected, Serial0/0/0
     192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.13.0/30 is directly connected, Serial0/0/1
L       192.168.13.1/32 is directly connected, Serial0/0/1
     192.168.23.0/30 is subnetted, 1 subnets
O       192.168.23.0/30 [110/128] via 192.168.12.2, 00:31:38, Serial0/0/0
Router#
```

— **En IPv6** — Puisque l'IPv6 n'a jamais eu de concept de classes, la structure d'une table de routage IPv6 est beaucoup plus évidente. Toutes les entrées sont formatées de la même façon.

```
R1# show ipv6 route
(output omitted for brevity)
OE2 ::/0 [110/1], tag 2
     via FE80::2:C, Serial0/0/1
C   2001:DB8:ACAD:1::/64 [0/0]
     via GigabitEthernet0/0/0, directly connected
L   2001:DB8:ACAD:1::1/128 [0/0]
     via GigabitEthernet0/0/0, receive
C   2001:DB8:ACAD:2::/64 [0/0]
     via GigabitEthernet0/0/1, directly connected
L   2001:DB8:ACAD:2::1/128 [0/0]
     via GigabitEthernet0/0/1, receive
C   2001:DB8:ACAD:3::/64 [0/0]
     via Serial0/1/1, directly connected
L   2001:DB8:ACAD:3::1/128 [0/0]
     via Serial0/1/1, receive
O   2001:DB8:ACAD:4::/64 [110/50]
     via FE80::2:C, Serial0/1/1
O   2001:DB8:ACAD:5::/64 [110/50]
     via FE80::2:C, Serial0/1/1
L   FF00::/8 [0/0]
     via Null0, receive
R1#
```

Distance administrative

Il ne peut y avoir qu'une seule route par adresse réseau dans la table de routage. Cependant, il est possible qu'un routeur découvre la même adresse réseau par plusieurs sources.

À priori, un seul protocole de routage dynamique devrait être configuré sur un routeur. Il est pourtant possible de configurer à la fois OSPF et EIGRP sur le même routeur, et ces deux protocoles peuvent découvrir le même réseau. Chaque protocole peut décider d'un chemin différent pour rejoindre la destination, en se basant sur la valeur `metric` de ce protocole.

Mais du coup, comment le routeur fait-il pour savoir quel protocole utiliser ? Quelle route devrait être ajoutée dans la table de routage ?

Pour déterminer quelle route fera partie de la table de routage, l'IOS utilise la distance administrative (Administrative Distance (AD)). Cette AD représente l'indice de confiance de la route. Plus elle est basse, plus la route est préférée.

EIGRP a une AD de 90 et OSPF une AD de 110. Mais cette valeur n'indique pas nécessairement quel protocole est meilleur.

Ce qui se produit plus souvent est lorsqu'un routeur découvre la même adresse de réseau par une route statique et par un protocole de routage dynamique. Une route statique a une AD de 1. La route statique sera donc mise dans la table de routage.

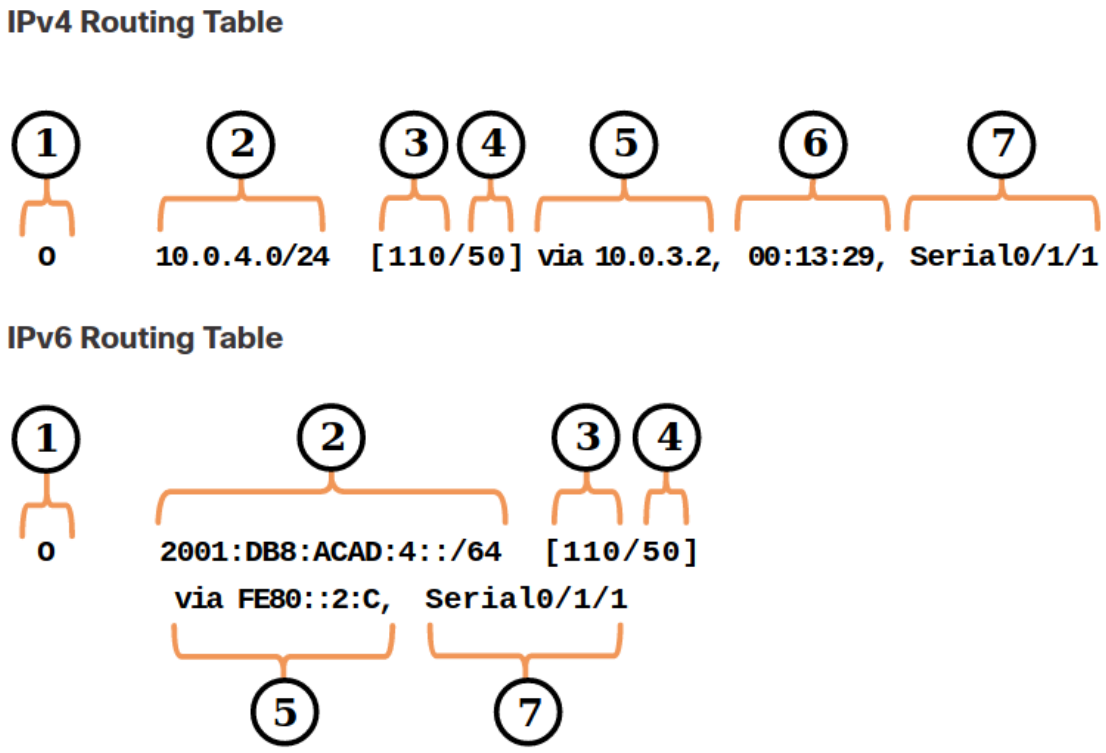
Il existe une AD encore plus basse, c'est-à-dire une AD de 0. Elle est réservée au réseaux directement connectés.

Les trois principes d'une table de routage

1. *Chaque routeur prend ses décisions seul : il se base sur les informations disponibles dans sa propre table de routage.* — Un routeur ne peut pas transférer de paquet autrement qu'avec ses propres informations. Il ne connaît pas les routes présentes dans les tables de routage d'autres routeurs.
2. *Les informations contenues dans la table de routage d'un routeur ne correspondent pas forcément à la table de routage d'un autre routeur* — Si par exemple R1 a une route pour un réseau qui indique qu'il faut transférer à R2, cela ne veut pas dire que R2 connaît ce réseau.
3. *La route pour un chemin ne constitue pas une route pour le chemin inverse* — Si R1 reçoit un paquet pour PC1 de la part de PC3 et sait transférer ce paquet par la bonne interface, cela ne veut pas dire qu'il saura transférer un paquet de PC1 à destination de PC3.

Être sûr de vraiment savoir lire une table de routage

Voici un découpage des informations contenues dans une table de routage d'un routeur, en IPv4 et en IPv6 :



1. **Source de la route** — Comment a été apprise la route.
2. **Réseau de destination (préfixe et longueur de préfixe)** — L'adresse du réseau distant.
3. **Distance administrative** — Indice de confiance de la source de la route. Plus la valeur est basse, plus la route sera préférée.
4. **Métrique** — Une valeur assignée pour atteindre le réseau distant. Là encore, une valeur plus basse en fera une route préférée.
5. **Prochain saut** — L'adresse IP du prochain routeur vers lequel le paquet doit être transféré.
6. **Timestamp de la route** — Combien de temps s'est écoulé depuis que la route a été découverte.
7. **Interface de sortie** — Interface egress pour envoyer un paquet.

La longueur de préfixe désigne le nombre minimum de bits à gauche devant correspondre à l'adresse IP du paquet.

5.5.4 Déterminer le meilleur chemin

On parle souvent de la *correspondance la plus longue* pour parler du meilleur chemin choisi par un routeur dans la table de routage. Il s'agit du processus que le routeur utilise pour trouver la correspondance entre l'adresse IP de destination et une route de la table.

La table de routage contient des entrées qui consistent en une adresse réseau (ou un préfixe) et un masque (ou une longueur de préfixe). Pour que la correspondance soit faite,

il faut qu'il y ait un nombre minimum de bits à gauche qui correspondent entre l'adresse IP du paquet et la route dans la table de routage. Le masque (ou la longueur de préfixe) est utilisé pour déterminer ce nombre minimum de bits. Le paquet ne contient pas le masque (ou longueur de préfixe), seulement l'adresse IP.

La correspondance la plus longue est la route qui a le plus grand nombre de bits correspondant à l'adresse IP de destination. Celle-ci sera toujours la route préférée.

Voici un exemple en IPv4 puis en IPv6 :

Destination IPv4 Address		Address in Binary
172.16.0.10		**10101100.00010000.00000000.00**001010
Route Entry	**Prefix/Prefix Length**	**Address in Binary**
1	172.16.0.0**/12**	**10101100.0001**0000.00000000.00001010
2	172.16.0.0**/18**	**10101100.00010000.00**000000.00001010
3	172.16.0.0**/26**	**10101100.00010000.00000000.00**001010

For the destination IPv6 packet with the address **2001:db8:c000**::99, consider the following three route entries:

Route Entry	Prefix/Prefix Length	Does it match?
1	**2001:db8:c0**00::**/40**	Match of 40 bits
2	**2001:db8:c000**::**/48**	Match of 48 bits (longest match)
3	**2001:db8:c000**:5555::**/64**	Does not match 64 bits

5.5.5 Transfert de paquet

Une fois que le routeur a déterminé où transférer son paquet grâce à sa table de routage, il doit l'encapsuler dans une trame ethernet.

Pour cela il vérifie s'il connaît l'adresse MAC correspondant à l'adresse IP de destination (ou du prochain saut pour un réseau distant) :

— En IPv4, il regarde dans sa table ARP. S'il ne trouve pas la correspondance il envoie une requête ARP.
— En IPv6, il regarde dans son cache de voisins. S'il ne trouve pas la correspondance il envoie une sollicitation NS.

À noter que si le paquet doit sortir par un port série, celui-ci ne possède pas d'adresse MAC : la liaison se fait de point à point (PPP). Quand le routeur encapsulera le paquet dans une nouvelle trame, il utilisera une adresse MAC de broadcast en destination.

Les routeurs peuvent utiliser un des trois mécanismes de transfert de paquet suivants :

1. **Commutation de processus** — Mécanisme de transfert de paquet plus ancien mais toujours présent sur les routeurs Cisco. Quand un paquet arrive sur une interface, il est transféré au domaine de contrôle où le Central Processing Unit (CPU) fait la correspondance de l'adresse de destination et la table de routage, puis détermine l'interface egress et transfère le paquet. Le routeur applique ces

étapes pour chaque paquet, même si la destination est identique pour un flux de paquets. La *commutation de processus* est un mécanisme très lent et elle est rarement implémentée dans des réseaux modernes.

2. **Commutation rapide** — Un autre mécanisme de transfert de paquet ancien, qui était le successeur de la *commutation de processus*. La *commutation rapide* utilise un cache pour enregistrer l'information du saut suivant. Quand un paquet arrive sur une interface, il est d'abord transféré au cache de *commutation rapide* pour vérifier si une correspondance est connue. Si rien n'est trouvé, le routeur effectue alors une *commutation de processus* comme précédemment pour transférer le paquet. Le cache contient aussi les informations de flux. Si un autre paquet avec la même destination arrive sur une interface, l'information du prochain saut dans le cache est réutilisée sans faire intervenir le CPU.
3. **Cisco Express Forwarding (CEF)** — C'est le mécanisme de transfert de paquet le plus récent et celui par défaut dans les IOS de Cisco. De la même manière que la *commutation rapide*, CEF construit une table Forwarding Information Base (FIB) et une table de contiguïté. Cependant, contrairement à la *commutation rapide* où le processus de transfert dépendait des paquets, le processus de transfert démarre par rapport aux changements de la topologie réseau. Ainsi, une fois qu'un réseau est en place, la table FIB et la table de contiguïté contiennent toutes les informations qu'un routeur pourrait avoir besoin pour transférer un paquet. Ce mécanisme est le plus rapide des trois.

5.5.6 Routage statique

Principe

En routage statique, c'est l'administrateur réseau qui configure le routage. Sur une topologie complexe, c'est d'avantage sujet à erreurs. S'il y a un changement dans la topologie, la route statique correspondante devra être mise à jour manuellement.

Mais les intérêts sont d'avantage de sécurité (pas de divulgation des routes sur le réseau) et une meilleure efficacité des ressources. Les routes statiques utilisent également moins de bande passante et de processus CPU que les protocoles de routage dynamique.

On peut tout à fait (et c'est commun) combiner une route statique avec un protocole de routage dynamique. Un protocole dynamique pourra prendre en compte les routes statiques ajoutées à la main.

Les routes statiques sont en général utilisées pour les scénarios suivants :

— En route par défaut vers un FAI.
— Pour des routes en dehors du domaine de routage et qui ne sont pas découvertes par le protocole de routage dynamique.
— Quand l'administrateur réseau souhait explicitement définir le chemin pour un réseau spécifique.
— Pour le routage entre réseaux d'extrémité.

Types de routes statiques

Différents types de routes statiques, à la fois pour IPv4 et IPv6 :

— **standard**
On ne met pas de route vers des réseaux directement connectés. C'est inutile, le routeur les connaît déjà.
— **par défaut**
C'est une route statique qui matche avec n'importe quel paquet. Ayant un masque à 0, aucun bit de l'adresse de destination n'aura besoin de correspondre. Elle sera donc sélectionnée si aucune meilleure route n'a été trouvée.
En général, elle part sur l'interface du routeur qui est connecté à internet. Elle représente donc tout réseau qui n'est pas dans la table de routage.
— **récapitulative** — sur-réseau
— **flottante** — chemin de secours
Pour la redondance : si un lien devient indisponible, cette route sera utilisée à la place. Dans ce cas la route statique aura une AD plus élevée que celle d'un protocole dynamique ou d'une route statique définie comme principale. La route flottante ne sera seulement ajoutée à la table de routage si la route principale a un problème.
Par défaut, les routes statiques ont une AD de 1, ce qui en fera des routes préférées par rapport à une route dynamique. Pour en faire une route flottante, il suffit de lui donner une valeur de distance plus élevée que la valeur AD d'un protocole dynamique :
— EIGRP = 90
— OSPF = 110
— IS-IS = 115
Tant que le lien principal est disponible, la route flottante n'apparaîtra pas dans la table de routage. Pour vérifier que la route flottante a été configurée, il faut afficher la `running-config`.

Prochain saut (next hop)

Quand on ajoute une route statique, on doit entre autres spécifier le prochain saut (ou tronçon suivant). Celui-ci peut être identifié par :

— **route de prochain saut** — une adresse IP seule.
L'adresse peut être distante : dans ce cas cela créera une route statique récursive. Le routeur fera des calculs additionnels pour déterminer l'interface de sortie.
C'est la méthode généralement recommandée.
— **route statique connectée directement** — une interface de sortie seule.
Devrait seulement être utilisée dans une configuration de point à point (connection `Serial`).
— **route statique entièrement spécifiée** — une interface de sortie et une adresse IP.
Nécessaire quand l'interface de sortie est une interface à accès multiple. Les liens ethernet sont à accès multiple : contrairement au liens série de point à point, il peut y avoir plus d'un périphérique sur un réseau à accès multiple. Il faut alors identifier le prochain saut de manière explicite.
Le prochain saut doit être directement connecté à l'interface de sortie.
En IPv6, si la route statique utilise une adresse *link-local* (voir 5.3.5) comme prochain saut, celle-ci ne sera pas incluse dans la table de routage. Ceci est dû au fait que les adresses *link-local* sont uniques seulement au sein d'un lien ou d'un

réseau. Il faut donc utiliser une *route statique entièrement spécifiée* en ajoutant l'interface de sortie.

Quand l'interface de sortie va sur un réseau ethernet, il est recommandé d'inclure dans la route une adresse IP de prochain saut. Cela se fait soit par une **route de prochain saut**, soit par une **route entièrement spécifiée**.

Pour les commandes Cisco, voir 8.3.3.

5.5.7 Routage dynamique (principes)

En routage dynamique, c'est le protocole qui gère, mais le routage dynamique est moins sécurisé que le routage statique.

Ces protocoles permettent à un routeur de découvrir automatiquement les réseaux distants grâce à d'autres routeurs. En effet un réseau distant pour un routeur est toujours un réseau connecté pour un autre routeur. Les routeurs qui utilisent des protocoles de routage dynamique partagent des informations de routage avec les autres routeurs. Cela prend en compte les changements dans la topologie et ne nécessite donc pas d'intervention de la part d'un administrateur.

Les protocoles de routage dynamique incluent OSPF et Enhanced Interior Gateway Routing Protocol (EIGRP).

L'administrateur n'a besoin d'ajouter que les réseaux directement connectés. Ensuite le protocole de routage dynamique fera les choses suivantes :

- découvrir des réseaux distants
- mettre à jour les informations de routage
- choisir le meilleur chemin vers des destinations réseau
- essayer de déterminer le nouveau meilleur chemin si le chemin actuel n'est plus accessible

Il est commun sur certains routeurs d'utiliser à la fois des routes statiques et un protocole de routage dynamique.

Les protocoles de routage dynamique sont en général utilisés pour les scénarios suivants :

- Dans des réseaux avec plus que quelques routeurs.
- Pour prévoir des changements dans la topologie.
- Pour l'évolutivité.

Les composantes principales d'un protocole de routage dynamique sont :

- **Les structures de données** — Les protocoles de routage utilisent souvent des tables ou des bases de données. Ces informations sont gardées dans la RAM.
- **Les messages de protocole de routage** — Types de messages variés pour découvrir les routeurs voisins, échanger des informations de routage, et d'autres tâches pour maintenir des informations correctes sur le réseau.
- **L'algorithme** — Liste finie d'étapes pour accomplir une tâche. Les algorithmes sont utilisés entre autres pour trouver le meilleur chemin.

Protocoles de routage dynamique

Quelques protocoles de routage dynamique et leur AD :

Source de la route	AD
Réseau directement connecté	0
Route statique	1
Route sommaire EIGRP	5
BGP externe	20
EIGRP interne	90
OSPF	110
IS-IS	115
RIP	120
EIGRP externe	170
BGP interne	200

Même si le protocole Routing Information Protocol (RIP) a été mis à jours vers RIPv2 pour faire face à la croissance des environnements réseaux, il ne permet toujours pas l'évolutivité des réseaux modernes.

Pour accomoder les besoins de réseaux plus larges, deux protocoles avancés de routage ont été mis en place : OSPF et Intermediate System to Intermediate System (IS-IS). Cisco a développé Interior Gateway Routing Protocol (IGRP), qui a plus tard été remplacé par EIGRP.

En plus de cela, il y avait le besoin de connecter différents domaines de routage de différentes organisations et de les router. Border Gateway Protocol (BGP), le successeur de Exterior Gateway Protocol (EGP) est utilisé entre FAI. BGP est aussi utilisé entre des FAI et certaines organisations privées pour échanger des informations de routage.

Les protocoles de la famille Interior Gateway Protocol (IGP) servent à échanger des informations de routage au sein d'un domaine de routage administré par une seule organisation.

Il n'y a qu'un seul protocole de la famille EGP et c'est BGP. BGP sert à échanger des informations de routage entre organisations (qu'on appelle des systèmes autonomes Autonomous System (AS)). BGP est utilisé par les FAI pour le routage des paquets sur internet.

Les termes de vecteur de distance (*distance vector*), d'état de liaison (*link-state*) et de vecteur de chemin (*path vector*) font référence au type d'algorithme de routage utilisé pour déterminer le meilleur chemin.

	Interior Gateway Protocols				Exterior Gateway Protocols
	Distance Vector		Link-State		Path Vector
IPv4	RIPv2	EIGRP	OSPFv2	IS-IS	BGP-4
IPv6	RIPng	EIGRP for IPv6	OSPFv3	IS-IS for IPv6	BGP-MP

Estimation du meilleur chemin

Avant d'ajouter une route à la table de routage, le protocole de routage dynamique va devoir évaluer le meilleur chemin vers ce réseau. Il va falloir par exemple comparer plusieurs chemins vers la même destination pour choisir le plus rapide ou le plus optimal. Si plusieurs chemins existent, chaque chemin utilisera une interface de sortie différente.

Pour choisir le meilleur chemin, le protocole se base sur la valeur du `metric`. Cette valeur est une mesure de la distance qui sépare le routeur du réseau de destination. Le meilleur chemin est celui qui aura la valeur de `metric` la plus basse.

C'est le protocole de routage qui génère une valeur de `metric`. Elle peut se baser sur une ou plusieurs charactéristiques du chemin. Par exemple la bande passante, le nombre de sauts, etc.

Protocole de routage	Métrique
RIP	= nombre de sauts (maximum 15).
OSPF	= « coût », basé sur la bande passante cumulative. Les liens plus rapides ont un coût plus bas.
EIGRP	Calculé par rapport à la bande passante la plus lente et du délai. Peut aussi inclure la charge et la fiabilité.

Équilibrage de la charge

Une table de routage peut contenir plusieurs chemins avec des métriques identiques vers le même réseau de destination. Dans ce cas, le routeur transmet les paquets en utilisant les deux chemins de manière égale. La table de routage contiendra alors pour le même réseau de destination plusieurs interfaces de sortie.

L'équilibrage de charge peut améliorer l'efficacité et les performances du réseau. Il est implémenté automatiquement par les protocoles de routage dynamique.

Pour le configurer avec des routes statiques, il faut plusieurs routes statiques vers le même réseau de destination avec différentes valeurs de `prochain saut`.

5.5.8 OSPF

En IPv4 on utilise OSPFv2. En IPv6 on utilise OSPFv3.

OSPF a été développé comme une alternative à RIP. RIP ne dépendait que du nombre de sauts pour calculer les chemins, ce qui est insuffisant dans les réseaux de plus grande taille, où les liens peuvent avoir des débits différents.

OSPF est un protocole d'état de lien qui utilise le concept de *zones*. On peut découper le domaine de routage en zones distinctes pour mieux contrôler le trafic de mises à jour des routes.

Un lien est une interface sur un routeur, ou un segment entre deux routeurs. Un peut aussi être un segment vers un réseau d'extrémité (comme un LAN ethernet connecté à

un seul routeur). Les informations sur l'état d'un lien incluent le préfixe de réseau, la longueur de préfixe et le coût.

— **Messages du protocole de routage** — Les routeurs qui utilisent OSPF s'échangent des messages en utilisant 5 types de paquets :

1. Hello (cf. 1)
2. Description de la base de données (cf. 2)
3. Requête d'état de lien (cf. 3)
4. Mise à jour d'état de lien (cf. 4)
5. Acquittement d'état de lien (cf. 5)

— **Structures de données** — Les messages OSPF sont utilisés pour créer et maintenir 3 bases de données OSPF.
Ces tables contiennent une liste de routeurs voisins pour échanger les informations de routage. Elles sont gardées dans la Random-Access Memory (RAM).

1. Base de données adjacente — crée la table des voisins. La table des voisins est unique pour chaque routeur.

```
R1# show ip ospf neighbor
```

2. Base de données d'état de lien (Link-State DataBase (LSDB)) — crée la table de topologie. La table de topologie liste les informations sur tous les autres routeurs du réseau. Tous les routeurs d'une zone ont le même LSDB.

```
R1# show ip ospf database
```

3. Base de données de transfert — crée la table de routage. La table de routage de chaque routeur est unique.

```
R1# show ip route
```

— **Algorithme** — Le routeur construit la table de topologie en utilisant l'algorithme de Dijkstra (Shortest Path First (SPF)). Cet algorithme se base sur le coût cumulatif pour rejoindre une destination.
L'algorithme SPF construit d'abord un arbre SPF. L'arbre est ensuite utilisé pour calculer les meilleures routes. Ces meilleures routes sont placées dans la base de données de transfert, qui sert à construire la table de routage.

Opération d'état de lien

Construire la table de routage n'est pas suffisant. Il faut aussi la maintenir à jour. Les routeurs suivent un processus d'état de lien pour atteindre un état de convergence.

Chaque lien entre deux routeurs se voit attribuer un coût. Un routeur suit les étapes suivantes pour l'état de lien :

1. *Établir les voisins adjacents*
   Les routeurs utilisant OSPF doivent se reconnaître sur le réseau avant de pouvoir échanger des informations. Ils envoient des paquets `Hello` pour déterminer si un routeur utilisant OSPF est présent sur ce lien.

2. *Échanger des annonces d'état de lien*
   Une fois qu'un contact est établi, les routeurs peuvent échanger des annonces d'état de lien (Link-State Advertisement (LSA)). Ces annonces contiennent l'état et le coût de chaque lien connecté. Chaque routeur envoient ses LSA à ses voisins directs, qui transfère les LSA à leur voisins, jusqu'à ce que tous les routeurs de la zone sont en possession de tous les LSA.
3. *Construire la base de données d'état de lien*
   Quand tous les LSA sont reçus, les routeurs peuvent construire leur table de topologie en fonction des LSA reçus. Au bout d'un moment, cette base de données contient toutes les informations sur la topologie de la zone.
4. *Exécuter l'algorithme SPF*
   Ensuite les routeurs exécutent l'algorithme SPF, ce qui crée l'arbre SPF.
5. *Choisir la meilleure route*
   Les meilleurs chemins, établis à l'étape précédente, constituent des routes prêtes pour la table de routage. Ces routes sont ajoutées à la table de routage si aucune autre route avec une AD plus basse n'y est présente.

OSPF à zone unique et à zones multiples

OSPF permet le routage hiérarchique à l'aide de zones. Une zone OSPF est un groupe de routeurs qui partagent les mêmes informations d'état de lien dans leurs LSDB.

On peut implémenter OSPF de deux façons :

1. **OSPF à zone unique** — Tous les routeurs sont dans une zone. Il est conseillé d'utiliser la zone 0.
2. **OSPF multizone** — OSPF est hiérarchique, en utilisant plusieurs zones. Toutes les zones doivent être connectées au cœur (zone 0). Les routeurs interconnectant les zones sont appelés Area Border Router (ABR).

Les calculs SPF sont gourmands en CPU et se font à tout changement de la topologie. Ils font créer une nouvelle table SPF et mettre à jour la table de routage. Le temps que met l'algorithme à effectuer ces calculs dépend de la taille de la zone.

C'est pourquoi une zone peut être découpée en zones plus petites pour faire de l'OSPF multizone. Le routage aura toujours lieu entre les zones, mais les opérations intensives pour le processeur seront cantonnées à une zone. Les routeurs d'autres zones reçoivent toujours les changements de topologie, mais ces routeurs ne font que mettre à jour leur table de routage. Ils ne relancent pas de calcul SPF.

Le découpage hiérarchique en multizones permet :

— **Des tables de routage plus petites** — Il y a moins d'entrées dans chaque table. On peut résumer des routes entre zones (cela n'est pas activé par défaut).
— **Une réduction de la charge des mises à jour d'états de lien** — Des zones plus petites signifient moins de besoins de mémoire et de charge de processeur.
— **Une réduction de la fréquence des calculs SPF** — L'impact d'un changement de topologie est localisé dans une zone. L'inondation des LSA s'arrête à une frontière de zone.

OSPFv3

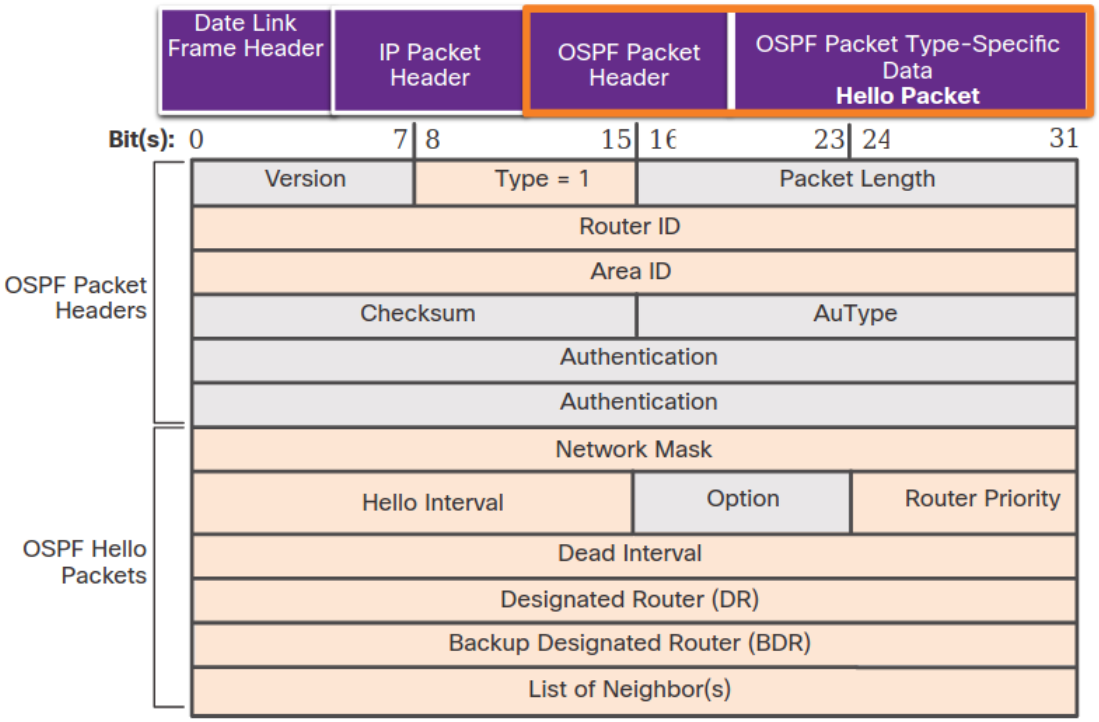
OSPFv3 est l'équivalent à OSPFv2 pour échanger des préfixes IPv6. OSPFv3 utilise aussi l'algorithme SPF pour déterminer les meilleurs chemins.

Les processus et les opérations sont à peu près les mêmes que pour OSPFv2, mais tournent indépendamment. Les tables sont séparées (voisins, topologie, et routage). Les commandes de configuration pour les deux versions sont similaires.

Types de paquets OSPF

Les Link-State Packet (LSP) sont échangés pour établir les voisins et faire les mises à jour de routage. Chaque paquet a un but spécifique :

1. **Hello**
   — Utilisé pour établir et maintenir les contiguïtés d'autres routeurs OSPF.
   — Transmet des paramètres pour établir quels routeurs sont voisins.
   — Élit le DR et le Backup Designated Router (BDR) sur des réseaux à accès multiples comme ethernet. Les liaisons de point à point ne nécéssitent pas de DR ou de BDR.



   Les champs importants :
   — **Type** — Identifie le type de paquet. 1 pour Hello, 2 pour DBD, 3 pour LSR, 4 pour LSU et 5 pour LSAck.
   — **Router ID** — Une valeur de 32 bits exprimée en notation décimale pointée comme une adresse IPv4. Identifie le routeur émetteur de manière unique. L'ID de routeur peut être définie par l'administrateur réseau ou assignée automatiquement par le routeur.
   Pour définir le Router ID, le routeur se base sur trois critères, par ordre préférentiel :

(a) Explicitement configuré (méthode recommandée).

(b) Adresse IPv4 la plus élevée parmi les adresses loopback configurées.

(c) Adresse IPv4 la plus élevée parmi les adresses de ses interfaces physiques. Une interface physique n'a pas besoin de faire du OSPF pour être choisie comme ID de routeur.

Pour les configurations du Router ID, voir 5.5.8.

— **Area ID** — Numéro de la zone d'où vient le paquet.
— **Network mask** — Masque de sous-réseau associé avec l'interface émettrice.
— **Hello interval** — La fréquence, en secondes, à laquelle un routeur envoie des paquets Hello. Par défaut, la valeur est de 10 secondes. Il faut qu'elle soit identique sur le routeur voisin, sinon la contiguïté n'est pas établie.
— **Router priority** — Utilisé dans le processus d'élection des DR/BDR. La priorité par défaut est de 1 mais peut être modifiée à la main entre 0 et 255. Le DR sera le routeur avec la plus grande priorité. Le BDR sera le routeur avec la deuxième plus grande priorité.
— **Dead interval** — Le temps, en secondes, que le routeur attend pour avoir des nouvelles de son voisin avant de le déclarer inactif. Par défaut, cette valeur est de 4 fois la valeur du `Router priority`. De même, il faut qu'elle soit identique sur le routeur voisin, sinon la contiguïté n'est pas établie.
— **DR** — Il s'agit du `Router ID` du DR.
— **BDR** — Il s'agit du `Router ID` du BDR.
— **List of Neighbors** — Une liste qui identifie les `Router ID` de tous les routeurs adjacents.

2. **Description de la base de données** (DataBase Description (DBD))
   Contient une liste abrégée de la LSDB du routeur émetteur. Les routeurs qui le reçoivent le comparent à leur propre base de données. Toutes les LSDB de la zone doivent être identiques.
3. **Requête d'état de lien** (Link-State Request (LSR))
   Les routeurs ayant reçu un paquet DBD peuvent alors demander plus d'informations en envoyant un paquet LSR.
4. **Mise à jour d'état de lien** (Link-State Update (LSU))
   Utilisé pour répondre aux LSR et annoncer de nouvelles informations. Un LSU peut contenir 11 différents types de LSA pour prévenir de changements de routage.
5. **Acquittement d'état de lien** (Link-State Acknowledgement (LSAck))
   Quand un LSU est reçu, le routeur envoie un LSAck pour confirmer la réception du LSU. Le champs de données du LSAck est vide.

États d'opération OSPF

Quand un routeur est connecté à un réseau, il passe par différents états :

— État **Down** — Pas de paquets Hello reçus. Le routeur envoie des paquets Hello à l'adresse multicast 224.0.0.5. Quand un routeur reçoit un paquet Hello avec un ID de routeur qui n'est pas dans sa liste de voisins, il tente d'établir une contiguïté avec le routeur émetteur du paquet. Passe à l'état *Init*.
— État **Init** — Les paquets Hello sont reçus par le voisin. Ils contiennent l'ID du routeur émetteur. Passe à l'état *Two-Way*.

— État **Two-Way** — La communication entre les deux routeurs est maintenant bidirectionnelle. Sur les liens à accès multiples (ethernet), les routeurs élisent un DR et un BDR. Passe à l'état *ExStart*.
— État **ExStart** — Sur des réseaux de point à point, les deux routeurs décident lequel des deux va initier l'envoi des paquets DBD. Ils se mettent aussi d'accord sur le numéro initial de séquence de paquet DBD. Le routeur avec l'ID le plus élevé sera le premier routeur à envoyer des paquets DBD.
— État **Exchange** — Les routeurs échangent des paquets DBD. Si des informations supplémentaires sont nécessaires, passe à l'état *Loading*. Sinon, passe directement à l'état *Full*.
— État **Loading** — Des informations additionnelles sont acquises avec les LSR et LSU. Les routes sont calculées avec l'algorithme SPF. Passe à l'état *Full*.
— État **Full** — La base de données d'état de lien du routeur est entièrement synchronisée. Les routeurs envoient des LSU à leurs voisins à tout changement ou toutes les 30 minutes.

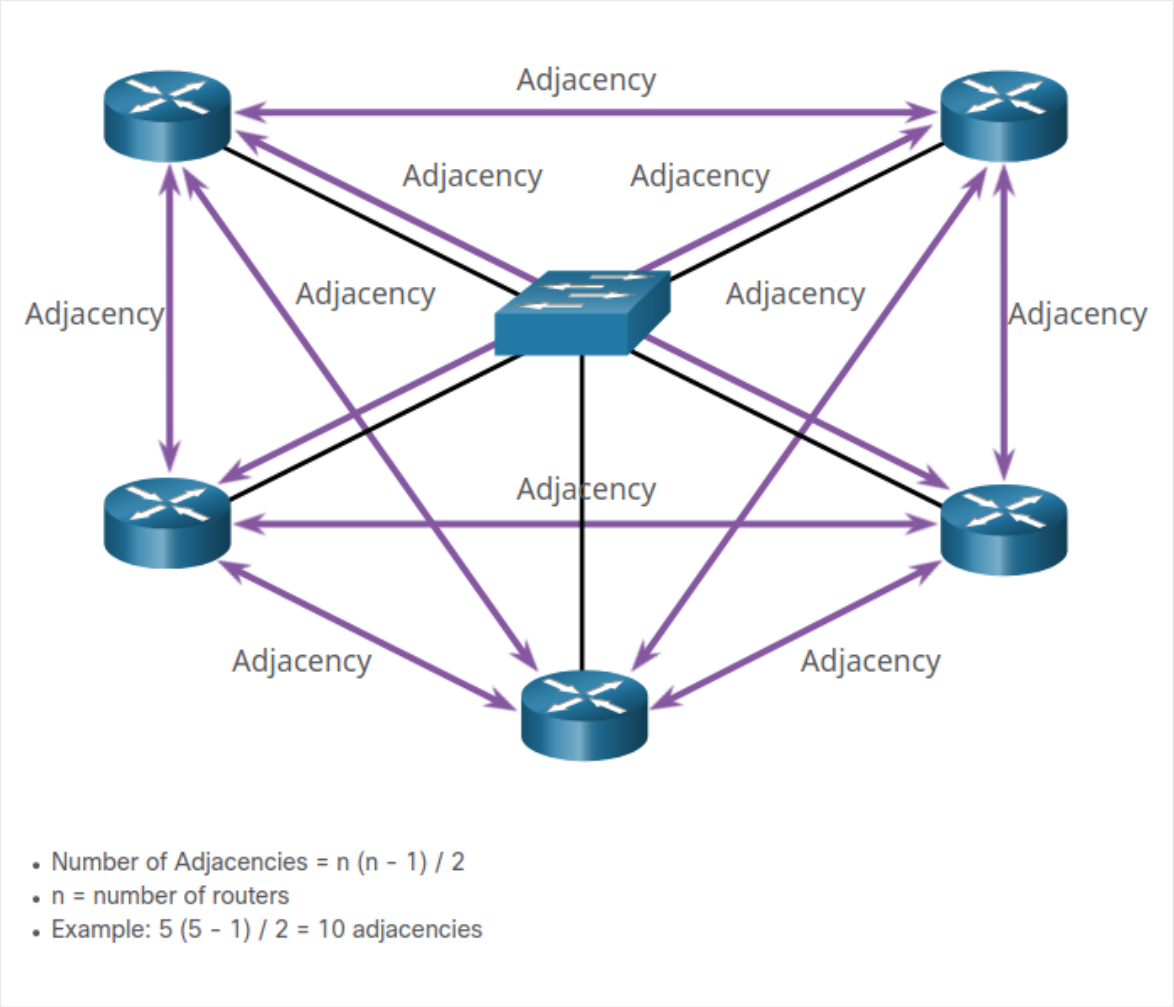
Pourquoi avoir un DR ?

On a parlé d'élection de DR et de BDR, mais à quoi ça sert ?

Les réseaux à accès multiples posent deux problèmes à OSPF :

— **Création de contiguïtés multiples** — Les réseaux ethernet peuvent potentiellement connecter beaucoup de routeurs sur un même lien. Cela crée une contiguïté entre tous les routeurs, ce qui n'est pas nécessaire ni désirable.
— **Trop d'inondations de LSA** — Les routeurs inondent le réseau de LSA à chaque fois qu'il s'initialisent ou qu'il y a un changement dans la topologie. Cette inondation peut devenir excessive.

Si n représente le nombre de routeurs, il y a $\frac{n(n-1)}{2}$ contiguïtés.



Cela ne semble pas beaucoup, mais selon cette formule, un réseau de 20 routeurs créerait 190 contiguïtés. Toutes ces contiguïtés créent une quantité de LSA envoyés beaucoup trop élevée.

La solution est le DR. Le DR est le point de collecte et de distribution des LSA. Le BDR est élu au cas où le DR tombe. Tous les autres routeurs deviennent des DROTHER.

Si on ajoute un routeur à la topologie après que l'élection a eu lieu et que le routeur a une priorité plus élevée ou un ID de routeur plus élevé que le DR, il ne devient pas DR. L'élection n'est pas refaite.

Les LSA sont donc envoyés au DR et c'est le DR qui transmet les LSA aux autres routeurs, qui lui répondent à lui. Les routeurs ne s'envoient donc pas mutuellement d'informations redondantes.

Le DR envoit les LSA à l'adresse multicast 224.0.0.5. Les DROTHER envoient les LSA à l'adresse multicast 224.0.0.6. Seuls le DR et le BDR écoutent sur l'adresse 224.0.0.6.

Configuration OSPF

D'abord, il faut activer OSPF :

```
R1(config)# router ospf <process-id>
```

La valeur du `process-id` est un nombre entre 1 et 65535, choisi par l'administrateur réseau. Il n'est pas obligatoire d'utiliser le même `process-id` sur tous les routeurs OSPF, mais c'est conseillé.

La commande ci-dessus nous fait entrer dans le mode de configuration de routeur, et le prompt devient `R1(config-router)#`.

Router ID

Pour ajouter manuellement un ID de routeur :

```
R1(config-router)# router-id 1.1.1.1
R1(config-router)# end
R1# show ip protocols | include Router ID
  Router ID 1.1.1.1
R1#
```

Pour configurer une interface loopback en tant qu'ID de routeur :

```
R1(config)# interface Loopback 1
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# end
R1# show ip protocols | include Router ID
  Router ID 1.1.1.1
R1#
```

On utilise le masque une longueur de préfixe de 32 (255.255.255.255) pour créer une route hôte. Les routes hôtes de 32 bits ne sont pas annoncées aux autres routeurs OSPF.

Une fois que le routeur a choisi un ID de routeur, il n'est pas possible de changer l'ID de routeur à moins de recharger le routeur, ou de redémarrer le processus OSPF. Cela est dû au fait que le routeur a déjà établi des contiguïtés avec d'autres routeurs avec l'ancien ID de routeur.

```
S1# show ip protocols
  Router ID 10.10.1.1
S1# clear ip ospf process
S1# show ip protocols
  Router ID 1.1.1.1
```

Configurer la priorité d'un routeur

On l'a vu, par défaut les routeurs ont une priorité de 1. Ce sera alors leur ID de routeur qui déterminera le DR et le BDR. Mais choisir le DR grâce à l'ID de routeur est vite limité dans des grands réseaux. Il vaut donc mieux configurer la priorité du routeur pour choisir le DR et le BDR.

De plus, on peut spécifier une priorité au niveau de l'interface, alors que l'ID de routeur s'applique au routeur entièrement. Utiliser la priorité permet donc qu'un routeur soit DR sur un réseau et DROTHER sur un autre.

```
R1(config-if)# ip ospf priority <value>
```

La valeur doit être comprise entre 0 et 255. Une valeur à 0 empêche l'interface d'agir en tant que DR.

Configurer le réseau OSPF

```
R1(config-router)# network <network-address> <wildcard-mask> area <area-id>
```

— `<network-address> <wildcard-mask>` — Active OSPF sur toutes les interfaces. Toute interface qui fait partie du réseau défini par `<network-address>` peut envoyer et recevoir des paquets OSPF.
La valeur `<wildcard-mask>` est l'inverse du masque de sous-réseau. On peut la calculer en soustrayant le masque de sous-réseau à 255.255.255.255.
Un masque de sous-réseau de 255.255.255.192 donnera :

$$255.255.255.255 - 255.255.255.192 = 0.0.0.63$$

Certaines versions d'IOS permettent de renseigner le masque de sous-réseau. Dans ce cas, l'IOS convertit le masque de sous-réseau en masque wildcard.

— `<area-id>` — Il s'agit de la zone OSPF. La convention veut qu'on utilise 0 comme zone dans le cas d'une zone unique. Cela permet d'ajouter d'autres zones plus tard si besoin.

Il est aussi possible de configurer OSPF directement dans l'interface physique plutôt que par la commande `network`.

```
R1(config)# interface g0/0/0
R1(config-if)# ip ospf 10 area 0
```

Interfaces passives

Par défaut, un routeur envoie ses messages OSPF par toutes les interfaces configurées en OSPF. Mais en réalité, un message ne devrait quitter une interface seulement si elle a un voisin OSPF. L'envoi de paquets OSPF inutiles vers un LAN pose 3 problèmes :

1. Utilisation inutile de la bande passante.
2. Utilisation inutile de ressources, les périphériques du LAN devant gérer et rejeter les paquets.
3. Risque de sécurité : les messages OSPF pouvant être interceptés et des informations modifiées et donc erronées renvoyées vers les routeurs.

Pour empêcher une interface d'envoyer des messages OSPF mais en permettant toujours le réseau d'être annoncé aux autres routeurs :

```
R1(config-router)# passive-interface <interace-id>
```

Pour mettre toutes les interfaces en passive :

```
R1(config-router)# passive-interface default
```

Et réactiver une interface seulement :

```
R1(config-router)# no passive-interface <interface-id>
```

Réseau de point à point

Par défaut, le routeur met le réseau en mode `BROACAST`, et il élit un DR et un BDR.

```
R1# show ip ospf interface <interface-id>
```

Si une interface est sur un lien de point à point, il faut le lui dire manuellement :

```
R1(config-if)# ip ospf network point-to-point
```

La commande `show ip ospf interface` affiche désormais le réseau en `POINT_TO_POINT`.

Réseau à accès multiples

Dans un réseau à accès multiples, un routeur est désigné pour distribuer les paquets LSA. Ce routeur devrait être choisi par l'administrateur réseau à travers une bonne configuration.

Quand plusieurs routeurs sont connectés au même switch, on parle de réseau à accès multiples.

On peut définir manuellement le DR et BDR quand on affecte un ID au routeur (5.5.8). Le routeur ayant l'ID le plus élevé deviendra DR, le routeur ayant le deuxième ID le plus élevé sera le BDR.

Afficher les contiguïtés

```
R1# show ip ospf neighbor
```

Les voisins peuvent avoir 4 états :

1. **FULL/DROTHER** — Le routeur est un DR ou un BDR et est adjacent à un DROTHER. Ces deux routeurs peuvent échanger des paquets Hello, LSU LSR et LSAck.
2. **FULL/DR** — Le routeur est adjacent à un DR. Ces deux routeurs peuvent échanger des paquets Hello, LSU LSR et LSAck.
3. **FULL/BDR** — Le routeur est adjacent à un BDR. Ces deux routeurs peuvent échanger des paquets Hello, LSU LSR et LSAck.
4. **2-WAY/DROTHER** — Un DROTHER adjacent à un autre DROTHER. Ces deux routeurs peuvent échanger des paquets Hello.

Configurer la bande passante de référence ou le coût à la main

La formule de calcul du coût est la suivante : 100000000 / bande passante de l'interface.

Interface Type	Reference Bandwidth in bps		Default Bandwidth in bps	Cost
10 Gigabit Ethernet 10 Gbps	100,000,000	÷	10,000,000,000	0.01 = 1
Gigabit Ethernet 1 Gbps	100,000,000	÷	1,000,000,000	0.1 = 1
Fast Ethernet 100 Mbps	100,000,000	÷	100,000,000	1
Ethernet 10 Mbps	100,000,000	÷	10,000,000	10

Same Costs due to reference bandwidth

Les interfaces Fast, Giga et 10 Giga auront donc un coût à 1.

Pour changer ça, on peut changer la bande passante de référence :

```
R1(config-router)# auto-cost reference-bandwidth <Mbps>
```

La valeur par défaut est 100. Pour un lien Gigabit, il faut donc mettre 1000. Pour un lien 10 Giga, il faut mettre 10000.

Cette commande est nécessaire sur tous les routeurs de la topologie.

Ou bien on peut définir manuellement le coût sur l'interface :

```
R1(config-if)# ip ospf cost <value>
```

Ceci a des conséquences, il faut s'assurer de bien mesurer les implications.

Intervalles des paquets Hello

Par défaut sur les liens à accès multiples broadcast et de point à point, les paquets Hello sont envoyés à l'adresse multicast 224.0.0.5 toutes les 10 secondes.

Les intervalles de `Hello` et de `Dead` sont configurables par interface. Les valeurs doivent être identiques pour les deux voisins.

Pour afficher les intervalles courantes :

```
R1# show ip ospf interface <interface-id>
```

Pour afficher le temps restant pour le dernier Dead Time :

```
R1# show ip ospf neighbor
```

Pour modifier les intervalles :

```
R1(config-if)# ip ospf hello-interval <seconds>
R1(config-if)# ip ospf dead-interval <seconds>
```

Propager la route par défaut

Le routeur Autonomous System Boundary Router (ASBR) agit comme passerelle pour que le trafic sorte du réseau. Il peut s'agir du lien vers internet, ou bien vers un réseau non OSPF.

Ce routeur n'a besoin que d'une route par défaut. Pour propager cette route par défaut dans le domaine OSPF, il faut 2 étapes :

— Ajouter une route statique par défaut :

```
R1(config)# ip route 0.0.0.0 0.0.0.0 <next hop>
```

— Faire propager la route :

```
R1(config)# router ospf 10
R1(config-router)# default-information originate
R1(config-router)# end
R1#
```

Vérification de la configuration

Vérifier que les interfaces sont actives et ont la bonne configuration IP :

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

Vérifier que la table de routage comprend les routes attendues :

```
R1# show ip route
```

D'autres commandes utiles :

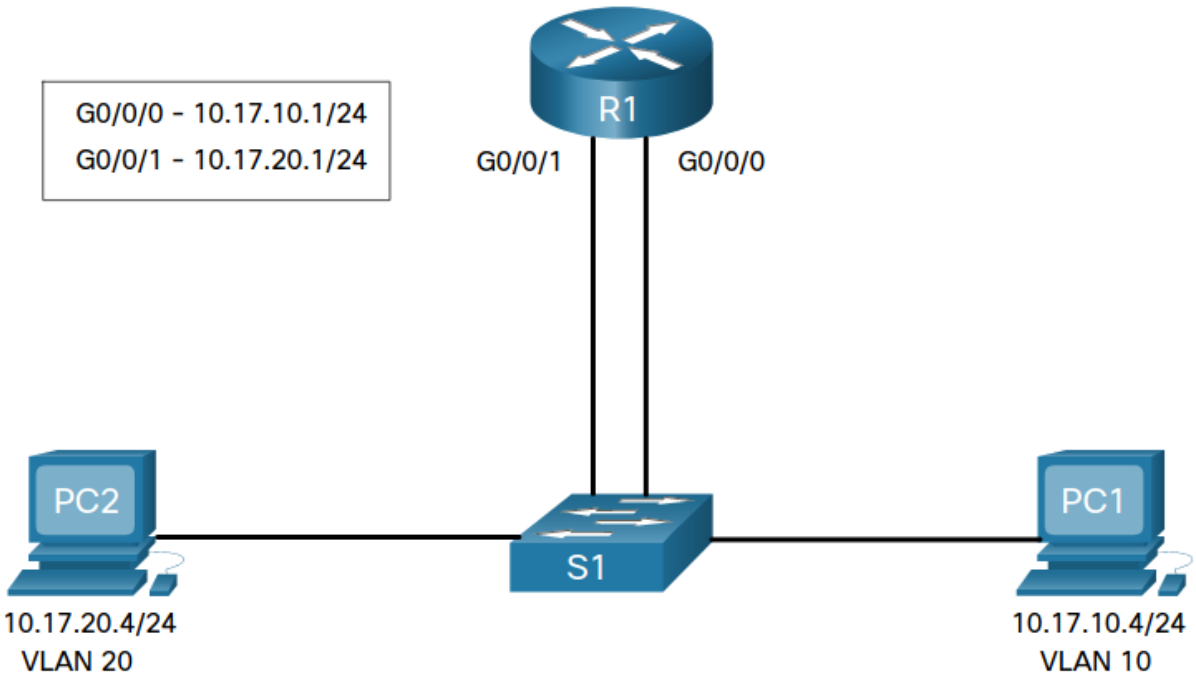
— `show ip ospf neighbor` — Affiche les informations suivantes :
  — **Neighbor ID** — ID de routeur du voisin.
  — **Pri** — Priorité de l'interface.
  — **State** — État OSPF de l'interface. Devrait être soit `FULL` soit `2-WAY` (entre routeurs DROTHER).
  — **Dead Time** — Temps restant avant de déclarer le voisin comme inactif. Remis à la valeur du Dead Time quand l'interface reçoit un paquet Hello.
  — **Address** — Adresse IPv4 de l'interface du voisin.
  — **Interface** — Interface à laquelle le voisin est connecté.
— `show ip protocols` — Inclut l'ID de processus OSPF, l'ID du routeur, les interface configurées explicitement pour annoncer des routes OSPF, les voisins, et la AD par défaut, qui est de 110 pour OSPF.
— `show ip ospf` — Afficher également les ID de processus OSPF et de routeur. Inclut aussi les informations de zone, et la dernière fois que l'algorithme SPF a été exécuté.
— `show ip ospf interface [<interface-id>]` — Affiche une liste détaillée pour chaque interface OSPF contenant l'ID de processus OSPF, l'ID de routeur, le type du réseau, le coût OSPF, le DR et le BDR, et les voisins adjacents.
On peut ajouter `brief` pour avoir un sommaire rapide.

5.5.9 Routage inter-VLAN

Normalement les hôtes d'un VLAN ne peuvent pas communiquer avec les hôtes d'un autre VLAN, sauf s'il existe un routeur ou un commutateur de couche 3. Il existe trois façons d'inter-connecter des VLAN :

1. **Routage inter-VLAN existant** — solution ancienne qui ne s'étend pas bien.
2. **Router-on-a-Stick** — solution acceptable pour un réseau de petite à moyenne taille.
3. **Commutateur de couche 3 utilisant des SVI** — solution la plus évolutive.

Routage inter-VLAN existant



Un routeur et un switch. Le routage se fait sur le routeur, tout ce qu'il y a de plus normal. Chaque interface du routeur est connectée à un port du switch dans différents VLAN. Les interfaces de routeur servent de passerelle par défaut pour les hôtes de chaque VLAN.

Les hôtes sont donc branchés sur le switch, et leurs interfaces sont dans les VLAN correspondant à chaque interface du routeur.

En quelque sorte, le switch sert à ajouter des interfaces au routeur : les VLAN permettent d'étendre ces interfaces. C'est comme si tous les hôtes d'un même VLAN étaient branchés à la même interface du routeur.

Lorsqu'un hôte d'un VLAN envoie un paquet à un hôte sur un autre VLAN, le paquet passe par le routeur en entrant par l'interface correspondant au premier VLAN et en sortant par l'interface correspondant au second VLAN.

Cette méthode de routage inter-VLAN fonctionne mais est très limitée. En effet, il faudra consacrer un port physique du routeur pour chaque VLAN configuré. On risque donc rapidement de dépasser la capacité du routeur.

Pas vraiment d'interêt d'avoir des réseaux virtuels si on utilise une interface physique à chaque fois...

Cette méthode n'est donc plus implémentée dans les réseaux commutés.

Router-on-a-Stick

Cette méthode n'a besoin que d'une seule interface ethernet physique. L'interface de routeur est alors configurée comme `trunk 802.1Q` et connectée a un port de trunk sur un switch de couche 2. Cette interface sur le routeur est configurée avec des sous-interfaces pour identifier chaque VLAN. Chaque sous-interface est une SVI qui est associée à une interface ethernet physique avec sa propre adresse IP.

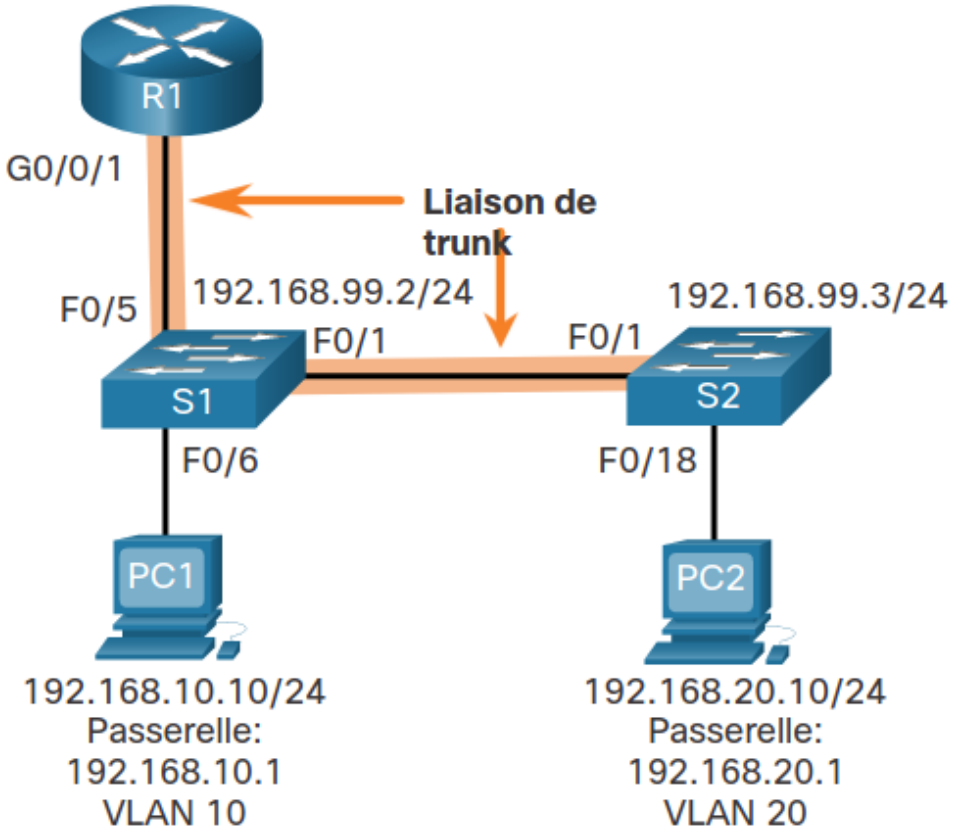
Quand le routeur reçoit un paquet étiqueté VLAN, il le transfère à la sous-interface VLAN. Ensuite il prend une décision de routage pour déterminer l'interface de sortie en fonction de l'adresse IP de destination. Si cette interface de sortie est configurée en tant que sous-interface `802.1Q`, la trame est marquée VLAN et renvoyée par l'interface physique.

Les deux sous-interfaces VLAN ont bien deux adresses IP différentes.

Avec cette méthode *router-on-a-stick* on ne peut pas dépasser 50 VLAN.

La méthode `router-on-a-stick` tient son nom du fait que le routeur n'est pas au centre de la topologie, mais apparaît sur le côté, comme sur un stick.

Le gros de la topologie consiste en liens entre le routeur et les switchs contenant les VLAN. Ces liens s'appellent des liens de trunk.



Pour que deux hôtes sur des VLAN différents puissent communiquer, il faut que les switchs soient configurés avec des VLAN et des trunks, et que le routeur soit configuré pour le routage inter-VLAN.

Avec la topologie ci-dessus, le routeur aura trois sous-interfaces :

Subinterface	VLAN	Adresse IP
G0/0/1.10	10	192.168.10.1 /24
G0/0/1.20	20	192.168.20.1 /24
G0/0/1.99	99	192.168.99.1 /24

Pour configurer les switchs avec des VLAN et des trunks :

1. **créer et nommer les VLAN** :

```
S1(config)# vlan 10
S1(config-vlan)# name LAN10
S1(config-vlan)# exit
S1(config)# vlan 20
S1(config-vlan)# name LAN20
S1(config-vlan)# exit
S1(config)# vlan 99
S1(config-vlan)# name Management
S1(config-vlan)# exit
S1(config)#
```

2. **créer l'interface de management** :
   Il faut sélectionner un VLAN et donner une passerelle par défaut vers le routeur.

```
S1(config)# interface vlan 99
S1(config-if)# ip address 192.168.99.2 255.255.255.0
S1(config-if)# no shutdown
S1(config-if)# exit
S1(config)# ip default-gateway 192.168.99.1
S1(config)#
```

3. **configurer les ports d'accès** :
   On sélectionne une interface connectée à un hôte pour l'assigner à un VLAN.

```
S1(config)# interface fa0/6
S1(config-if)# switchport mode access
S1(config-if)# switchport access vlan 10
S1(config-if)# no shutdown
S1(config-if)# exit
S1(config)#
```

4. **configurer les ports de trunk** :
   On sélectionne une interface connectée à un switch et une interface connectée à un routeur.

```
S1(config)# interface fa0/1
S1(config-if)# switchport mode trunk
S1(config-if)# no shutdown
S1(config-if)# exit
```

```
S1(config)# interface fa0/5
S1(config-if)# switchport mode trunk
S1(config-if)# no shutdown
S1(config-if)# end
S1#
```

L'autre switch de la topologie est configuré de la même manière.

Pour le routeur on va devoir créer une sous-interface pour chaque VLAN à router.

Pour cela on utilise la commande suivante :

```
R1(config)# interface <ID de l'interface>.<ID de la sous-interface>
```

Pour l'ID de sous-interface, il est de coutûme d'utiliser le numéro du VLAN correspondant. Ensuite on configure chaque sous-interface avec les deux commandes suivantes :

1. ```
   R1(config-subif)# encapsulation dot1q <ID du VLAN> [native]
   ```

   Cette commande configure le routeur pour qu'il réponde au trafic encapsulé `802.1Q` de l'ID de VLAN spécifié. L'option `native` n'est seulement utilisée pour définir le VLAN native sur autre chose que `VLAN1`.

2. ```
   R1(config-subif)# ip address <IP address> <subnet mask>
   ```

   Cette commande configure l'adresse IPv4 pour la sous-interface. Cette adresse est en général utilisée comme passerelle par défaut pour le VLAN identifié.

On refait les mêmes étapes pour chaque VLAN à router, puis il faut activer l'interface physique avec `no shutdown`.

Commutateur de couche 3 utilisant des SVI

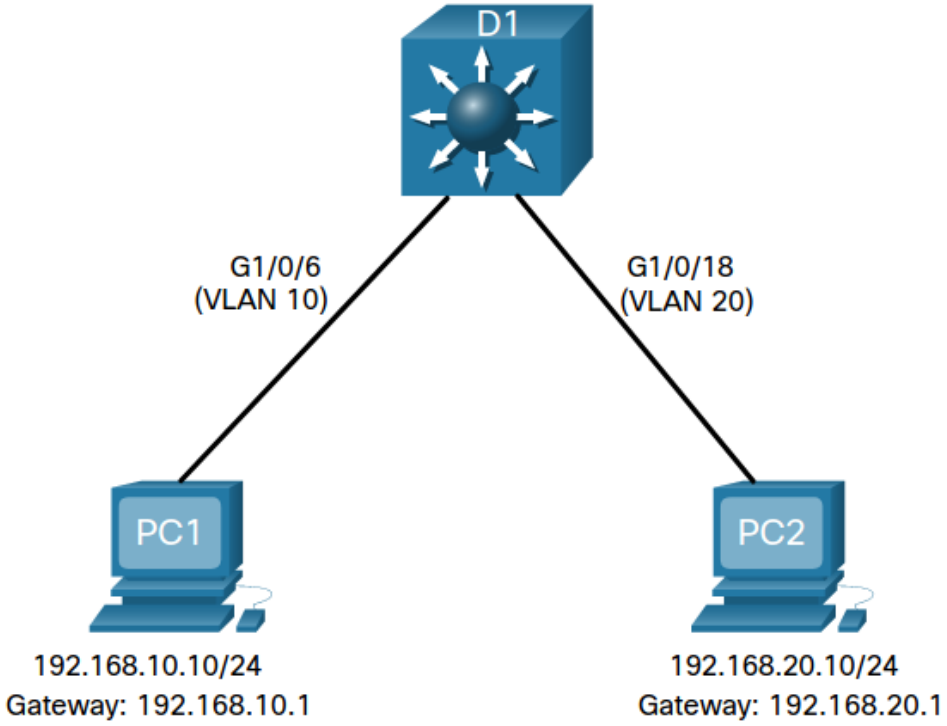
C'est la méthode moderne de routage inter-VLAN. Elle utilise des switchs de couche 3 et des SVI.

Une interface SVI est créée pour chaque VLAN du switch. Le SVI exécute les mêmes fonctions qu'une interface de routeur. Il fait un traitement de couche 3 des paquets vers tous les ports associés à ce VLAN.

Les avantages de cette méthode sont les suivants :

- beaucoup plus rapide que le `router-on-a-stick` car la commutation et le routage sont assurés de manière matérielle
- pas besoin de liaisons externe entre le switch et le routage
- les canaux `EtherChannel` peuvent être utilisés comme liaisons de trunk entre switchs pour augmenter la bande passante
- moins de latence puisque les données n'ont pas besoin de quitter le switch pour changer de réseau
- plus souvent déployés dans un réseau local de campus que les routeurs

Le seul inconvénient est que les switchs de couche 3 sont plus chers.



Chaque VLAN aura son adresse IP :

Interface VLAN	**Adresse IP**
10	192.168.10.1 /24
20	192.168.20.1 /24

Pour configurer le switch de couche 3 :

1. **créer les VLAN** :

```
D1(config)# vlan 10
D1(config-vlan)# name LAN10
D1(config-vlan)# vlan 20
D1(config-vlan)# name LAN20
D1(config-vlan)# exit
D1(config)#
```

2. **créer les interfaces SVI de chaque VLAN** :

   Les adresses IP vont servir de passerelle par défaut pour les hôtes dans chaque VLAN.

```
D1(config)# interface vlan 10
D1(config-if)# description Default Gateway SVI for 192.168.10.0 /24
D1(config-if)# ip address 192.168.10.1 255.255.255.0
D1(config-if)# no shutdown
D1(config-if)# exit
D1(config)# interface vlan 20
D1(config-if)# description Default Gateway SVI for 192.168.20.0 /24
D1(config-if)# ip address 192.168.20.1 255.255.255.0
D1(config-if)# no shutdown
```

```
D1(config-if)# exit
D1(config)#
```

3. **configurer les ports d'accès** :
   On choisit chaque port connecté aux hôtes pour leur assigner un VLAN.

```
D1(config)# interface g1/0/6
D1(config-if)# description Acces port to PC1
D1(config-if)# switchport mode access
D1(config-if)# switchport access vlan 10
D1(config-if)# exit
D1(config)# interface g1/0/18
D1(config-if)# description Acces port to PC2
D1(config-if)# switchport mode access
D1(config-if)# switchport access vlan 20
D1(config-if)# exit
D1(config)#
```

4. **activer le routage IP** :

```
D1(config)# ip routing
```

Pour que les VLAN soient joignables par d'autres périphériques de couche 3, il faut que le switch de couche 3 ait le routage activé. Pour cela il faut qu'un port routé soit configuré. On désactive la fonction de commutation sur un port (qui par défaut est de couche 2) connecté à un équipement de couche 3. Ainsi, la commande suivante sur l'interface qu'on souhaite configurer va la convertir en interface de couche 3 :

```
D1(config-if)# no switchport
```

Ensuite, on peut lui ajouter une adresse IP pour se connecter à un routeur ou un autre commutateur de couche 3.

Pour configurer le routage sur un switch de couche 3 en utilisant le protocole OSPF :

1. **configurer le port routé** :

```
D1(config)# interface g1/0/1
D1(config-if)# description routed Port Link to R1
D1(config-if)# no switchport
D1(config-if)# ip address 10.10.10.2 255.255.255.0
D1(config-if)# no shutdown
D1(config-if)# exit
D1(config)#
```

2. **activer le routage** :

```
D1(config)# ip routing
```

3. **configurer le routage** :

```
D1(config)# router ospf 10
D1(config-router)# network 192.168.10.0 0.0.0.255 area 0
D1(config-router)# network 192.168.20.0 0.0.0.255 area 0
D1(config-router)# network 10.10.10.0 0.0.0.3 area 0
D1(config-router)# end
```

4. **vérifier le routage** :

```
D1# show ip route | begin Gateway
Gateway of last resort is not set
     10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C       10.10.10.0/30 is directly connected, GigabitEthernet1/0/1
L       10.10.10.2/32 is directly connected, GigabitEthernet1/0/1
O       10.10.20.0/24 [110/2] via 10.10.10.1, 00:00:06,
    GigabitEthernet1/0/1
     192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.10.0/24 is directly connected, Vlan10
L       192.168.10.1/32 is directly connected, Vlan10
     192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.20.0/24 is directly connected, Vlan20
L       192.168.20.1/32 is directly connected, Vlan20
```

Résolution de problèmes de routage inter-VLAN

Les problèmes courants sont parmi les suivants :

- **VLAN manquants** :
  - créer ou re-créer les VLAN inexistants
  - s'assurer que le port hôte est assigné au bon VLAN

  Commandes pour vérifier :

```
D1# show vlan [brief]
D1# show interfaces switchport
D1# ping
```

- **Problèmes de port trunk sur les switchs** :
  - s'assurer que les trunks sont configurés correctement
  - s'assurer que le port est un port de trunk et qu'il est activé

  Commandes pour vérifier :

```
D1# show interfaces trunk
D1# show running-config
```

- **Problèmes de port d'accès sur les switchs** :
  - assigner le bon VLAN au port d'accès
  - s'assurer que le port est un port d'accès et qu'il est activé
  - l'hôte peut être mal configuré dans le mauvais sous-réseau

  Commandes pour vérifier :

```
D1# show interfaces swithport
D1# running-config interface
ipconfig
```

- **Problèmes de configuration de routeur** :
  - la sous-interface du routeur peut être configurée avec la mauvaise adresse IP
  - la sous-interface du routeur peut être assignée au mauvais ID de VLAN

  Commandes pour vérifier :

```
D1# show ip interface brief
D1# show interfaces
```

Chapitre 6

Couche Transport

6.1 Rôle de la couche Transport

1. **Suivre les conversations individuelles** :
   On entend par conversation tout ensemble de données qui voyage entre une application source et une application destinatrice. Chaque conversation est suivie séparément par la couche Transport (couche 4 du modèle OSI).
2. **Segmenter les données et réassembler les segments** :
   Les réseaux ne permettent en général pas d'envoyer toute une conversation en un bloc. C'est la couche Transport qui se charge de fragmenter les données applicatives. En fonction du protocole de couche Transport utilisé (TCP ou UDP), les blocs de couche Transport s'appellent des segments ou des datagrammes.
3. **Ajouter des informations d'en-tête** :
   Bien-sûr, la couche Transport ajoute aussi son en-tête à la PDU de couche supérieure (c'est-à-dire les données). Ces informations d'en-tête servent par exemple à reconstituer les blocs de données reçues.
4. **Identifier les applications** :
   La couche Transport doit suivre des données venant de plusieurs applications simultanément. Pour transmettre les flux de données à la bonne application, la couche Transport utilise des identifiants appelés des *ports*. Chaque application ayant besoin d'accéder au réseau se voit assignée un *port* unique.
5. **Multiplexer les conversations** :
   Si on envoyait un flux de données (comme une vidéo) en un flux continu sur le réseau, cela consommerait toute la bande passante. Cela empêcherait d'autres communications de se faire en même temps et rendrait compliqué la retransmission de données manquantes.
   En envoyant les données en blocs indépendants, la couche Transport peut suivre les données simultanément.

6.2 Protocoles de couche 4

La couche Transport propose deux protocoles complémentaires qui couvrent donc deux besoins différents.

1. TCP — mode connecté, transmission fiable mais plus lente.
2. UDP — mode non connecté, transmission sans suivi, plus rapide mais moins fiable.

Ainsi chaque application va nécessiter d'utiliser soit TCP, soit UDP.

Par exemple, certaines applications comme la VoIP ou les vidéo-conférences peuvent tolérer une perte de données mais étant des applications en temps réel elles ne pourront absolument pas accepter des délais de transmission. Dans ce cas, UDP sera plus approprié.

D'autres applications de requête/réponse avec peu de données où la retransmission peut être faite rapidement utilisent aussi UDP. Dans ce cas c'est à la couche Application que revient la responsabilité du suivi des données échangées. Ainsi, le Domain Name System (DNS) ou le SNMP utilisent UDP.

Pour d'autres utilisations au contraire, il est important que toutes les données arrive de manière fiable et qu'elles soient traitées dans le bon ordre. Dans ces cas TCP sera utilisé. Les bases de données, les navigateurs web, les clients mail en sont de bons exemples.

La transmission de vidéo pré-enregistrée n'est pas en temps réel. TCP sera souvent utilisé, ce qui permet la mise en mémoire tampon et le contrôle de congestion.

6.2.1 TCP

Présentation

TCP divise les données en segments. Il effectue un suivi de bout en bout de chaque bloc.

Pour cela il effectue les choses suivantes :

— Numéroter et suivre les segments de données transmis à un hôte spécifique à partir d'une application spécifique.
— Acquitter les données reçues.
— Retransmettre toute donnée non acquittée après un certain temps.
— Séquencer les données qui pourraient arriver dans le mauvais ordre.
— Envoyer les données à une vitesse efficiente mais acceptable pour le receveur.

Pour tout cela, TCP doit d'abord établir une connexion entre la source et la destination. Ainsi, on dit que TCP est un protocole orienté connexion.

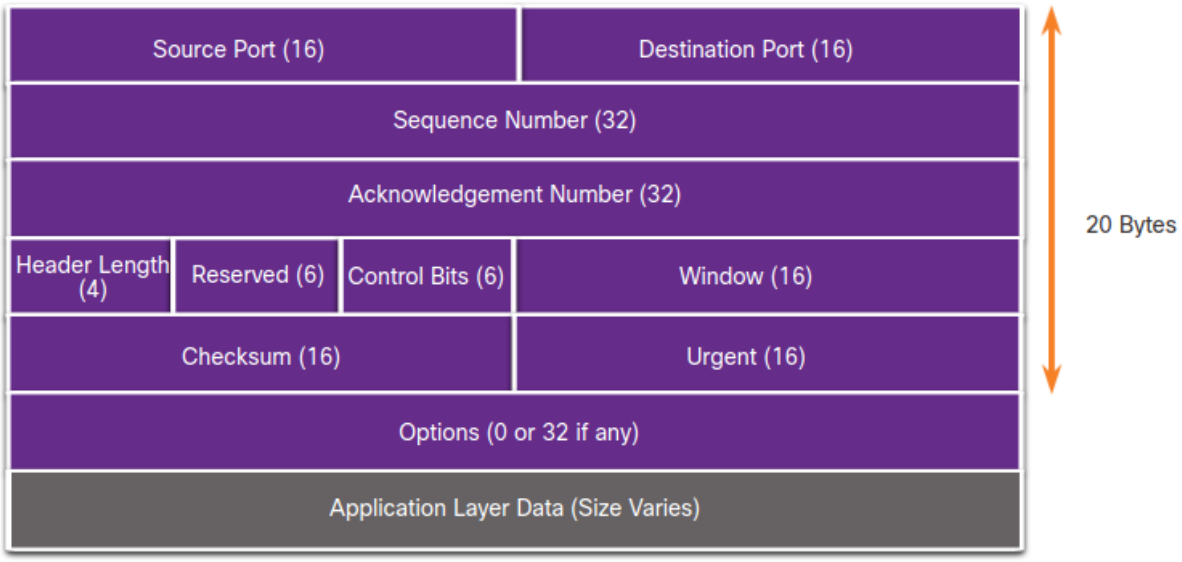
En s'occupant de la fragmentation des données en segments, en offrant la fiabilité, en contrôlant le flux de données, et en réassemblant les segments, TCP permet aux protocoles applicatifs de ne pas se soucier de tout ça. Ces protocoles applicatifs incluent :

— FTP
— HTTP
— Simple Mail Transfer Protocol (SMTP)
— SSH

En-tête TCP

TCP est un protocole stateful, ce qui veut dire qu'il garde le suivi de l'état d'une session de communication. Pour effectuer ce suivi, TCP enregistre ce qu'il a envoyé et quelles informations ont été acquittées (confirmées comme reçues).

L'en-tête TCP a une taille de 20 octets.



— **Source Port** (port source) — 16 bits
Identifie l'application source par son numéro de port.
— **Destination Port** (port de destination) — 16 bits
Identifie l'application de destination par son numéro de port.
— **Sequence Number** (numéro de séquence) — 32 bits
Utilisé pour réassembler les données.
— **Acknowledgment Number** (numéro d'acquittement) — 32 bits
Indique si les données ont été reçues.
— **Header Length** (longueur d'en-tête) — 4 bits
On l'appelle aussi *data offset*. Indique la longueur de l'en-tête du segment TCP.
— **Reserved** (réservé) — 6 bits
Réservé pour un usage ultérieur.
— **Control bits** (bits de contrôle) — 6 bits
Inclut du code binaire, ou des *flags*, pour indiquer la fonction et l'utilité du segment TCP.
Il y a 6 flags pour le champs de contrôle :

1. **URG** — urgent.
2. **ACK** — flag d'acquittement utilisé pour établir une connexion ou terminer une session.
3. **PSH** — fonction push.
4. **RST** — réinitialiser la connexion quand il y a une erreur ou plus de réponse.
5. **SYN** — synchroniser les numéros de séquence et utilisé pour établir une connexion.
6. **FIN** — plus de données de la part de l'émetteur et utilisé pour mettre fin à une session.

— **Window size** (taille de fenêtre) — 16 bits
Indique le nombre d'octets qui peuvent être acceptés à la fois.
— **Checksum** (checksum) — 16 bits
Pour la vérification d'erreurs.

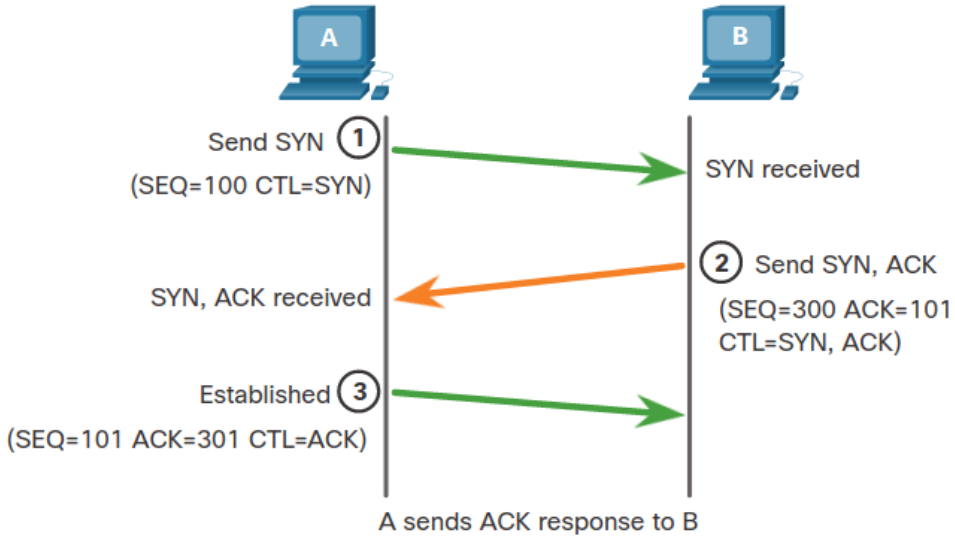
— **Urgent** (urgent) — 16 bits
Indique si les données sont urgentes.

Établissement d'une connexion

L'établissement d'une connexion TCP s'appelle un *three-way handshake*. C'est une poignée de main en 3 étapes :

1. `SYN`
2. `SYN` + `ACK`
3. `ACK`

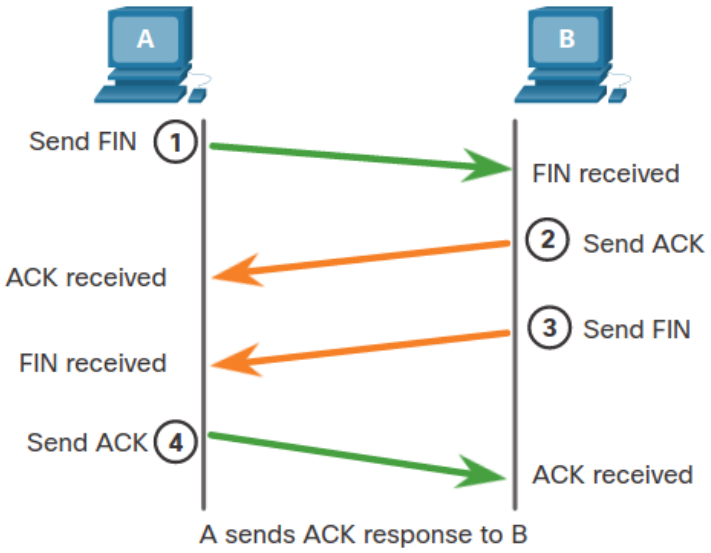
Les flags `SYN` et `ACK` veulent respectivement dire `Synchronization` (synchronisation) et `Acknowledgment` (acquittement).



Ce procédé d'établissement d'une connexion permet de s'assurer que le destinataire est disponible.

Fin d'une session

Pour clore une session, le flag `FIN` pour `Finish` (Fin) est utilisé. Cette fois ce ne sera pas un *three-way handshake*. Chaque côté de la connexion enverra son `FIN` puis son `ACK` pour confirmer la réception du `FIN`.



Segmentation

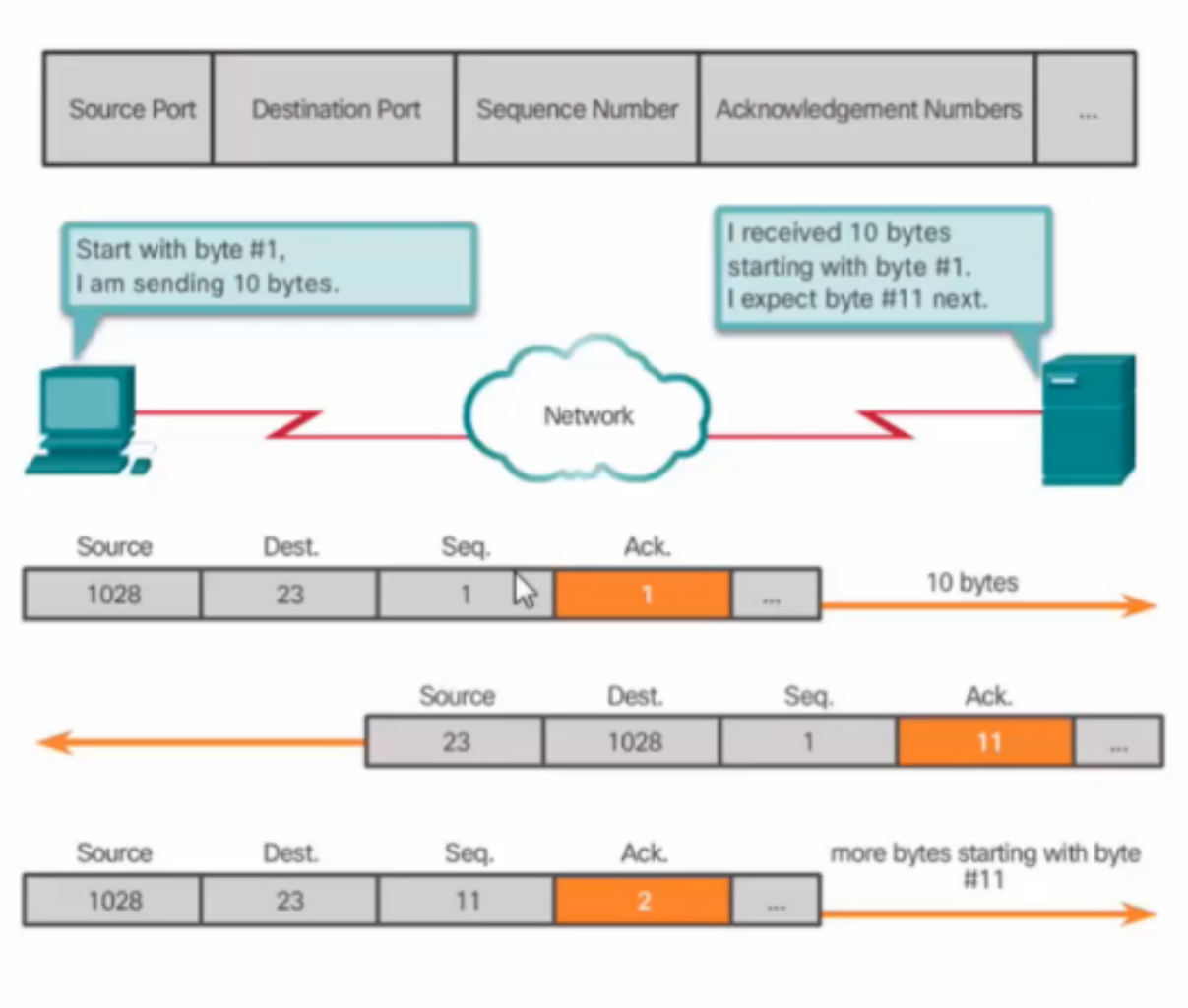
Parfois les segments n'arrivent pas à destination ou arrivent dans le mauvais ordre. Pour que le message original soit compréhensible il faut que toutes les données soient reçues et les segments doivent être réassemblés.

Pour cela, chaque segment reçoit dans son en-tête un numéro de séquence, un numéro d'acquittement et une taille de fenêtre.

1. *Numéro de séquence* (`SEQ`)
   Le premier numéro de séquence pourrait en toute logique être 0. Mais cela crée des problèmes de sécurité : quelqu'un peut intercepter une conversation et effectuer une attaque MitM en devinant le numéro de séquence.
   Pour éviter cela, le premier numéro de séquence est défini aléatoirement à l'établissement de la session initiale. Ce premier numéro de séquence s'appelle un Initial Sequence Number (ISN).
   Il est important de noter que l'ISN est dans la réalité un numéro aléatoire, mais que dans les cours en guise d'exemple et dans les traces Wireshark, l'ISN est parfois représenté par 1. Cela rend plus simple le suivi des traces Wireshark par exemple.
   À chaque envoi de données, le numéro de séquence est incrémenté du nombre d'octets transmis. Grâce à ce suivi d'octets de données, chaque segment peut être identifié de manière unique et acquitté.
   À la réception, le processus TCP place les données dans un buffer. Ces données sont envoyées à la couche Application une fois que les segments ont été réassemblés.
2. *Numéro d'acquittement* (`ACK`)
   L'hôte recevant un segment utilise ensuite un numéro d'acquittement pour informer l'hôte émetteur du nombre d'octets reçus. Le numéro d'acquittement correspond au numéro d'octet que le récepteur s'attend à recevoir ensuite. Il prend donc le numéro de séquence reçu et l'incrémente du nombre d'octets reçus.
3. *Taille de fenêtre* (voir 6.2.1)

La taille de fenêtre est un autre champ de l'en-tête TCP qui définit la quantité de données que peut émettre un hôte avant de recevoir un acquittement. Grâce à cette taille de fênetre, l'hôte émetteur peut adapter la vitesse à laquelle il envoie les données.

Voici un exemple simplifié d'un échange TCP :



Il y a un problème avec cette méthode. Si l'hôte A envoie 10 segments et que les segments 3 et 4 sont perdus, l'hôte B va répondre qu'il attend le segment 3. Mais l'hôte A ne sait pas si d'autres segments ont été perdus, donc il renvoie les segments 3 à 10. L'hôte B avait déjà reçu les segments 5 à 10 donc ceux-ci sont reçus deux fois. Cela congestionne et ralentit le réseau, des paquets non nécessaires étant renvoyés.

Pour contrer ce problème les systèmes aujourd'hui utilisent une fonctionnalité TCP optionnelle qui s'appelle le Selective Acknowledgement (SACK). Elle est négocié pendant le three-way handshake. Si les deux hôtes supportent le SACK, le récepteur peut acquitter explicitement quels octets ont été reçus.

En reprenant l'exemple précédent, l'hôte B aurait répondu avec un `ACK` 3 et un `SACK` de 5 à 10. L'hôte A n'aurait renvoyé que les segments 3 et 4.

Contrôle de flux

Le contrôle de flux désigne la quantité de données qu'un hôte destinataire peut recevoir de manière fiable. TCP a besoin d'ajuster la vitesse du flux de données entre source et destination pour maintenir la fiabilité d'une transmission.

Il le fait grâce à un champ d'en-tête : `Window size` (taille de fenêtre).

La taille de fenêtre détermine le nombre d'octets pouvant être envoyés avant de s'attendre à un acquittement. Ce champ est inclus dans l'en-tête de tous les segments TCP pour qu'un hôte puisse à tout moment modifier la taille de fenêtre en fonction de son cache.

La taille initiale est négociée pendant le three-way handshake. Une fois que le périphérique source (hôte A) a envoyé une quantité de données égale à la taille de la fenêtre, il ne peut pas envoyer d'avantage de données avant d'avoir reçu un acquittement. Cela dit, un hôte récepteur (hôte B) n'attend pas d'avoir reçu autant d'octets que sa taille de fenêtre avant d'envoyer des acquittements. Le périphérique A peut donc ajuster sa fenêtre d'émission (correspondant à la taille de fenêtre du périphérique B) au fur et à mesure qu'il reçoit des acquittements de la part de B. Le périphérique A va en fait incrémenter sa fenêtre d'émission de la valeur de l'acquittement reçu.

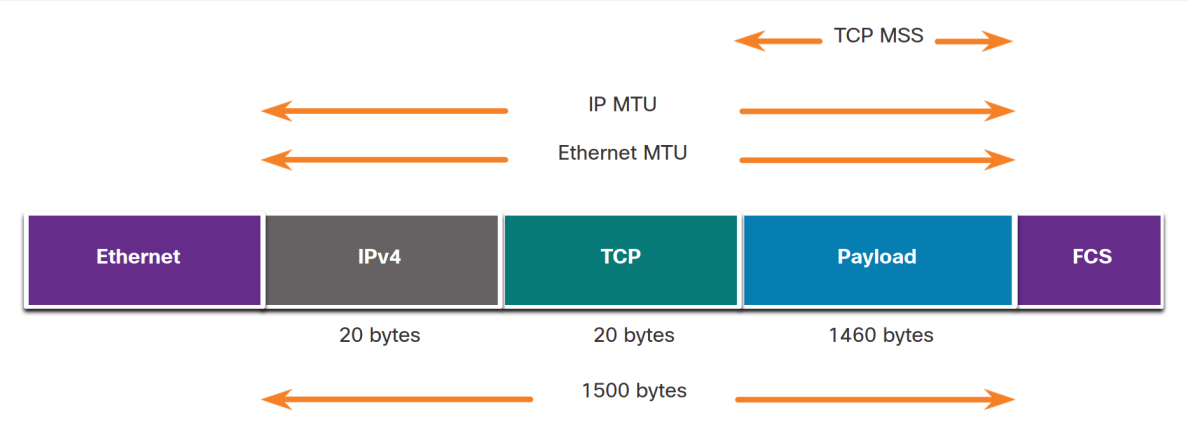
Si B envoie à A une taille de fenêtre de 10000 octets, la fenêtre d'envoi de A sera de 10000 octets. Si A reçoit un acquittement de la part de B correspondant à 2920, sa nouvelle fenêtre d'envoi sera de 12920. A peut donc envoyer jusqu'à 12920 octets de données sans recevoir d'acquittement.

Ce procédé permet au périphérique source d'émettre des segments de manière continue, tant que le périphérique destinataire acquitte les segments reçus.

Maximum Segment Size (MSS)

Le MSS est la taille maximale qu'un hôte peut recevoir pour un segment. Dans l'en-tête TCP, le MSS fait partie du champ `Options`. Cette taille est en octets et n'inclut pas l'en-tête TCP. Elle est incluse lors du three-way handshake.

En IPv4 il est commun que le MSS soit de 1460 octets. Un hôte détermine son MSS en retirant la taille des en-têtes IP et TCP du MTU. Sur une interface ethernet, le MTU par défaut est de 1500 octets. Les en-têtes IP et TCP faisant chacun 20 octets, on obtient bien un MSS de 1460 octets.



Évitement de la congestion

Une congestion entraîne une perte de paquets, rejetés par un routeur surchargé. Lorsqu'il y a congestion, il va y avoir retransmission de segments TCP perdus. Mais une retransmission sous entend de nouveaux paquets sur le réseau et de nouveaux acquittements de ces nouveaux paquets. Cela peut aggraver la congestion.

TCP a donc plusieurs mécanismes pour gérer la congestion. Si la source se rend compte qu'il ne reçoit pas les acquittements ou que ces acquittements n'arrivent pas de manière régulière, il peut réduire le nombre d'octets envoyés avoir de recevoir un acquittement. C'est bien la source qui adapte ces octets transmis, et non la *taille de fenêtre* du destinataire.

6.2.2 UDP

Présentation

UDP divise les données en datagrammes, mais ceux-ci peuvent aussi être appelés segments, comme pour TCP.

UDP est plus simple que TCP. Il ne fait pas de contrôle de flux. Il se contente d'envoyer les données. Cela veut dire qu'UDP est plus rapide que TCP.

Contrairement à TCP, UDP est un protocole sans connexion. Il n'a pas besoin d'établir de connexion parce qu'il ne suit pas les informations envoyées sur le réseau.

- Les données sont reconstituées dans l'ordre dans lequel elles sont reçues.
- Les segments perdus ne sont pas renvoyés.
- Il n'y a pas d'établissement d'une session.
- À l'envoi il n'y a pas de connaissance de la disponibilité des ressources.

UDP est un protocole stateless : ni la source ni la destination ne suit l'état d'une session de communication. S'il y a un besoin de fiabilité en utilisant UDP, ceci doit être géré au niveau applicatif.

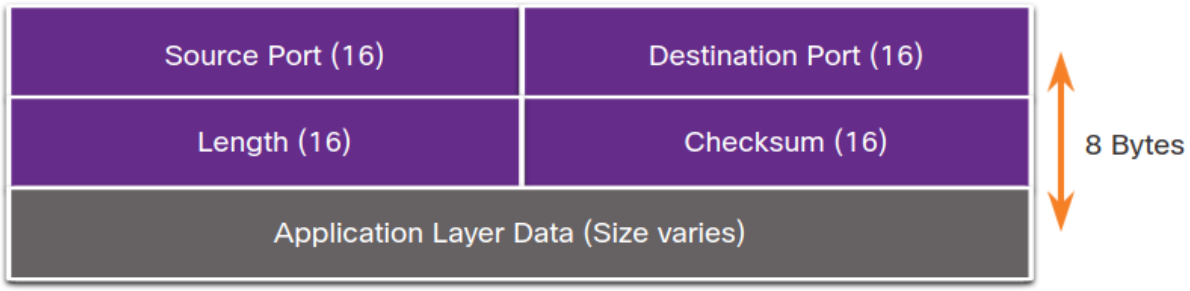
Les protocoles applicatifs qui utilisent UDP incluent :

- SNMP
- DNS
- DHCP
- Trivial FTP (TFTP)
- VoIP

En-tête UDP

L'en-tête UDP est bien-sûr bien plus simple et court que l'en-tête TCP.

Il n'a que 4 champs et fait seulement 8 octets.



— **Source Port** (port source) — 16 bits
Identifie l'application source par son numéro de port.
— **Destination Port** (port de destination) — 16 bits
Identifie l'application de destination par son numéro de port.
— **Length** (longueur) — 16 bits
Indique la longueur de l'en-tête du datagramme UDP.
— **Checksum** (checksum) — 16 bits
Pour la vérification d'erreurs.

Réassemblage d'un datagramme UDP

Comme pour TCP, les datagrammes UDP envoyés prennent souvent différents chemins et donc n'arrivent pas dans l'ordre. Mais contrairement à TCP, UDP ne suit pas de numéro de séquence. UDP n'a donc aucun moyen de réassembler les datagrammes. Il se contente juste de réassembler les données dans l'ordre dans lequel elles sont reçues avant de les envoyer à la couche Application. Si l'ordre des données est important pour l'application, ce sera à elle d'identifier la bonne séquence.

6.3 Numéros de port

Quel que soit le protocole (TCP ou UDP), la couche Transport utilise des numéros de port pour identifier les applications et leurs conversations. Ces communications étant simultanées pour plusieurs applications, les numéros de port permettent de transmettre les données désencapsulées au bon processus applicatif.

Dans un échange client/serveur, le serveur est assigné un port (souvent défini en fonction du protocole pour les *well-known ports*, 1). On dit qu'il *écoute* sur ce port.

Le client en revanche initie la conversation. Il n'a donc pas besoin d'avoir un port fixe. Le port du client sera généré dynamiquement pour identifier la conversation. Chaque requête faite par un hôte aura donc un port client différent, ce qui permet d'avoir de multiples conversations simultanées.

Certains *well-known ports* à connaître :

— 20 : FTP (données)
— 21 : FTP (contrôle)
— 22 : SSH
— 23 : Telnet
— 25 : SMTP
— 465 et 587 : Simple Mail Transfer Protocol Secure (SMTPS)

— `53` : DNS
— `80` : HTTP
— `443` : HTTPS
— `110` : Post Office Protocol version 3 (POP3)
— `995` : Post Office Protocol version 3 Secure (POP3S)
— `161` : SNMP (pour l'agent)
— `162` : SNMP (pour la réception des traps sur le manager)

On appelle *socket* la combinaison adresse IP source / port source ou adresse IP de destination / port de destination. Mis ensemble, le socket du serveur et le socket du client forment une paire de socket (`socket pair`).

C'est grâce à ces paires de socket que les processus d'un client peuvent se distinguer les uns des autres, et que de multiples connexions à un processus serveur peuvent se distinguer.

6.3.1 Groupes de numéros de port

L'Internet Assigned Numbers Authority (IANA) est l'organisation responsable d'attribuer différents standards d'addressage. Cela inclut les numéros de port de la couche Transport.

Ces numéros de port sont codés sur 16 bits, ce qui veut dire qu'il peut exister jusqu'à $2^{16} = 65536$ ports différents (de 0 à 65535).

L'IANA a divisé cet intervalle en trois groupes de ports :

1. **Well-known ports** (de 0 à 1023) :
   — Réservés pour les services connus et communs comme le web, le mail, etc.
   — Permettent aux clients d'identifier facilement le type de service associé à une application.
2. **Ports réservés** (de 1024 à 49151) :
   — Ces numéros sont attribués par l'IANA aux entités faisant une demande pour une application spécifique.
   — Il s'agit en général d'applications individuelles que l'utilisateur a installé.
3. **Ports privés et/ou dynamiques** (de 49152 à 65535) :
   — On les appelle aussi des ports *éphémères*.
   — L'Operating System (OS) client les assigne en général dynamiquement lors de l'initialisation d'une connexion.
   — Identifie l'application client.

Chapitre 7

Couche Application

7.1 DHCPv4

7.1.1 Présentation

Le protocole DHCP assigne dynamiquement des adresses IPv4 et d'autres éléments de configuration réseau.

Un serveur DHCP est assez facile à maintenir, mais dans une très petite organisation, il est possible de configurer un service DHCP sur un routeur Cisco.

Le serveur assigne les adresses IPv4 à partir d'un pool d'adresses pour une période donnée, ou jusqu'à ce que le client n'en ait plus besoin. Le bail dure en général entre 24 heures et une semaine voire plus. Une fois le bail écoulé, le client doit demander une nouvelle adresse, même si, en pratique, le serveur lui réassigne souvent la même adresse.

7.1.2 Fonctionnement

DHCPv4 fonctionne en mode client/serveur. L'attribution de configuration réseau se fait par bail, ce qui oblige le client à faire des demandes régulières pour étendre les baux. Ce mécanisme permet aux clients qui déménagent ou sont éteints de ne pas garder d'adresses dont ils n'ont plus besoin. Quand un bail expire, l'adresse retourne dans le pool pour pouvoir être réattribué si nécessaire.

Quand le client démarre ou veut rejoindre le réseau, il entame un processus en 4 étapes pour obtenir un bail :

1. `DHCPDISCOVER` — Le client démarre le processus avec un broadcast `DHCPDISCOVER` avec son adresse MAC pour découvrir les serveurs DHCP disponibles. Comme il n'a à ce stade pas d'information IPv4, il utilise des broadcasts de couche 2 et 3.
2. `DHCPOFFER` — Une fois que le serveur reçoit un `DHCPDISCOVER`, il réserve une adresse IPv4 disponible pour un bail. Le serveur crée aussi une entrée ARP contenant l'adresse MAC du client avec l'adresse IPv4 réservée pour lui. Il envoie ensuite un message `DHCPOFFER` au client contenant l'adresse allouée.
3. `DHCPREQUEST` — Le client reçoit le `DHCPOFFER` du serveur, il envoie maintenant un `DHCPREQUEST`. Ce message est utilisé à la fois pour une nouvelle demande et pour

un renouvellement de bail. Quand le message est utilisé pour une demande de bail, le `DHCPREQUEST` sert à indiquer une acceptation au serveur sélectionné, et un refus implicite à d'autres serveurs qui auraient pû fournir un `DHCPOFFER`.

Beaucoup de réseaux utilisent plusieurs serveurs DHCP. Le `DHCPREQUEST` est un broadcast pour informer le serveur DHCP et tout autre serveur DHCP de l'offre acceptée.

4. `DHCPACK` — Quand le serveur reçoit un `DHCPREQUEST`, le serveur vérifie l'information du bail en faisant un `ping` Internet Control Message Protocol (ICMP) à cette adresse pour s'assurer qu'elle n'est pas déjà utilisée. Il crée une nouvelle entrée ARP et répond avec un `DHCPACK`. Ce message duplique le `DHCPOFFER`, à la différence d'un changement dans le champ `Message Type`. Quand le client reçoit un `DHCPACK`, il enregistre la configuration et opère un `ARP lookup` pour l'adresse assignée. S'il ne reçoit pas de réponse, le client peut utiliser l'adresse.

Pour renouveler un bail, le client démarre un processus en 2 étapes :

1. `DHCPREQUEST` — Avant que le bail n'expire, le client envoie un `DHCPREQUEST` directement au serveur DHCPv4 avec qui le bail a été signé. Si un `DHCPACK` n'est pas reçu après un certain temps, le client renvoie un `DHCPREQUEST`, en broadcast cette fois, pour qu'un autre serveur DHCP puisse étendre le bail.
2. `DHCPACK` — À la réception du `DHCPREQUEST`, le servuer vérifie l'information du bail en retournant un `DHCPACK`.

La RFC 2131 autorise l'envoi de ces messages (`DHCPREQUEST` et `DHCPACK`) en unicast ou en `broadcast`.

7.1.3 Configuration d'un serveur DHCPv4 Cisco

Si on n'a pas de serveur DHCP dédié, on peut configurer un routeur Cisco pour qu'il assure un service DHCP.

Il y a 3 étapes pour configurer un serveur DHCPv4 sur un routeur Cisco :

1. **Exclure des adresses IPv4** — Le routeur assigne toutes les adresses IPv4 d'un pool DHCP sauf s'il est configuré pour exclure certaines adresses. En pratique, on assigne certaines adresses du pool à des périphériques réseaux ayant besoin d'une adresses statique. Il faut alors réserver ses adresses pour ne pas les assigner à d'autres périphériques. La commande est la suivante :

```
Router(config)# ip dhcp excluded-address <low-address> [<high-address>]
```

   On peut donc exclure une adresse ou une étendue d'adresses. Les adresses doivent être celles qu'on compte configurer manuellement sur des périphériques devant recevoir des adresses statiques.

2. **Définir un nom de pool DHCPv4** — La création du pool met le routeur dans un mode de configuration DHCPv4 :

```
Router(config)# ip dhcp pool <nom>
Router(dhcp-config)#
```

3. **Configurer le pool DHCPv4** — Le pool d'adresses et le routeur passerelle par défaut doivent être configurés. Pour définir l'étendue d'adresses disponibles, on utilise le mot clé `network`. Pour définir le routeur passerelle par défaut, on utilise le mot clé `default-router`. La passerelle est en général l'interface LAN du routeur qui est la plus proche des périphériques clients. Une passerelle est requise, mais on peut lister jusqu'à 8 adresses s'il y a plusieurs passerelles.
   Les autres commandes de pool DHCPv4 sont optionnelles.
   — Définir le pool d'adresses :

   ```
   Router(dhcp-config)# network <numéro> [<masque> | </cidr>]
   ```

   — Définir la passerelle par défaut :

   ```
   Router(dhcp-config)# default-router <adresse> [<adresse2> ... <
       adresse8>]
   ```

   — Définir un serveur DNS :

   ```
   Router(dhcp-config)# dns-server <adresse> [<adresse2> ... <adresse8>]
   ```

   — Définir le nom de domaine :

   ```
   Router(dhcp-config)# domain-name <domaine>
   ```

   — Définir la durée du bail :

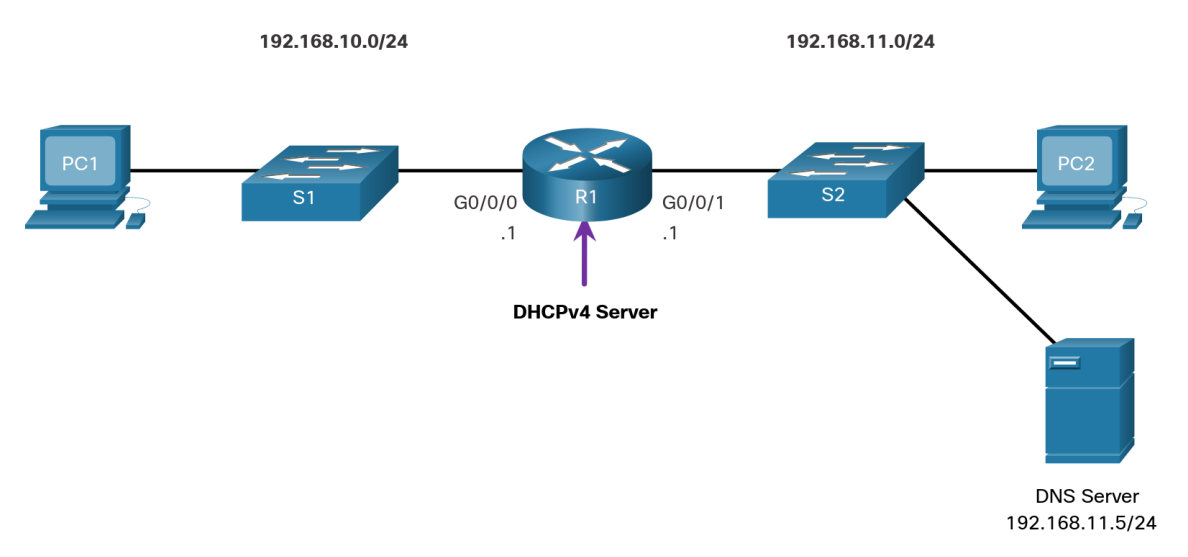
   ```
   Router(dhcp-config)# lease {<jours> [<heures> [<minutes>]] | infinite}
   ```

   — Définir le serveur Windows Internet Name Service (WINS) netBIOS :

   ```
   Router(dhcp-config)# netbios-name-server <adresse> [<adresse2> ... <
       adresse8>]
   ```

   Il faut noter que Microsoft recommande de ne pas déployer WINS, mais plutôt de configurer le DNS pour Windows.

Voici un exemple de configuration pour la topologie suivante :




```
R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9
R1(config)# ip dhcp excluded-address 192.168.10.254
R1(config)# ip dhcp pool LAN-POOL-1
```

```
R1(dhcp-config)# network 192.168.10.0 255.255.255.0
R1(dhcp-config)# default-router 192.168.10.1
R1(dhcp-config)# dns-server 192.168.11.5
R1(dhcp-config)# domain-name exemple.com
R1(dhcp-config)# end
R1#
```

Commandes de vérification

Afficher les commandes DHCPv4 configurées sur le routeur :

```
Router# show running-config | section dhcp
```

Afficher une liste de toutes les correspondances entre adresses IPv4 et adresses MAC fournies par le service DHCP :

```
Router# show ip dhcp binding
```

Afficher un compteur des messages DHCPv4 envoyés et reçus :

```
Router# show ip dhcp server statistics
```

Désactivation du serveur DHCPv4 sur Cisco

Le service DHCP est activé par défaut. Activer le service n'a pas d'effet si aucune configuration n'est présente.

```
R1(config)# no service dhcp
R1(config)# service dhcp
```

7.1.4 DHCPv4 Relay

Dans un réseau complexe, les serveurs d'entreprise sont en géréral situés au centre. Ces serveurs peuvent fournir des services DHCP, DNS, FTP... Les clients sont donc probablement dans un sous-réseau différent. Pour localiser les serveurs et recevoir des services, les clients utilisent souvent des broadcasts.

Imaginons qu'un `PC1` tentent d'acquérir une adresse IPv4 en envoyant un broadcast. Si le routeur de son réseau ne fournit pas de service DHCP et ne transfère pas le broadcast, le serveur DHCP étant sur un autre réseau, `PC1` ne pourra pas recevoir d'adresse IPv4. Il faut donc que le routeur soit configuré pour transférer (*relay*) les messages DHCPv4 au serveur DHCP.

Pour configurer le relai sur un routeur Cisco :

```
R1(config)# interface g0/0/0
R1(config-if)# ip helper-address 192.168.11.6
R1(config-if)# end
R1#
```

Ceci va permettre à `R1` de relayer les messages DHCPv4 au routeur DHCPv4. Cette configuration est faite sur l'interface recevant le broadcast.

Une fois cette configuration faite, le routeur transfère les broadcasts DHCPv4 en tant qu'unicast au serveur mentionné. On peut vérifier l'application de la configuration :

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

Le PC1 peut maintenant recevoir ses configurations DHCPv4 normalement.

7.1.5 D'autres relais de broadcast de service

DHCPv4 n'est pas le seul service pouvant être relayé par un routeur. Par défaut, la commande `ip helper-address` transfère 8 services UDP :

1. port 37 : temps
2. port 49 : Terminal Access Controller Access-Control System (TACACS)
3. port 53 : DNS
4. port 67 : DHCP/BOOTstrap Protocol (BOOTP) serveur
5. port 68 : DHCP/BOOTP client
6. port 69 : TFTP
7. port 137 : netBIOS service de nom
8. port 138 : netBIOS service datagramme

7.1.6 Configuration d'un client DHCPv4 Cisco

S'il y a un serveur DHCP, par exemple par le FAI, on peut configurer un routeur Cisco en tant que client DHCP.

Pour configurer une interface en tant que client DHCP :

```
R1(config-if)# ip address dhcp
```

Cette configuration est souvent requise par le FAI sur le routeur qui sert de passerelle (qui donne accès à internet au réseau local).

7.2 Fonctionnement de FTP

7.2.1 Mode normal

FTP utilise deux ports TCP : 21 pour le contrôle (la connexion et les commandes), et 20 pour les données.

Le port 21 est donc le port destination, initialisé par le client (port source aléatoire supérieur à 1024). Pour cela, le client fait sa requête avec la commande `port` qui démarre une connexion en mode normal. Le port 20 est le port *source*, initialisé par le serveur, sur un port destination fourni par le client lors de la phase précédente.

Cela veut dire que d'un point de vue sécurité, pour autoriser une connexion FTP, il faut autoriser n'importe quelle machine du monde à initialiser une connexion sur n'importe quel port interne en prenant comme port source 20. La solution est bien sûr d'utiliser un firewall stateful.

7.2.2 Mode passif

Même chose pour l'initialisation : le client démarre une connexion sur le port 21 du serveur en destination. La commande du client pour entrer en mode passif est `pasv`. Dans sa réponse, le serveur inclut un numéro de port aléatoire d'écoute. Il n'utilise plus le port 20 pour les données, mais un port aléatoire.

De plus, contrairement au mode normal, c'est le client qui initialise les deux connexions.

7.3 VoIP

7.3.1 VoIP — ToIP

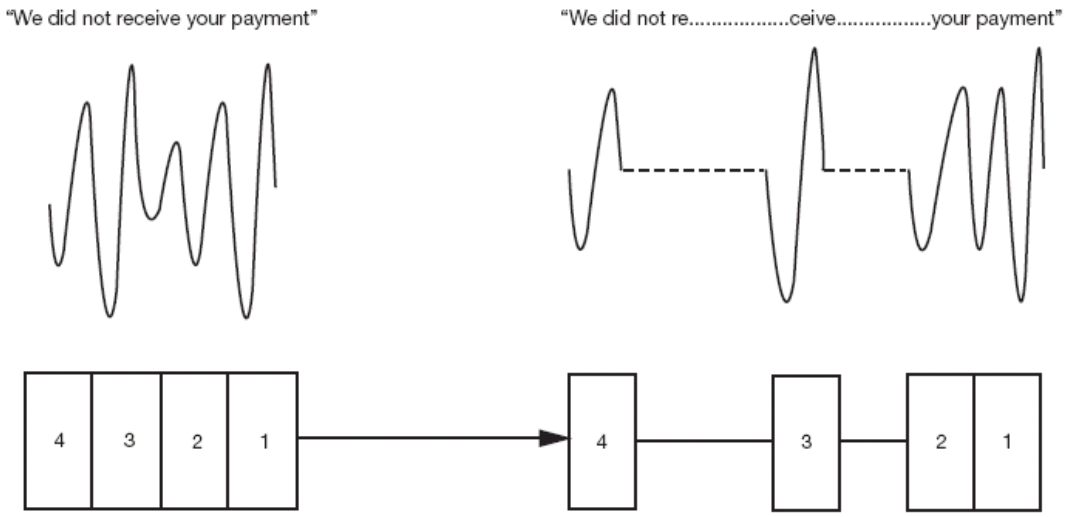
Différence entre Voice over IP (VoIP) et Telephony over IP (ToIP) :

- **VoIP** — voix sur IP.
  Qualifie le transport de la voix sous forme de paquet IP entre deux points d'un réseau donné, tout en s'affranchissant des contraintes qu'elle impose (latence, gigue, perte, etc. . .).
- **Telephony over IP (ToIP)** — téléphonie sur IP.
  Qualifie un service de communication entre deux téléphones IP (*IP phone*, *soft phone*, *IP Private Branch eXchange (IPBX)*, etc. . .). Pour ces services, un ensemble de fonctionnalités de téléphonie sont mises en œuvre (présentation du numéro de l'appelant, transfert d'appel, suspension d'appel, communication de groupe, rappel, etc. . .).

7.3.2 Contraintes

- **Le débit** — Pas un problème puisque grâce aux codecs on peut descendre à 5.6Kbps.
- **Le temps de transmission** — L'application est interactive, donc il ne doit pas y avoir de délai trop important entre la source et la destination. Pour se mettre d'accord sur les temps acceptables, la norme *G.114* a été créée :
  - 0 à 150 ms : acceptable pour la plupart des applications utilisateur.
  - 150 à 400 ms : acceptable sous réserve que les administrations connaissent l'effet du temps de transmission sur la qualité de transmission.
  - plus de 400 ms : inacceptable pour la planification générale des réseaux. Il est cependant reconnu que cette limite pourra être dépassée dans certains cas exceptionnels.

  En résumé, le délai maximum acceptable de bout en bout dans un sens de transmission doit être inférieur à 300 ms.
- **La gigue (*jitter*)** — Il s'agit de la variation du temps de transmission des informations. Il faut pouvoir reconstituer à l'arrivée le décalage relatif du départ. Toute variation dans le décalage à l'arrivée créera soit des intervalles de silences soit des troncatures de mots. La gigue est générée par la variation de charge réseau.



— **L'écho** — L'écho est habituel mais imperceptible quand il est quasiment instantané. Si le décalage est plus important, il devient gênant. Le phénomène est plus fréquent à l'international (à cause de la distance de la liaison).

Au vu de ces contraintes, IP et à fortiori internet paraissent incompatibles avec un transport en temps réel.

Cependant, si on peut garantir :

— un débit < 64Kbps
— un temps de latence ≤ 150 ms (acceptable jusqu'à 300 ms)
— une gigue dans des proportions raisonnables
— une perte réduite de paquets
— pas trop d'écho

... il est alors possible d'utiliser le protocole IP pour gérer la VoIP.

7.3.3 Numérisation

La convergence des réseaux (la VoIP utilise l'IP) nécessite de numériser la voix humaine pour la transporter.

Les travaux de Harry Nyquist et Claude Shannon permettent de reconstruire fidèlement un signal continu dans le temps à partir du signal échantillonné si la fréquence d'échantillonnage est au moins égale au double de la fréquence maximale du signal que l'on souhaite numériser.

Pour réaliser cette numérisation, il faut :

— d'un Convertisseur Analogique Numérique (CAN) (Analog to Digital Converter (ADC))
— d'un codec
Aujourd'hui le codec le plus utilisé est le codec Pulse Code Modulation (PCM).

L'opération de numérisation se fait en trois étapes :

1. **L'échantillonnage**

En téléphonie analogique, la fréquence allouée pour la voix humaine est d'environ 4KHz (de 300 à 3400Hz). Selon la règle ci-dessus, la fréquence d'échantillonnage sera donc de 8KHz, c'est-à-dire 8000 échantillons par seconde, ou un échantillon toutes les 125μs.

2. **La quantification**

   Une fois que le signal est échantillonné, l'amplitude de chaque échantillon est mesurée (on dit *quantifiée*) par rapport à une échelle de référence, sur plusieurs échelons. Dans l'opération de quantification, les erreurs d'approximation commises dans la mesure des échantillons introduisent des erreurs d'autant plus importante que le nombre d'échelons est faible. Pour limiter cette erreur, chaque échantillon est codé sur 8 bits (1 bit pour le signe et 7 bits pour l'échelon). Cela donne 128 échelons pour les valeurs positives et 128 échelons pour les valeurs négatives. Ces 256 échelons au total sont universellement adoptés dans le monde de la téléphonie.

   Quand on quantifie un signal dont le niveau est faible, ces échelons ne seront pas assez précis (une erreur de 1 sur l'échelon 125 fait une erreur de $\frac{1}{125}$, alors qu'une erreur de 1 sur l'échelon 5 fait une erreur de $\frac{1}{5}$). Pour palier à ça on modifie l'échelle de quantification avec une fonction de transfert non linéaire.

   Cette fonction de transfert, logarithmique, est différente en Europe (*a-law*) et aux USA (*u-law*).

3. **Le codage**

   Chaque échelon de l'échelle de référence ayant servi à la quantification correspond à une valeur binaire. Pour véhiculer de la voix humaine brute (sans compression), il faut un débit de 8 bits, 8000 fois par seconde, c'est-à-dire 64000 bits/s, ou 64Kbps. Cette technique de numérisation a été normalisée sous la norme G.711.

7.3.4 Transport de la voix numérisée

Dans les réseaux téléphoniques classiques Real-Time Communication (RTC) et Réseau Numérique à Intégration de Service (RNIS) (Integrated Services Digital Network (ISDN) en anglais), qui utilisent la technique Time Division Multiplexing (TDM), après que la voix est numérisée, chaque échantillon est envoyé en série, un échantillon toutes les 125μs.

Mais dans les réseaux IP cela serait une perte de ressources. Le signal analogique est découpé en séquences jointives de $x$ms, avec x dépendant du type de codec utilisé :

G.711	10ms
G.729	10ms
GSM 6.10	20ms
G.723.1	30ms

Chaque séquence est numérisée puis compressée pour donner un paquet. Cela s'appelle la *paquetisation*. La paquetisation introduit un délai qui correspond au minimum à la durée de la séquence d'analyse du codec, auquel il faut ajouter le temps de compression.

Le débit résulte donc directement du type de codec.

Pour améliorer les temps de transmission, on commute des paquets de petite taille et on utilise un VLAN (`802.1q`) dédié voix. Des techniques de QoS assurent aux paquets VoIP un traitement prioritaire. Les trames ethernet appartenant aux VLAN voix seront taggées

en `802.1p` pour qu'ils soient traités de façon prioritaire sur les ports trunk (`802.1q`) des switchs.

7.3.5 Codec

Le débit de base résultant de la numérisation est de 64Kbps (G.711). Pour diminuer ce débit, il y a d'autres techniques de codage :

— **Codage différentiel** — permet de diminuer le débit jusqu'à 16Kbps.
Repose sur le fait que deux échantillons proches ont à peu près la même valeur, on peut donc ne transmettre que la différence. Comme cette différence est petite, on peut la coder sur moins de bits.

Désignation	Différentiel	Fréquence	bits de codage	Débit	Norme
PCM (origine)	non	8KHz	8	64Kbps	G.711
ADPCM 6	oui	8KHz	6	48Kbps	G.722
ADPCM 4	oui	8KHz	4	32Kbps	G.722.1
ADPCM 3	oui	8KHz	3	24Kbps	G.722.1
ADPCM 2	oui	8KHz	2	16Kbps	G.722.1

— **Codage par synthèse** — (plus complexe) autorise des taux de compression plus importants, ce qui diminue le débit à quelques Kbps.
Les techniques de codage différentiel sont simples mais ne peuvent pas descendre sous 16Kbps. Le codage par synthèse modélise la parole au lieu de mesurer le signal. Les données à transmettre sont alors beaucoup plus légères.
On stocke des formes d'onde correspondant à des phonèmes dans un dictionnaire. Lors de la numérisation, la voix est codée suivant le dictionnaire. À la réception, le signal source est reconstitué par le codec
Cette méthode donne des débits plus faibles mais sollicitent d'avantage le CPU.

7.3.6 Compression des silences

Dans une conversation, il y a une grande quantité de silences. On peut donc encore gagner en débit en cessant de produire des trames de voix quand on ne détecte aucune activité.

Il y a 3 composants principaux de la compression des silences :

— **Voice Activity Detection (VAD)** — Pour détecter les zones de silences, il faut un temps de réponse court, sinon on risque de perdre le premier mot de reprise d'activité vocale, ou d'ajouter des périodes de silence après des périodes actives.
— **Discontinuous Transmission (DTX)** — Permet au codec d'arrêter la transmission quand le module VAD a détecté une période de silence.
— **Confort Noise Generation (CNG)** — Pour transitionner entre une phase d'activité et de silence, on peut ajouter du bruit de confort. De plus, quand personne ne parle, le silence complet peut être gênant et peut faire penser que la communication a été stoppée. Le CNG vise à recréer une ambiance sonore.

La suppression des silences est native dans les codec G.723 et G.729. Pour le codec G.711, il faut l'implémenter séparément.

7.3.7 MOS, qualité d'un codec

La variété des algorithmes de compression a conduit à un besoin de comparaison de leurs qualités. Pour cela on utilise le MOS. Ce score se base sur un sondage d'un échantillon supposé représentatif de la population des utilisateurs.

Les valeurs de MOS sont :

1. Mauvais
2. Médiocre
3. Moyen
4. Bon
5. Excellent

Normes pour les codec audio :

	Débit	Codec	Complexité	Retard	MOS	Utilisation
G.711	64 Kbps	PCM	0.1 Mips	125 μs	4.2	RNIS, VoIP
G.722.1	16, 24, 32 Kbps	ADPCM	10 Mips	300–375 μs	4.0	RNIS, VoIP
G.723.1	6.3, 5.3 Kbps	ACELPM P-MLQ	16, 18 Mips	80–90 ms	3.8	VoIP
G.726	16, 24, 32, 40 Kbps	ADPCM	12 Mips	250–375 μs	3.85	
G.727	16, 24, 32, 40 Kbps	ADPCM	12 Mips	250–375 μs	4.0	
G.728	16 Kbps	LD-CELP	33 Mips	2.5–3 ms	4.0	GSM, VoIP
GSM 6.10	13 Kbps	RPE-LTP	2.5 Mips	50 ms	3.6	VoIP, DECT
G.729	8 Kbps	CS-ACELP	22 Mips	10 ms	3.9	VoIP

7.3.8 IP en temps réel

Le protocole IP n'est pas fait pour gérer des applications en temps réel. Pour cela il faut lui ajouter d'autres protocoles.

L'importance du temps de latence ne permet pas d'utiliser TCP. On utilise donc pour le temps réel le protocole de transport UDP. Il faut cependant gérer la connexion en mode connecté. Sans TCP, il faudra ajouter d'autres protocoles :

— RTP (RFC 1889) — Permet la reconsitution des propriétés temps réel des flux grâce à l'horodatage et le séquencement des paquets.
— RTCP (RFC 3611) — Protocole de contrôle des flux RTP, permet de véhiculer des informations sur la qualité de la distribution (gigue, délai, taux de perte de paquets, etc...), sur les participants d'une session, etc...

Les protocoles RTP et RTCP font partie de la couche transport (4) du modèle OSI.

RTP

RTP fournit des informations sur le type véhiculé, la gigue, et les pertes de paquets. Ceci permet aux applications de :

— synchroniser pour délivrer un flux isochrone.
— restaurer les paquets perdus pour délivrer une information de contenu exploitable.

RTP utilise les numéros de port UDP pairs.

RTCP

Les messages RTCP effectuent la surveillance de chaque flux RTP en transportant des informations sur les participants de la session et la qualité de la distribution.

RTCP utilise les numéros de port UDP impairs (immédiatement supérieur à celui utilisé par RTP).

7.3.9 Protocoles de signalisation

Les protocoles IP, UDP, RTP et RTCP à eux seuls ne suffisent pas à faire de la ToIP. Il faut lui ajouter des protocoles permettant de véhiculer et interpréter efficacement la signalisation (ensemble des processus pour établir, maintenir et libérer une communication).

Ces protocoles sont :

— **H.323** de la ITU-T
Date de 1996, c'est la suite de protocole la plus utilisée aujourd'hui. Utilise un équipement particulier pour acheminer la signalisation : le *gatekeeper*.
— **Session Initiation Protocol (SIP)** de l'IETF — RFC 2543
Date de 1999, protocole beaucoup plus léger que *H.323* et adapté à l'IP. Utilise en général un proxy SIP.
— **MGCP** de l'IETF — RFC 2705
Initialement développé pour piloter des passerelles, il est également utilisé pour installer des IP phones.

Ces 3 protocoles sont des standards ouverts, ce que est bénéfique pour l'utilisateur mais les entreprises préfèrent développer des protocoles propriétaires :

— **Skinny Client Control Protocol (SCCP)** de Cisco
— **Universal Alcatel (UA)**
— **UniStim** de Nortel
— **Minet** de Mitel
— **Inter Asterisk eXchange (IAX)**

7.3.10 Composants d'une architecture ToIP

Peu importe le protocole utilisé, l'architecture ToIP s'appuie sur ces composants :

- **un serveur de contrôle** — Gatekeeper H.323, Call Manager Cisco, Proxy SIP, Call Agent, Call Switch... Permettent l'enregistrement et l'authentification des équipements.
- **une passerelle** — Ou Gateway, permettant la communication entre les réseaux IP et la téléphonie traditionnelle.
- **des terminaux IP** — Ou IP phone, soit physiques (hard phone), soit émulés par un logiciel sur PC (soft phone).
- **des adaptateurs** — Permettent à des téléphones traditionnels de s'interfacer sur un réseaux IP.

Le serveur de contrôle peut être amené à communiquer avec d'autres serveurs de données (messagerie, annuaire...). Il se peut qu'un équipement héberge plusieurs composants : IPBX.

7.3.11 Services de téléphonie

Le fait de fusionner le réseaux téléphonique avec le réseau IP permet l'ajout de services plus facilement qu'avec la téléphonie analogique :

- répondeur téléphonique
- consultation du message du répondeur sur le PC
- écoute des e-mails reçus convertis au format sonore
- indicateur de présence
- via le client de messagerie unifié, appel d'un correspondant en cliquant sur un nom dans le carnet d'adresses
- visioconférence
- chat

7.3.12 Le protocole SIP

Introduction

Le protocole SIP a été initialisé par le groupe Multiparty MUtimedia SessIon Control (MMUSIC). Il est aujourd'hui repris et maintenu par le group SIP de l'Internet Engineering Task Force (IETF) donnant la RFC 3261 et rendant obsolète la RFC 2543.

L'objectif de SIP est de concevoir un protocole simple et rudimentaire pour établir une session multimédia entre terminaux de toute nature. Sa syntaxe reprend les syntaxes de HTTP et SMTP : les commandes et messages sont codés en texte, ce qui facilite l'interaction avec d'autres applications.

Format des messages SIP

- **Type de message** — L'architecture de SIP est de type client / serveur. Les messages peuvent donc être soit des requêtes (de client), soit des réponses (de serveur).
- **Method** — En HTTP on a `GET`, `POST`, `PUT`, etc. En SIP on a :

- Méthodes de base
  - `Invite` — Initialise une session.
  - `Ack` — Acquittement final des réponses codées : 2XX, 3XX, 4XX, 5XX et 6XX.
  - `Bye` — Met fin à une session.
  - `Options` — Permet d'échanger la capacité et l'état d'un UA.
  - `Cancel` — Annule une requête.
  - `Register` — Enregistre un terminal à un service de localisation.
- Méthodes étendues
  - `Info` — Permet l'échange de signalisation pendant la session.
  - `Refer` — Permet le transfert d'un utilisateur vers une autre ressource.
  - `Prack` — Acquittement provisoire pour les réponses de type 1XX.
  - `Subscribe` — Permet de s'enregistrer à un serveur pour ensuite être notifié d'évènements.
  - `Notify` — Permet à l'agent d'être notifié d'évènements.
  - `Message` — Permet le transport de message instantané
  - `Update` —
    - Mise à jour de l'état de l'agent lors d'un service de présence
    - Mise en attente du destinataire.
- **URI** — Dans tout réseau il faut pouvoir identifier un utilisateur. Le format des URI est emprunté au format d'adressage de SMTP :
  - `sip:yelena.marie@free.fr`
  - `sip:fany:password@fresnes.com;transport=tcp`
  - `sip:+33164688186@nogent.net;user=phone`
- **Header** — L'en-tête contient les informations suivantes :
  - *From*
  - *To*
  - *Request-URI* : destination actuelle
  - *Contact* : apparaît dans les méthodes `Invite`, `Options`, `Ack`, `Register` et dans les réponses. Il indique l'adresse de réponse directe à laquelle seront envoyées toutes les nouvelles transactions. Il inclut aussi l'adresse de redirection dans les réponses `3XX` et `485`. Dans les erreurs `4XX`, `5XX` et `6XX`, il contient des informations supplémentaires. Dans les méthodes `Register` il inclut la position actuelle. Pour les contacts multiples, un champ `Header` peut être inclus.
- **Message body** — Il transporte les informations en Session Description Protocol (SDP) pour permettre la participation à une session. Il décrit les éléments suivants :
  - Média à utiliser (codec, taux d'échantillonnage)
  - Destination du média (adresse IP et port)
  - Nom et but de la session
  - Temps pendant lequel la session est active
  - etc.

Les réponses

Les réponses SIP véhiculent un code d'état sur 3 chiffres comme en HTTP : un chiffre de classe et deux de motifs.

Les classes sont les suivantes :

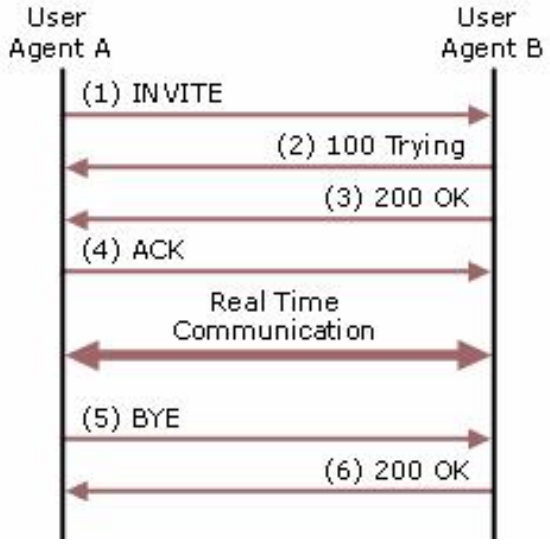
— **Informational** (code 1XX)
— **Success** (code 2XX)
— **Redirection** (code 3XX)
— **Client Error** (code 4XX)
— **Server Error** (code 5XX)
— **Global Failure** (code 6XX)

Échanges SIP

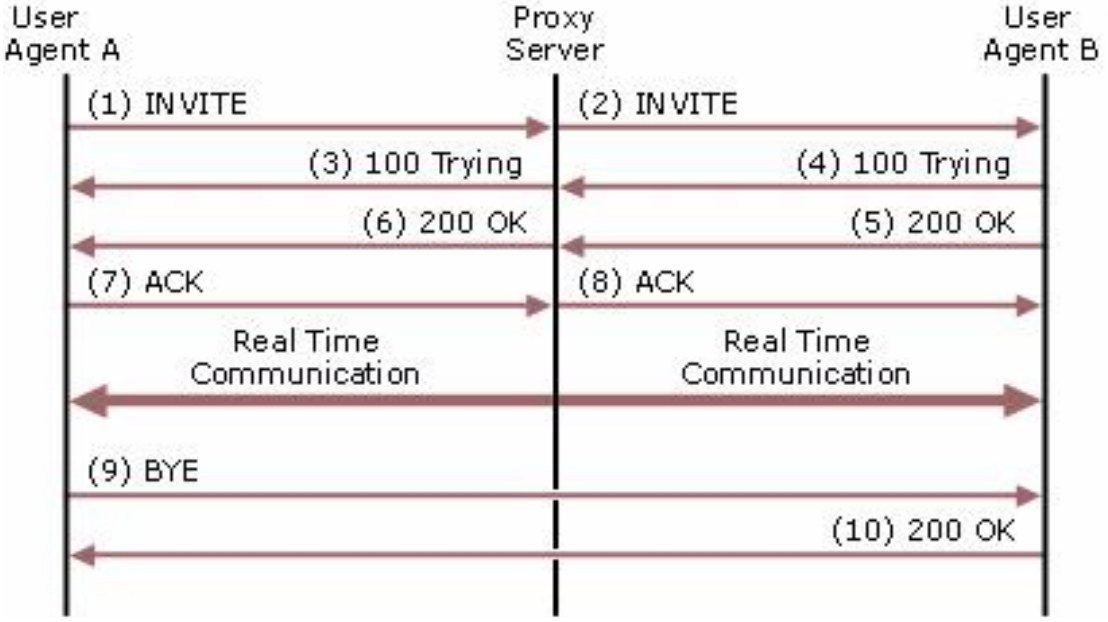
Le rôle de SIP est d'ouvrir, de modifier et de libérer des sessions. Pour ouvrir une session, un utilisateur émet une requête `Invite` avec un descripteur de session qui permet à l'utilisateur appelé de s'accorder sur la comptabilité de leur média.

Cette session peut être réalisée avec ou sans proxy SIP.

— **Sans proxy** — Le proxy est optionnel : deux équipements SIP peuvent communiquer directement s'ils connaissent leurs adresses IP.



— **Avec proxy** — Pour rejoindre des réseaux différents ou lorsque l'équipement ne connaît pas l'adresse IP distante.



7.3.13 Le protocole MGCP

Introduction

Le protocole MGCP est un protocole de l'IETF développé en 1999 pour piloter des passerelles.

Les protocoles SIP et `H.323` supposent que le terminal est suffisamment intelligent pour gérer seul des fonctions comme la tonalité à numéroter, la mise en attente d'appels, etc. MGCP est différent : c'est un protocole maître/esclave optimisé pour le contrôle de dispositifs sans intelligence.

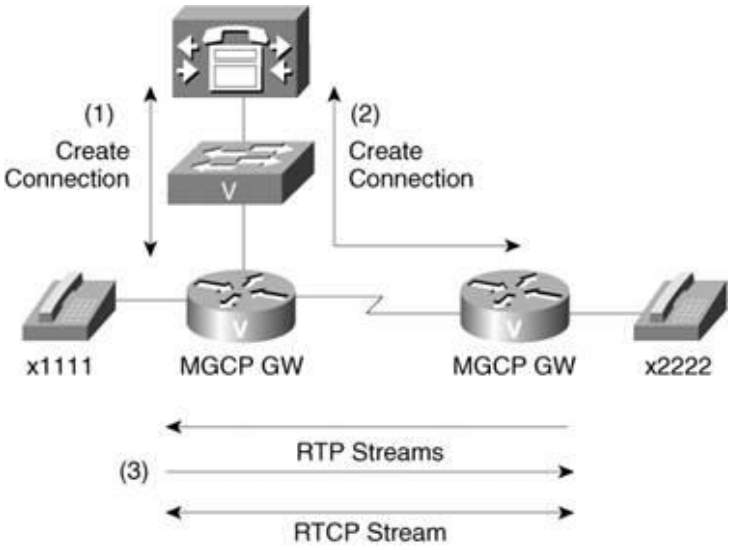
MGCP est donc complémentaire de SIP et `H.323`.

Le protocole MGCP peut être connu sous le nom de MEdia GAteway COntrol (MEGACO) (RFC 3015).

Les composants de MGCP

Il y a deux principaux composants en MGCP :

1. **Media Gateway (MG)**
   - passerelle résidentielle (modem ADSL, câble modem, ...)
   - Private Branch eXchange (PABX), router
   - IP phone
2. **Media Gateway Controller (MGC)**
   - consiste essentiellement dans le contrôle d'appel
   - analyse des frappes de touches
   - routage des appels
   - éventuellement, conversion de la signalisation



7.3.14 Le protocole `H.323`

Introduction

`H.323` est développé par l'ITU-T. L'origine est le développement de la visioconférence sur les réseaux locaux dans la continuité de la norme `H.320` (visioconférence sur RNIS).

H.323 est parfaitement compatible avec les systèmes de visioconférence H.320 mais il est lourd et rigide.

H.323 est en fait une norme qui décirt un cadre général de fonctionnement pour un terminal multimédia.

Les autres normes se référant à la signalisation sont :

— H.255.0 — définition de la signalisation d'appel
— H.245 — gestion des flux multimédias
— H.450 — recommandations de services supplémentaires

Les composants de H.323

Une architecture H.323 renferme plusieurs entités :

— **End Point (EP)**, qui peuvent être :
  — **Terminaux IP** — Soit physiques (hard phone), soit émulés (soft phone).
    — initie et reçoit les appels
    — initie les sessions H.323
    — initie et termine les connexions H.245
    — génère et encode les flux voix
    — implémente les services supplémentaires
  — **Gateway** — Passerelle permettant le fonctionnement avec d'autres réseaux. Chaque réseau utilise ses propres protocoles de signalisation. La passerelle doit donc traduire chaque protocole. Elle convertit aussi les formats de transmission audio, vidéo et de données.
  — **Multipoint Control Unit (MCU)** — Équipement optionnel. Permet les conférences entre plusieurs terminaux. Il négocie le codec audio et vidéo. Il gère l'établissement, le mixage, la diffusion et l'accès de nouveaux terminaux à une conférence en cours.
— **Gatekeeper** — Optionnel mais essentiel quand on veut déployer un service de téléphonie.

7.4 Supervision — SNMP

7.4.1 Introduction

L'information doit être disponible :

— à une population technique variée
— au bon endroit
— au bon moment
— dans sa juste représentation

L'administration du réseau, devenue primordiale, est passée de séparée à intégrée, d'où les noms de *superviseur* et d'*hyperviseur*. On doit avoir une vision globale du réseau.

Gestion d'anomalies

Fault management

Avertit l'administrateur de tout incident survenant dans le réseau : connexion coupée, serveur en panne, routeur défectueux... Il peut alors agir en conséquence :

— soit lui-même à distance en utilisant les fonctions de configuration du réseau : Reset, Reboot, modification de configuration d'un matériel, isolement d'un port...
— soit en faisant intervenir un technicien local

Corrélation d'alarmes

Parfois, de multiples pannes qui, ensemble, peuvent donner l'impression que tout est cassé sont en fait corrélées, et une résolution à la source du problème peut arrêter toutes les alarmes. Pour cela, il faut corréler les alarmes.

Gestion de configuration

La gestion de configuration est composée de 3 parties :

1. la cartographie du réseau
2. la configuration et la reconfiguration distante des matériels
3. la gestion de parcs (des matériels et des logiciels)

Cartographie du réseau

Il y a des applications de type "discovery" qui découvrent automatiquement le réseau et élaborent une représentation graphique correspondante.

Les couleurs d'indication de fonctionnement sont normalisées :

— **rouge** : hors service
— **jaune** : attention dérive dangereuse
— **vert** : bon
— **bleu** : pas de connexion, pas adressable, mais connu dans la base
— **magenta** : erreur
— **gris** : sans erreur mais remontée impossible

Configuration des matériels

La configuration des matériels doit être faite par un personnel qualifié. L'administrateur du réseau doit donc pouvoir, à distance et à partir de sa console, configurer les matériels nouvellement installés. Il doit aussi pouvoir relire ses configurations et les modifier. Idem pour les serveurs et si possible pour les stations de travail.

Gestion des matériels

Certains logiciels de supervision incluent des fonctions de gestion de parc. Cela consiste à établir une Base de Données contenant toutes les informations sur tous les matériels du réseau. L'administrateur peut depuis son poste connaître les caractéristiques matérielles d'un serveur ou d'une station (type de processeur, taille de la mémoire, capacité disque...).

Gestion des logiciels

De la même façon, on peut obtenir une liste des logiciels installés sur une machine, ainsi que leur version. On peut dans certains cas gérer la télé-installation automatique de logiciels et leur mise à jour. On peut également limiter le nombre d'utilisateurs travaillant avec une application en fonction par exemple du nombre de licences déclarées.

Gestion de performance

Permet de savoir, par exemple si un segment ethernet du réseau n'a pas un taux d'utilisation trop important. La plupart des logiciels de supervision permettent de connaître :

— le taux d'utilisation du réseau : sa disponibilité
— le temps de réponse d'un réseau
— le nombre de paquets par seconde
— le nombre de trames broadcast par seconde
— le nombre de trames erronées par seconde
— les stations les plus chargées
— etc.

Ces informations permettent une maintenance proactive : anticiper les pannes.

Gestion de sécurité

On peut implémenter la fonction de sécurité à plusieurs niveaux :

— **Matériels réseau** — On peut réserver les ports à des adresses MAC déterminées ou n'envoyer des trames sur un port que si l'adresse de destination est celle qui est indiquée pour ce port.
— **Stations et serveurs** — Cette fonction est pour l'instant assurée par les OS réseau, mais devrait être intégrée dans le logiciel de supervision.
— **Protection contre les virus** — Des anti-virus peuvent être embarqués dans les logiciels d'administration réseau. Ils tournent en permanence sur les serveurs et lors de la connexion des stations.
— **Sauvegarde des données** — Les plans de sauvegarde peuvent se faire à partir de la console de supervision. En cas d'incident, une alerte remonte sur la console.

Gestion de comptabilité

L'administrateur peut gérer le coût du réseau. Il doit pouvoir connaître le taux d'utilisation pour chaque utilisateur, service, ou département, et ainsi répartir les coûts. Cette fonction n'est pas toujours implémentée sur les logiciels de supervision.

7.4.2 Composants du système d'administration

Il existe 2 types d'entités :

1. La station d'administration : NMS.
2. Les équipements managés : les *agents* (routeurs, switchs, PC, hôtes, BD, serveurs, imprimantes, sondes, etc.).

Ils échangent l'information au moyen d'un protocole : SNMP.

La station d'administration (NMS)

Il s'agit généralement d'une station dédiée recevant un logiciel de supervision.

— Elle contrôle la présence des différents équipements dont elle a connaissance.
— Elle interroge tous les équipements qui disposent d'un agent.
— Elle traite les traps (messages non sollicités).

Les équipements administrés (agents)

Un équipement devant être administré doit disposer d'un agent. C'est un logiciel opérant à l'intérieur de l'équipement. On pourrait dire qu'il joue le rôle d'un "espion" au profit du gestionnaire (*manager*).

— Il répond aux requêtes de la NMS.
— Il effectue sur l'équipement des opérations commandées par la NMS.
— Il émet des traps.

Les protocoles d'administration

La plupart du temps, les équipements utilisent SNMP pour dialoguer avec la NMS, mais on peut aussi trouver :

— Common Management Information Protocol (CMIP)
— Administrative Exchange Protocol (AEP) (propriétaire)
— SNA (propriétaire)

La notion de proxy

Si l'équipement à gérer n'implémente pas le même protocole que la NMS, il doit passer par un système intermédiaire (*Proxy Agent*). Ce proxy traduit le protocole pour qu'il soit compréhensible par le manager, et réciproquement. Il peut être implémenté sur un élément dédié ou directement dans la station d'administration.

7.4.3 La modélisation d'objets pour l'administration

Notion d'objets

Le principe est le même qu'en programmation orienté objet. Il est caractérisé par :

— un ensemble de données (attributs)
— un ensemble de procédures manipulant ces données

En supervision, les objets sont une représentation abstraite des ressources à administrer.

Les objets à administrer sont définis par :

— Les opérations qui peuvent être effectuées ainsi que leurs effets sur les objets managés et leurs attributs.
— Les notifications émises par l'agent en charge de l'objet managé et contenant l'information sur un évènement.
— Les attributs, qui sont des paramètres caractérisant l'objet.
— Les relations avec les autres objets.

L'ensemble des objets supervisés par un NMS constitue la MIB.

Les classes d'objet (MOC)

La classe décrit un ensemble d'objets partageant la même structure et le même comportement. C'est un modèle pour générer ses représentants physiques, ou instances : Managed Object Instance (MOI).

La définition d'une classe comprend :

- la définition de son nom et identifiant (OID)
- l'héritage des propriétés d'une autre classe modèle
- la liste des *attributs* spécifiques à cette classe
- la ou les relations avec les autres classes (name-binding), défini aussi par un OID, si cette classe doit être instanciée (c'est-à-dire qu'elle ne sert pas seulement de modèle pour définir d'autres classes). Le name-binding sert aussi à définir l'attribut de nommage de la classe.

L'ensemble des classes décrivant les objets supervisés et leur relation entre elles, définie par les *name-binding*, constitue un schéma de MIB.

Notion de MIB

La MIB est l'ensemble des objets (MOI) pris en charge par un NMS. Elle est répartie dans tout le réseau.

La station d'administration (*Manager Station*) doit contenir le schéma de la MIB (*MIB Template*) pour connaître tous les objets que contiennent les différents agents.

Un agent fournit les informations d'un équipement supervisé. Pour cela, il doit avoir une Base de Données contenant la liste des objets manipulables.

Le sous-ensemble des objets manipulables porte le nom de MIBlet. Par abus de langage on parle aussi de MIB pris en charge par un agent.

Chaque ressource gérée correspond à un objet. La MIB est une collection structurée de ces objets.

Pour que des superviseurs différents puissent utiliser la MIB :

- Pour une même ressource, l'objet doit être le même dans le NMS et dans l'équipement administré.
- Des règles d'écriture d'une MIB doivent exister (structure, classification, encodage...).

Ceci est décrit dans un document appelé le Structure of Management Information (SMI). L'ISO utilise la norme Guidelines for the Definition of Managed Objects (GDMO). Dans le monde TCP/IP on utilise les RFC 1155 et 1212.

Arbre d'enregistrement

Pour identifier des objets de manière unique alors qu'il représentent des normes très diverses, l'ISO a constitué une BD structurée en arborescence : l'arbre d'enregistrement

(*registration tree*). Il existe un seul exemplaire de cet arbre au monde. Il représente différents organismes et constructeurs.

Dialogue entre le NMS et l'agent

Peu importe le protocole utilisé, le NMS et l'agent utilisent deux types de requêtes :

1. requêtes en mode question/réponse (request/response)
2. requêtes en mode envoi simple (trap, event report)

Dans le modèle ISO de management, le protocole utilisé est le Common Management Information Service (CMIS)/CMIP. La plupart des requêtes peuvent alors être en mode confirmé ou en mode non confirmé.

7.4.4 La modélisation d'objets pour l'administration SNMP

OID

Contrairement au modèle SMI de l'ISO, le modèle SMI pour SNMP n'utilise ni *classes*, ni *attributs*, ni *namebinding*.

Il utilise essentiellement des *Object-Types* qui sont nommés et immatriculés par un OID. Certains composants du modèle SMI pour SNMP (*groups*, *sequence* et *sequence of*) ne sont pas instanciables : ils sont utilisés comme objets parents.

Contrairement au modèle GDMO, dans le modèle SMI pour SNMP, la relation entre objets découle directement de leur immatriculation par un OID.

À chaque nom de l'arbre est associé par convention une valeur numérique. Les objets sont donc identifiés par les chemins depuis la racine jusqu'à une feuille (comme pour les répertoires et fichiers). L'OID de l'objet `sysDescr` de la MIB2 est par exemple :

`iso.org.dod.internet.mgmt.mibII.system.sysDescr`, ce qui donne :

`1.3.6.1.2.1.1.1`.

L'instance de cet objet, par convention, est constitué de la concaténation de son OID et du suffixe `0` : `1.3.6.1.2.1.1.1.0`.

La MIB II

La MIB standard, ou publique, est communément appelée MIB II. La MIB I, RFC 1156, définissait 114 objets répartis en 8 classes. Elle a été remplacée par la MIB II (RFC 1213) qui définit 170 objets répartis dans 10 groupes normalisés.

Un autre groupe, RMon, a été ajouté sous l'identificateur d'objet 16.

Résumé des objets de la MIB II :

1. **system** — groupe décrivant le système.
2. **interface** — groupe traitant des accès réseau.
3. **at** — groupe traitant des conversions d'adresses IP.
4. **ip** — groupe traitant des évènements IP.

5. **icmp** — groupe traitant des évènements ICMP.
6. **tcp** — groupe traitant des évènements TCP.
7. **udp** — groupe traitant des évènements UDP.
8. **egp** — groupe traitant des évènements EGP.
9. **cmot** — groupe traitant des évènements CMIP over TCP/IP (CMOT).
10. **transmission** — groupe traitant de la qualité de transmission.
11. **snmp** — groupe traitant des évènements SNMP.
12. **rmon** — groupe traitant du trafic observé par un agent RMon.

Les MIB privées

SNMP permet d'étendre la gestion à de nouveaux éléments par l'addition de nouvelles MIB à la MIB II. Beaucoup de constructeurs comme Cisco, Hewlett Packard, Novell, Synoptics, Wellfleet, etc. ont développé leurs propres MIB.

Elles concernent les produits spécifiques et on les appelle les MIB privées.

Pour intégrer une nouvelle MIB sur le NMS, il faut d'abord la récupérer auprès du constructeur. On peut alors télécharger un fichier texte écrit en langage Abstract Syntax Notation One (ASN.1). On doit ensuite les compiler sur la station de supervision.

Les MIB expérimentales

Les MIB expérimentales sont issues de collaborations entre organismes de standardisation, universités et constructeurs. À terme, ils peuvent donner de futures MIB normalisées.

7.4.5 Le protocole SNMP

C'est un protocole applicatif développé par l'IETF. Il se rencontre bien sûr dans l'environnement TCP/IP, mais il était aussi utilisé sur les réseaux Novell IPX/SPX.

Les échanges SNMP

Avec SNMP, le dialogue entre le NMS et l'agent se fait grâce à 5 commandes standards :

1. `GetRequest`, générée par le NMS — Permet au NMS de demander la lecture d'une des valeurs actuelles des variables de la MIB.
2. `GetNextRequest`, générée par le NMS — Permet au NMS de demander la lecture de la valeur suivante dans une table. Elle est donc précédée d'une requête de type `Get`.
3. `SetRequest`, générée par le NMS — Permet au NMS de modifier la valeur d'une des variables dans la MIB d'un composant. Cette variable doit être accessible en Lecture/Écriture. L'agent répond en renvoyant une PDU avec les valeurs modifiées.
4. `GetResponse`, générée par l'agent — Pour chaque PDU reçue en provenance du NMS, l'agent envoie une réponse avec la ou les valeurs demandées.
5. `Trap`, générée par l'agent — Message non sollicité : l'agent informe le NMS d'un évènement survenu dans l'équipement. L'envoi du trap est conditionné au niveau de gravité de l'évènement (par défaut 3, mais modifiable).
   Il existe 7 niveaux de gravité, de 1 (les plus graves) à 7 (sans importance).

Protocoles de transport

Le plus souvent, SNMP est rencontré dans le contexte TCP/IP. Les échanges s'appuient alors sur UDP.

Un agent reçoit ses messages sur le port UDP 161. En termes de service TCP/IP, le manager est le client (qui envoit des requêtes), et l'agent est le serveur (qui écoute sur le port 161).

Un manager reçoit les traps SNMP sur le port UDP 162. En termes de service TCP/IP, l'agent est le client et le trap manager est le serveur, qui écoute sur le port UDP 162.

Si ces messages sont trop fréquents, cela alourdit sensiblement la charge d'un réseau (de 10 à 20%). Ceci explique pourquoi SNMP s'appuie sur UDP et non TCP.

Champs des PDU SNMP

PDU de type requête/réponse

- **PDU type** — Indique la nature de la PDU :
  0. GetRequest
  1. GetNextRequest
  2. GetResponse
  3. SetRequest
  4. Trap
- **Request ID** — Entier qui permet d'identifier les requêtes et les réponses les unes des autres.
- **Error Status** — Indique la nature de l'erreur constatée :
  0. noError
  1. tooBig
  2. noSuchName
  3. badValue
  4. readOnly
  5. genErr
- **Error Index** — Entier qui permet d'indiquer la variable responsable de l'erreur courante.
- **Variable Bindings** —

PDU de type trap

- **PDU type** — Indique la nature de la PDU, ici trap donc 4.
- **Enterprise** — Précise le type d'équipement qui a généré la PDU. C'est la valeur de l'objet `sysObjectID` de la MIB standard.
- **Agent Address** — L'adresse IP de l'agent qui a généré le trap.
- **Generic Trap** — Type de trap.
- **Specicif Trap** — Code du trap. Précise le type de trap développé par son propriétaire (le constructeur).

— **Time Stamp** — Temps écoulé depuis la dernière initialisation de l'équipement. C'est la valeur de l'objet `sysUpTime`.
— **Variable Bindings** — Rassemble le nom et la valeur des objets considérés.

Authentification par communautés

En SNMP v1 ou v2, il y a un mécanisme simple d'authentification entre le manager et l'agent. Ce mécanisme est basé sur un nom de communauté. C'est l'équivalent d'un mot de passe que doit donner le manager pour accéder aux informations de l'agent.

La communauté est associée à des droits d'accès : `RO`, `RW`. Par défaut, la communauté s'appelle `public` est est en `RO`. Pour avoir un droit d'écriture, il faut créer une autre communauté sur l'agent ainsi que sur le manager, et lui accorder les droits `RW`.

Le premier problème de ce système est que le nom de la communauté passe en clair sur le réseau. Il peut donc facilement être récupéré et réutilisé pour modifier la configuration du réseau.

Le deuxième problème est qu'une communauté apporte une visibilité "globale" de la MIB de l'agent. Quiconque fait donc une requête dans une communauté autorisée peut accéder à la totalité des objets de l'agent.

SNMPv2

La v2 de SNMP a pour objectifs d'améliorer le protocole au niveau :

— de la sécurité
— du trafic généré par le protocole lui-même
— de la souplesse dans la gestion des managers sur le réseau

Sécurité

L'authentification permet de s'assurer qu'un message transmis n'a pas été modifé pendant son trajet sur le réseau. Elle utilise la méthode `Digest Authentication Protocol` qui repose sur l'algorithme Message-Digest Algorithm (MD5).

Le champ `Context` permet une identification plus fine que la communauté en v1 parce qu'il porte sur une partie de la MIB, non pas sur la MIB entière. L'agent contient une petite Base de Données des parties qu'il connaît. À chaque demande sur l'agent, il consulte d'abord son ACL pour déterminer si l'opération est permise ou non.

Diminution du trafic

Un nouveau type de requête permet de diminuer le trafic réseau quand on veut rappatrier de grandes quantités de données d'un seul coup : `GetBulkRequest`. Auparavant, il fallait envoyer plusieurs requêtes `GetRequest` pour obtenir des blocs de données importants.

Dialogue manager à manager

La possibilité de dialoguer entre stations d'administration est très importante puisqu'elle permet de distribuer l'administration d'un réseau entre plusieurs managers. Le réseau est segmenté, ce qui limite le trafic au trafic lié à son administration.

Pour cela, il y a une nouvelle PDU : `InformRequest` et une nouvelle MIB, la MIB `Manager to Manager` (RFC 1451).

Coexistance SNMP v1 et v2

La version 1 de SNMP utilise 5 types de requêtes et la version 2 en utilise 6, dont seulement 4 sont compatibles avec la version 1.

— `GetBulkRequest` est une nouveauté de la version 2 et ne pourra donc pas fonctionner en version 1.
— Le trap nécessite une traduction parce que le message a changé de format.

Pour continuer à utiliser les 2 versions ensemble, deux solutions ont été mises en place :

1. Passer par un `proxy-agent` qui traduit les requêtes.
2. Utiliser un manager compatible avec les deux versions. Ce manager utilise des messages en v1 pour les agents en v1 et des messages en v2 pour les agents en v2.

Les deux versions peuvent donc cohabiter, mais ne sont pas compatibles. La version 2 n'est pas un standard et ne le sera probablement jamais puisqu'une nouvelle version a vu le jour.

SNMPv3

La version 3 est proposée en janvier 1998 par les RFC 2262 et 2263. Elle a eu pour but de faire converger différents travaux existants de SNMPv2. Il fallait aussi respecter les impératifs de sécurité et rester le plus simple possible. La version 3 fait aussi l'effort de prendre en compte la migration depuis l'un des deux protocoles antérieurs.

7.4.6 Les sondes RMon

Les sondes RMon, ou les agents RMon, sont des outils de suiveillance de trafic qui livrent une synthèse du réseau. L'idée est de prévenir d'une faille ou d'une dégradation de performance, et d'agir avant la panne.

Fonctionnement

Contrairement au système SNMP, où la station d'administration interroge régulièrement (*poll*) les agents répartis sur le réseau, les sondes RMon évitent une surcharge de trafic réseau. Pour cela, elles utilisent un échange plus intelligent : ce serait dommage qu'un outil de gestion de charge réseau génère de la charge réseau. Grâce à un processeur qui exécute les traitements de données, une sonde RMon est autonome et ne transmet que les informations réellement nécessaires.

Avantages

Pouvoir placer des sondes sur des réseaux distants et rapatrier les informations déjà filtrées à travers des liens inter-réseau permet de faciliter la présentation statistique de l'état d'un réseau. Le NMS n'a pas besoin de faire de *poll* (multiples `GetRequest`) puis d'en faire un traitement statistique local puisque c'est déjà fait en amont. Ainsi, la seule chose qui transite sur le réseau, c'est le résultat déjà traité de ce que la sonde RMon a recueilli.

La MIB RMon Ethernet

RMon est défini par les RFC 1271 et 1757 pour les réseaux Ethernet et la RFC 1513 pour les réseaux Token Ring. Il est rangé au point `16` du schéma d'enregistrement.

On y trouve :

— **Statistiques** (statistics) — Délivre des renseignements sur l'activité du réseau local incluant le volume des trames, le trafic broadcast, multicast, la répartition des trames par taille et les erreurs de transmission.
— **Historique** (history) — Extraits de statistiques précédentes recueillis à intervalles réguliers puis enregistrés pour un traitement ultérieur.
— **Alarmes** (alarms) — Clignotants associés aux variables MIB pour notifier d'un évènement.
— **Serveurs** (hosts) — Table d'informations associées à chaque serveur de même nature que celle du groupe *statistiques*. Pour déclarer un nouveau serveur, il suffit de lire une adresse MAC.
— **Serveurs Top N** (hosts Top N) — Similaire au groupe *serveurs* mais restreints à N unités sélectionnées arbitrairement en fonction de certains critères. On peut par exemple retenir les 3 nœuds engendrant le plus for taux d'erreurs.
— **Matrice de trafic** (traffic matrix) — Conserve une trace sur le volume de trafic échangé ou sur les erreurs de transmission.
— **Filtre** (filter) — Isole un type de trafic.
— **Capture** (packet capture) — Copie des trames répondant au format de filtrage précédent. Ensuite on analyse ces trames à travers un décodeur.
— **Évènements** (events) — Contrôle l'envoi des messages SNMP. La sonde peut signaler différents évènements comme un dépassement de seuil ou une capture de trames.

RMon 2

RMon 2 complète la version 1 en surveillant les couches supérieures du modèle Open Systems Interconnection (OSI). Ainsi RMon 2 ne remplace pas RMon 1 mais le complète.

RMon 2 apporte :

— Les renseignements statistiques recueillis sont détaillés par adresses réseaux et types d'application.
— Les statistiques associées au volume de trafic sont agrégés par type de protocole.
— Tous les protocoles interprétés par la sonde sont enregistrés dans un répertoire pour faciliter leur identification (en séparant les vendeurs).
— La correspondance entre adresses réseaux et adresses MAC facilite l'identification des utilisateurs et la détection des doublons d'adresses IP.

RMon 1 ne s'occupe que des couches OSI 1 et 2 (physique et liaison).

Chapitre 8

Cisco

8.1 Démarrage d'un switch

Un switch a une séquence de boot en 5 étapes.

1. le programme Power-On Self-Test (POST), enregistré en ROM, vérifie le sous-système CPU, le CPU, le DRAM, le système de fichiers flash
2. charge le logiciel de démarrage, enregistré en ROM
3. le boot loader effectue une initialisation CPU de bas niveau, initialise les registres CPU (où la mémoire physique est mappée, ainsi que sa quantité et vitesse)
4. le boot loader initialise le système de fichiers flash
5. le boot loader trouve et charge le système d'exploitation IOS en mémoire et lui donne le contrôle

L'IOS initialise les interfaces en utilisant le fichier startup-config qui s'appelle `config.text` et est situé en flash.

Pour voir à quoi correspond le fichier boot de l'IOS :

```
Switch# show boot
```

Définir la variable d'environnement BOOT :

```
Switch(config)# boot system flash:/c2960-lanbasek9-mz.150-2.SE/c2960-
    lanbasek9-mz.150-2.SE.bin
```

boot system	commande principale
flash :	lieu d'enregistrement
c2960-lanbasek9-mz.150-2.SE/	chemin vers le système de fichiers
c2960-lanbasek9-mz.150-2.SE.bin	nom de fichier de l'IOS

8.2 LED de switch

— SYST — System
  — éteint : pas de jus

  - vert : ok
  - orange : pas ok
- RPS — Redundant Power System
  - éteint : RPS éteint ou mal branché
  - vert : prêt à fournir l'alimentation de secours
  - vert clignotant : non accessible parce qu'aliment un autre périphérique
  - orange : mode standby ou mauvaise condition
  - orange clignotant : l'alimentation interne ne marche plus et le RPS fournit l'alimentation
- STAT — Port Status

  vert : le mode de port status est sélectionné, valeur par défaut
  - les LED des ports affichent des couleurs qui veulent dire différentes choses
  - éteint : pas de lien ou `administratevely down`
  - vert : lien présent
  - clignotant : actif, envoie ou reçoit des données
  - alterne entre vert et orange : problème de lien
  - orange : port bloqué pour s'assurer qu'aucune boucle n'existe dans le domaine de transfert (premières 30 secondes après activation)
  - orange clignotant : port bloqué pour empêcher les boucles possibles du domaine de transfert
- DUPLX — Port Duplex

  vert : le mode de port duplex est sélectionné
  - les LED éteintes sont en mode half-duplex
  - vert : port en mode full-duplex
- SPEED — Port Speed

  vert : le mode port speed est sélectionné, les LED des ports vont afficher des couleurs qui veulent dire différentes choses
  - éteint : 10 Mbps
  - vert : 100 Mbps
  - vert clignotant : 1000 Mbps
- PoE — Power over Ethernet

  présent si le PoE est supporté
  - éteint : PoE non sélectionné, pas de port n'a été privé d'alimentation ou placé en condition de faute
  - orange clignotant : non sélectionné mais au moins un port a été privé d'alimentation ou placé en condition de faute
  - vert : sélectionné, les LED des ports vont afficher des couleurs qui veulent dire différentes choses
    - éteint : PoE éteint
    - vert : PoE allumé
    - alterne entre vert et orange : le PoE est refusé parce qu'il dépasserait les capacités du switch
    - orange clignotant : PoE éteint à cause d'un problème
    - orange : désactivé pour ce port

8.3 Commandes

Les valeurs entre chevrons (< >) sont des variables qu'il faut adapter. Les valeurs entre crochets ([ ]) sont optionnelles.

8.3.1 Navigation

Entrer dans le mode d'exécution privilégié (privileged EXEC mode)

```
Switch> enable
```

Retourner vers le mode d'exécution utilisateur (user EXEC mode)

```
Switch# disable
```

Entrer dans le mode de configuration global

```
Switch# configure terminal
```

Entrer dans le mode de la ligne console

```
Switch(config)# line console 0
```

Entrer dans le mode de la ligne vty

```
Switch(config)# line vty 0 15
```

Retourner au mode parent

```
Switch(config-line)# exit
```

Retourner directement au mode d'exécution privilégié

```
Switch(config-line)# end
```

8.3.2 Configuration basique

Donner un nom à un périphérique

```
Switch(config)# hostname <name>
```

Sécuriser l'accès au mode d'exécution privilégié

```
Switch(config)# enable secret my-password
```

Sécuriser l'accès au mode d'exécution utilisateur

```
Switch(config)# line console 0
Switch(config-line)# password <password>
Switch(config-line)# login
Switch(config-line)# exit
Switch(config)# line vty 0 15
Switch(config-line)# password <password>
Switch(config-line)# login
Switch(config-line)# exit
```

Chiffrer tous les mots de passe dans les fichiers de configuration

```
Switch(config)# service password-encryption
```

Configurer la bannière pour le Message Of The Day (MOTD) légal

```
Switch(config)# banner motd $ MessageOfTheDay $
```

Afficher la configuration active

```
Switch# show running-config
```

Afficher la configuration de démarrage

```
Switch# show startup-config
```

Il y a deux fichiers de configuration :

1. startup-config : enregistrée dans la NVRAM pour le démarrage et le redémarrage, ne perd pas son contenu à la mise hors tension
2. running-config : enregistrée dans la RAM pour la configuration courante, effective immédiatement mais perdue à la mise hors tension

Sauvegarder les modifications de la configuration

```
Switch# copy running-config startup-config
```

Remettre le périphérique dans l'état de sa configuration précédente en redémarrant

```
Switch# reload
```

. . . ou en sauvegardant la `startup-config` vers la `running-config`

```
Switch# copy start run
```

Effacer toute configuration (supprime les modifications sauvegardées mais non désirées)

```
Switch# erase startup-config
```

. . . puis redémarrer le périphérique (ceci le remet dans les paramètres d'usine).

Augmenter la taille de l'historique du terminal à 200 lignes (par défaut = 10)

```
R1# terminal history size 200
```

8.3.3 Mise en réseau

Pour être joignable, une interface doit :

— être configurée avec au moins une adresse IP

```
R1(config-if)# ip address
R1(config-if)# ip address <ipv4 address> <mask>
    et/ou
R1(config-if)# ipv6 address <adresse ipv6>/<prefix length>
```

Noter que pour l'IPv6, il n'y a pas d'espace entre l'adresse et la longueur de préfixe. Pour l'adresse link-local sur le routeur :

```
R1(config-if)# ipv6 address fe80::1 link-local
```

— être activée

```
R1(config-if)# no shutdown
R1(config-if)# exit
```

— avoir une description
Jusqu'à 240 caractères, optionnel mais une bonne pratique, aide à diagnostiquer les problèmes et identifier les information de contact tièrces.

```
R1(config-if)# description Link to LAN 1
```

D'autres commandes utiles :

Entrer le mode `range` pour par exemple mettre toutes les 16 interfaces d'un switch à `no shutdown` :

```
Switch(config)# interface range FastEthernet 0/0 - 15
Switch(config-if-range)# no shutdown
```

Vérifier les états des interfaces :

```
Switch# show ip interface brief
```

Ajouter des routes :

— En IPv4 :

```
R1(config)# ip route <dest> <mask> <next hop> [<distance>]
```

— Le `mask` peut être modifié pour faire un sommaire d'un groupe de réseaux (sur-réseau).
— Le `next hop` peut être soit une interface de sortie, soit une adresse IP, soit les deux.
— La `distance` correspond à une valeur de distance administrative (AD) entre 1 et 255. Cette valeur peut être plus élevée que celle d'un protocole dynamique pour créer une route flottante. Si rien n'est indiqué, la valeur de `distance` par défaut est 1.

— En IPv6 :
Le routage IPv6 n'est pas activé par défaut. Pour activer l'IPv6 sur un routeur :

```
R1(config)# ipv6 unicast-routing
```

La commande pour l'IPv6 est presque la même qu'en IPv4 sauf qu'on remplace `ip route` par `ipv6 route` et qu'il n'y a pas d'espace entre le préfixe et sa longueur.

```
R1(config)# ipv6 route <prefix>/<length> <next hop> [<distance>]
```

... une route par défaut :

```
R1(config)# ip route 0.0.0.0 0.0.0.0 172.26.200.200
R1(config)# ipv6 route ::/0 2001:db8:acad::1
```

Afficher la table de routage d'un routeur :

```
R1# show ip[v6] route
```

Ajouter une passerelle par défaut sur un switch :

```
Switch(config)# ip default-gateway <ip address>
```

Configurer un switch pour le routage inter-VLAN `router-on-a-stick` :

Voir 5.5.9

Activer à assigner une adresse de rebouclage sur un routeur :

```
R1(config)# interface loopback <nombre>
R1(config-if)# ip address 10.0.0.1 255.255.255.0
R1(config-if)# exit
```

8.3.4 Accès à distance

Pour un accès à distance, un switch doit avoir un SVI configuré en IPv4 ou IPv6. Le SVI doit être confiuré avec une passerelle par défaut.

Par défaut, l'administration est controlée à travers le VLAN1, qui comprend tous les ports. Par sécurité, il vaut mieux utiliser un VLAN autre que VLAN1 (comme VLAN99).

Mise en place du SVI

1. **Configurer l'interface d'administration**
   - Entrer dans le mode de configuration global :

     ```
     S1# configure terminal
     ```

   - Entrer dans le mode de configuration d'interface pour le SVI :

     ```
     S1(config)# interface vlan 99
     ```

   - Configurer l'adresse IPv4 pour l'interface d'administration :

     ```
     S1(config-if)# ip address 172.19.99.11 255.255.255.0
     ```

   - Configurer l'adresse IPv6 pour l'interface d'administration :

     ```
     S1(config-if)# ipv6 address 2001:db8:acad:99::11/64
     ```

   - Activer l'interface d'administration :

     ```
     S1(config-if)# no shutdown
     ```

   - Retourner au mode d'exécution privilégié :

     ```
     S1(config-if)# end
     ```

   - Sauvegarder :

     ```
     S1# copy running-config startup-config
     ```

2. **Configurer la passerelle par défaut**
   Ceci est seulement nécessaire si le switch est administré depuis des réseaux qui ne sont pas directement connectés. La passerelle par défaut IPv6 n'est pas nécessaire (le switch la recevra par un message RA).
   - Entrer dans le mode de configuration global :

     ```
     S1# configure terminal
     ```

— Configurer la passerelle par défaut :

```
S1(config)# ip default-gateway 172.17.99.1
```

— Retourner au mode d'exécution privilégié :

```
S1(config)# end
```

— Sauvegarder :

```
S1# copy running-config startup-config
```

3. **Vérifier la configuration**
   L'adresse IP appliquée au SVI n'est que valable pour l'accès à distance. Il ne permet pas au switch de router des paquets de couche 3.

```
S1# show ip interface brief
S1# show ipv6 interface brief
```

Mise en place de SSH

— Telnet : Utilise le port 23, non sécurisé.
— SSH : Utilise le port 22, chiffré. Méthode privilégiée par rapport à Telnet.
— Vérifier que le switch supporte SSH
Si le nom du fichier de l'IOS inclut `k9`, le périphérique supporte la cryptographie.

```
S1# show version
```

— Configurer SSH
Avant de commencer, le switch doit être configuré avec un nom d'hôte unique et une connectivité réseau (voir 8.3.3).

1. Vérifier le support SSH :

```
S1# show ip ssh
```

2. Configurer le domaine IP :

```
S1(config)# ip domain-name example.com
```

3. Générer une paire de clés RSA :
   — SSH version 1 a des défauts de sécurité, activer la version 2 :

```
S1(config)# ip ssh version 2
```

   — Générer des clés RSA, ce qui va aussi activer SSH :

```
S1(config)# crypto key generate rsa
```

   — Supprimer la paire de clés RSA, ce qui désactive aussi SSH :

```
S1(config)# crypto key zeroize rsa
```

4. Configurer l'authentification de l'utilisateur :

```
S1(config)# username <user> secret <password>
```

5. Configurer les lignes vty :
   Activer le protocole SSH sur les lignes vty. Ceci prévient également les connection non SSH (comme Telnet).

```
S1(config)# line vty 0 15
S1(config-line)# transport input ssh
S1(config-line)# login local
S1(config-line)# exit
```

8.3.5 Temps

Afficher l'heure courante :

```
Switch# show clock
```

Ajouter des serveurs de temps :

```
Switch(config)# clock timezone CEST 1
Switch(config)# clock summer-time CEST recurring last Sun Mar 2:00 last Sun
    Oct 3:00
Switch(config)# ip domain-lookup
Switch(config)# ip name-server 194.2.0.20
Switch(config)# ntp server time.nist.gov source FastEthernet0/0
```

. . . puis attendre une minute ou deux

8.3.6 Filtrer la sortie

— `section` — entièrement afficher la section commençant par l'expression du filtre

```
R1# show running-config | section line vty
```

— `include` — inclure toutes les lignes qui correspondent à l'expression du filtre

```
R1# show ip interface brief | include up
```

— `exclude` — exclure toute les lignes qui correspondent à l'expression du filtre

```
R1# show ip interface brief | exclude unassigned
```

— `begin` — afficher toutes les lignes à partir d'un certain point, qui commence à la ligne correspondant à l'expression du filtre

```
R1# show ip route | begin Gateway
```

8.3.7 Auto-MDIX

Retire le besoin d'utiliser un câble droit ou croisé. Quand la fonction Auto-MDIX est activée, le switch détecte automatiquement le type de câble connecté et configure les interfaces en conséquence.

— sans Auto-MDIX :
  — câble droit : vers les serveurs, terminaux ou routeurs
  — câble croisé : vers d'autres switches ou hubs
— avec Auto-MDIX :
  — les deux câbles peuvent être utilisés
  — le duplex et le débit doivent être configurés sur `auto`

```
S1(config-if)# mdix auto
```

Auto-MDIX et activé par défaut sur les switchs plus récents (Catalyst 2960, Catalyst 3560), mais n'est pas disponible sur des switches plus anciens (Catalyst 2950, Catalyst 3550).

Examiner le réglage d'Auto-MDIX pour une interface spécifique :

```
S1# show controllers ethernet-controller f0/1 phy | include MDIX
```

— *phy* : mot clé
— *include* : filtrer pour limiter la sortie aux lignes référençant Auto-MDIX

8.3.8 Duplex

Les ports d'un switch peuvent être manuellement configurés avec des paramètres spécifiques de duplex et de débit.

Les débits de 10 ou 100 Mbps peuvent être soit en half-duplex ou en full-duplex. Les débits à 1000 Mbps (1 Gbps) ne peuvent qu'être en full-duplex.

Entrer dans le mode de configuration global :

```
S1# configure terminal
```

Entrer dans le mode de configuration d'interface :

```
S1(config)# interface FastEthernet 0/1
```

Configurer le duplex d'une interface :

```
S1(config-if)# duplex full
```

Configurer le débit d'une interface :

```
S1(config-if)# speed 100
```

Retourner au mode d'exécution privilégié :

```
S1(config-if)# end
```

Sauvegarder :

```
S1# copy running-config startup-config
```

L'autonégotiation est pratique quand les paramètres sont inconnus ou peuvent changer, mais si on connecte des périphériques connus, il vaut mieux manuellement renseigner le débit et le duplex.

Erreur dans l'autonégotiation → paramètres de duplex et de débit non correspondants → problèmes de connectivité.

La fibre optique aura toujours un débit prédéterminé et en full-duplex.

8.3.9 Vérification

Sommaire de commandes de vérification utiles sur un switch

— Afficher les états et statistiques :

```
S1# show interfaces f0/18
```

— Afficher la startup-config courante :

```
S1# show startup-config
```

— Afficher la running-config courante :

```
S1# show running-config
```

— Afficher les commandes appliquées à <interface> :

```
R1# show running-config interface <interface>
```

— Afficher des informations sur le système de fichiers flash :

```
S1# show flash
```

— Afficher l'état matériel et logiciel du système :

```
S1# show version
```

— Afficher l'historique des commandes entrées :

```
S1# show history
```

— Afficher les informations IP pour une interface :

```
S1# show ip[v6] interface <interface-id>
```

— Afficher un sommaire pour toutes les interfaces :

```
S1# show ip[v6] interface brief
```

— Afficher le contenu de la table de routage :

```
R1# show ip[v6] route [static] [<ip address>]
```

— Afficher la table MAC :

```
S1# show mac-address-table
```

ou

```
S1# show mac address-table
```

— Faire un ping IPv4 ou IPv6 :

```
R1# ping <ipv4 or ipv6 address>
```

Types d'erreurs :

— *Input Errors :* Nombre total d'erreurs, ce qui inclue les runts, les baby giants, pas de cache, CRC, trame, overrun, et les valeurs ignorées.
— *Runts :* Trames rejetées parce que plus petites que la taille minimum d'une trame (64 octets). Les Network Interface Card (NIC) qui fonctionnent mal sont la cause la plus commune de trames runt excessives, mais elles peuvent aussi être causées par des collisions.
— *Giants :* Trames rejetées parce que plus grandes que la taille maximum d'une trame (1518 octets).

— *CRC :* Généré quand la valeur CRC calculée n'est pas la même que la valeur reçue dans le FCS. En général cela veut dire un problème de câble.
— *Output Errors :* Somme de toutes les erreurs qui ont empêché la transmission finale de trames en sortie de l'interface.
— *Collisions :* Nombre de messages retransmises à cause d'une collision ethernet. Ces collisions sont normales en half-duplex mais ne devraient jamais apparaître en full-duplex.
— *Late Collisions :* Collision qui apparaît après que 512 bits de la trame ont été transmises. Une longueur de câble excessive est une cause usuelle, ou bien une mauvaise configuration de duplex. Des réseaux proprement architecturés et configurés ne devraient jamais avoir de `late collision`.

8.3.10 Récupérer d'un crash

Si l'IOS ne peut pas être utilisé à cause de fichier manquants ou corrompus, le boot loader a une interface en ligne de commande qui donne accès aux fichier en flash.

Pour accéder au boot loader :

1. se connecter avec un câble console
2. débrancher le câble d'alimentation du switch
3. rebrancher le câble d'alimentation et sous 15 secondes appuyer et maintenir enfoncé le bouton `Mode` pendant que la LED `SYST` clignote en vert
4. continuer d'appuyer sur `Mode` jusqu'à ce que la LED `SYST` devienne brièvement orange puis vert, puis relâcher le bouton `Mode`
5. l'invite de commandes `switch:` apparaît dans le terminal du PC

Visualiser le chemin de la variable d'environnement `BOOT` :

```
switch: set
```

Initialiser le système de fichiers flash pour visualiser les fichiers courants dans le flash :

```
switch: flash_init
```

Visualiser les dossiers et les fichiers du flash :

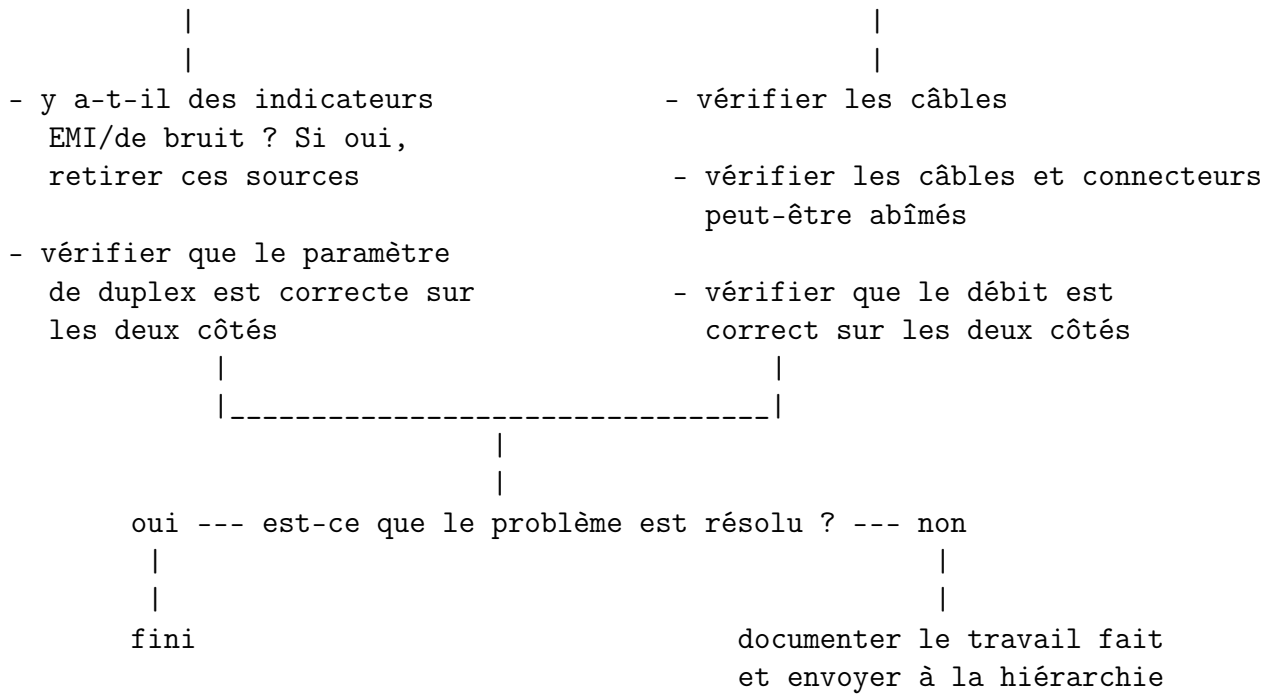
```
switch: dir flash:
```

Changer la variable d'environnement BOOT :

```
switch: BOOT=flash:c2960-lanbasek9-mz.150-2.SE8.bin
switch: set # check new variable
switch: boot
```

8.3.11 Résoudre les erreurs réseau

```
                S1# show interfaces
                          |
                          |
oui --- est-ce que l'interface est up --- non
```



Certaines commandes utiles pour diagnostiquer des problèmes de réseau :

```
R1# ping
R1# traceroute
R1# show ip route
R1# show ipv6 route
R1# show ip interface brief
R1# show cdp neighbors
```

La commande `show cdp neighbors` indique les périphériques Cisco directement connectés. Elle vérifie ainsi le fonctionnement de la connectivé des couches 1 et 2.

Deuxième partie

Système

Chapitre 1

LDAP

1.1 Présentation de LDAP

Dans une entreprise, les applications et serveurs on besoin de données pour l'authentification, les droits d'accès, etc. Tenir ces informations à jour, y accéder, les modifier peut vite devenir compliqué, voire ingérable.

Les annuaires LDAP proposent de centraliser les informations et d'y connecter des applications clientes par le biais d'un protocole standardisé.

LDAP est né pour simplifier la mise en œuvre de Directory Access Protocol (DAP), qui était utilisé avec les annuaires X.500.

LDAP est donc une version allégée de DAP et propose les évolutions suivantes :

- utilisation de l'encodage UTF-8
- authentification via Simple Authentication and Security Layer (SASL) et Transport Layer Security (TLS)
- support des *Referrals* (branche qui pointe vers un autre annuaire)
- support d'Unicode (internationalisation)
- capacité d'étendre le protocole
- support des schémas dans l'annuaire

Un annuaire d'entreprise ressemble un peu à un annuaire téléphonique, sauf qu'il gère plus de choses.

Dans un Système d'information (SI), on peut trouver d'autres annuaires :

- DNS
- Network Information Services (NIS)
- Whois (base d'informations concernant les noms de domaine)

Pourquoi ne pas utiliser une base de données ?

- On lit plus souvent un annuaire qu'on ne le met à jour. Les bases de données contiennent des informations en mouvement.
- Un annuaire fournit une méthode de consultation standardisée.
- La norme LDAP définit le modèle de données, alors qu'une base de données varie.

1.1.1 Principes

Les données sont centralisées dans un système de stockage, peu importe si c'est dans un fichier ou une base de données ou autre. Un serveur LDAP agit en tant qu'intermédiaire entre cette source de données et un client. Comme le client n'interagit pas avec les données directement mais avec le serveur LDAP, il n'a pas besoin de connaître le stockage des données. Cela permet aussi d'avoir plusieurs serveurs LDAP avec un même système de stockage.

La RFC 2251 qui définit LDAP sépare l'annuaire en 2 composants : modèle de données et modèle de protocole. Mais on peut en définir 4 :

1. Le **modèle de nommage** définit comment l'information est stockée et organisée.
2. Le **modèle fonctionnel** définit les services fournis par l'annuaire (recherche, ajout...)
3. Le **modèle d'information** définit le type d'informations stockées.
4. Le **modèle de sécurité** définit les droits d'accès aux ressources.

1.1.2 Modèle de nommage

Le modèle de nommage désigne la manière dont sont organisées les données dans l'annuaire.

L'annuaire est un arbre d'entrées : il y a donc une représentation hiérarchique des données. Toutes les informations descendent d'une même *racine*.

Cette arborescence est liée au nommage de chaque élément. Pour signifier qu'un élément appartient à l'élément supérieur, il reprend son nom, qu'il complète par le sien.

Exemple : l'arbre `cn=ventes,ou=groups,dc=afpa,dc=fr` nous donne `ventes > groups > afpa.fr`. La racine est composée du nom de domaine où est hébergé le serveur LDAP. Ce nom de domaine est décomposé en Domain Component (DC). Ensuite l'arbre se découpe deux Organisational Unit (OU) qui constituent des branches. Ici on a la branche `groups`, dans laquelle on trouve les feuilles de l'arbre, les `ventes`.

Le fait d'associer les DC aux parties du nom de domaine à la racine de l'annuaire LDAP est une convention.

Quelques notions à connaître :

— **Entrée** (*entry*) — Une entrée peut être une branche (*node*) ou un élément terminal (*leaf*).
— **Distinguished Name (DN)** — C'est le nom complet de l'élément. Il permet de le positionner dans l'arborescence, et donc il est unique dans l'annuaire.
— **Relative Distinguished Name (RDN)** — C'est la partie du DN qui est relative au DN supérieur.
— **Attribut** — Une entrée est composée d'un ensemble d'attributs. Un attribut possède un nom, un type et une ou plusieurs valeurs.

La RFC 2253 normalise l'écriture des DN. Il ne faut pas ajouter d'espace autour du signe "=", ni à la fin du DN.

1.1.3 Modèle fonctionnel

Le modèle fonctionnel désigne la manière dont on accède à l'annuaire, c'est-à-dire le protocole LDAP lui-même.

La base

DN à partir duquel une recherche est faite. `dc=afpa,dc=fr` effectue une recherche sur tout l'arbre, puisque c'est la racine.

La portée (scope)

Nombre de niveaux sur lesquels l'action est effectuée. Il y a 3 niveaux différents :

1. **sub** — L'action est effectuée récursivement à partir de la base spécifiée.
2. **one** — L'action est effectuée sur les fils directs, c'est-à-dire un seul niveau inférieur par rapport à la base spécifiée.
3. **base** — L'action est effectuée uniquement sur la base spécifiée.

Les filtres

Un filtre donne des critères de recherche. Il se compose d'opérations combinées avec les opérateurs booléens classiques : `ET`, `OU` et `NON`. La syntaxe est la suivante : `attribut OPERATEUR valeur`. Les opérateurs :

Égalité	`:=`
Aproximation	`~=`
Supérieur ou égal	`>=`
Inférieur ou égal	`<=`

Un test d'inforiorité stricte : `($(X>=Y)(!(X=Y)))`.

On peut également utiliser '*' en tant que valeur pour faire des recherches sur des parties de chaînes. Seul, ce caractère permet de tester la présence d'un attribut. Dans ce cas il faut que l'opérateur soit '='.

Une opération doit obligatoirement se trouver entre parenthèses. Pour agréguer des tests complexes, on utilise les opérateurs suivants :

Intersection (et)	`:`
Union (ou)	`|`
Négation (non)	`!`

— Combiner plusieurs éléments :
`(&(objectClass=person)(|(givenName=Jean-Christian)(mail=jranu*)))`
— Toutes les personnes ayant leur numéro de téléphone renseigné dans la base :
`&(&(objectClass=person)(telephoneNumber=*))`
— Toutes les personnes dont le nom commence par 'A' et n'habitant pas à Paris :
`&(&(objectClass=person)(cn=A*)(!(l=Paris)))`
— Toutes les personnes dont le nom ressemble à Febvre (Faibre, Fèvre, Lefebvre, ...) :
`&(&(objectClass=person)(cn~=febvre))`
`(&(objectClass=person)(cn=*f*vre))`

Les filtres étendus

Avec les filtres étendus, il est possible de considérer les éléments du DN comme faisant partie de l'entrée elle-même lors de la recherche.

`attribut:dn:=valeur`

Le filtre `(ou:dn:=users)` récupèrera toutes les entrées qui ont un attribut qui a pour valeur `users` et les entrées dont le DN contient un attribut avec la valeur `users`.

Autre exemple d'utilisation des filtres étendus : si un attribut a une règle de comparaison par défaut qui est insensible à la casse, mais qu'on veut faire une recherche avec une valeur précise qui tienne compte de la casse, il faut modifier la règle de comparaison par défaut. Dans ce cas on peut faire `attributid-matching-rule:=value`.

Les opérations

À chaque requête, le client donne un identifiant (Message ID). Le serveur répond avec le même identifiant, en incluant un code indiquant l'état (succès, échec, ...). La réponse du serveur inclut également les données éventuelles qui résultent d'une recherche et un code ID.

Pour interagir avec les données, il y a les fonctions suivantes :

— **Abandon** — Abandonne l'opération précédemment envoyée au serveur.
— **Add** — Ajoute une entrée au répertoire.
— **Bind** — Initie une nouvelle session sur le serveur LDAP.
— **Compare** — Compare les entrées d'un répertoire selon des critères.
— **Delete** — Supprime une entrée d'un répertoire.
— **Extended** — Effectue des opérations étendues.
— **Modify** — Modifie une entrée.
— **Modify DN** — Déplace ou renomme une entrée.
— **Rename** — Modifie le nom d'une entrée.
— **Search** — Recherche des entrées d'un répertoire.
— **Unbind** — Termine une session sur le serveur LDAP.

Bind

L'opération bind authentifie le client au sein du serveur. Il vérifie le mot de passe en le comparant avec l'attribut userPassword de l'entrée correspondante. La valeur de l'attribut contenant le mot de passe commence avec entre accolades le nom de l'algorithme utilisé pour coder le mot de passe (`userPassword:{md5}aGZh5...`).

Le bind anonyme (sans fournir d'identifiant ni de mot de passe) permet de faire une connexion dans un état anonyme. En fonction des droits en place, le client ne pourra pas faire certaines opérations.

Le SASL bind permet de s'authentifier d'autres manières : Kerberos, certificat client, ... Le simple bind envoie le DN de l'utilisateur et son mot de passe en clair. Il faut donc que la connexion soit sécurisée (startTLS).

Search and Compare

Search est utilisé pour faire une recherche et rapatrier des entrées.

- **baseObject** — Le DN de l'entrée à partir de laquelle effectuer la recherche.
- **scope** — La portée, voir 1.1.3.
- **filter** — Le filtre, voir 1.1.3.
- **derefAliases** — Indique si la recherche doit suivre les alias dans les entrées. Un alias est une entrée qui fait référence à d'autres entrées.
- **attributes** — Liste des attributs à ramener à l'issue de la recherche.
- **sizeLimit** — Limitation du nombre d'entrées ramenées à l'issue de la recherche.
- **timeLimit** — Limitation du délai de recherche, en secondes.
- **typesOnly** — Ne renvoie que les types d'attribut et non les valeurs.

Compare prend en argument le DN, un nom d'attribut et une valeur d'attribut. Ensuite il vérifie si l'entrée correspondante contient bien un attribut ayant cette valeur.

Il n'y a pas d'opération *Read*. Pour cela on utilise *Search* avec les paramètres `baseObject` = le DN recherché et `scope` = `base`.

Mise à jour : Add, Delete, Modify

- **Add** — La modification de DN prend en argument le RDN de l'entrée et le DN du nouveau parent, ainsi qu'un indicateur de suppression de l'ancien RDN.

  ```
  dn: <distinguished name>
  changetype: add
  objectclass: top
  objectclass: <objectclassvalue>
  <attrdesc>: <attrvalue>
  <attrdesc>: <attrvalue>
  ```

  Certaines commandes d'administration permettent d'insérer dans un annuaire des objets qui sont décrits dans un fichier, sans qu'il soit nécessaire que ce fichier contienne la commande d'insertion.
- **Delete** — L'object ne peut être effacé que s'il n'a pas de descendents.

  ```
  dn: <distinguished name>
  changetype: delete
  ```

- **Ajout de valeurs à un attribut** — On peut donner à l'attribut autant de valeurs que l'on souhaite. Les attributs précédents de l'objet ne sont pas effacés.

  ```
  dn: <distinguished name>
  changetype: modify
  add: <attribut>
  <attribut>: <attrvalue>
  <attribut>: <attrvalue2>
  ```

- **Suppression de valeurs à un attribut** — Si l'on souhaite effacer uniquement certaines valeurs, il faut les passer en paramètre.

  ```
  dn: <distinguished name>
  changetype: modify
  delete: <attribut>
  <attribut>: <attrvalue>
  <attribut>: <attrvalue2>
  ```

Si l'on souhaite effacer toutes les valeurs d'un attribut d'un objet, il ne faut pas passer de valeur d'attribut en paramètre.

```
dn: <distinguished name>
changetype: modify
delete: <attribut>
```

— **Remplacer les valeurs d'un attribut** — Similaire aux deux cas précédents, les paramètres contiennent cette fois les valeurs qui remplacent les valeurs précédentes.

```
dn: <distinguished name>
changetype: modify
replace: <attribut>
<attribut>: <nouvelle valeur 1>
<attribut>: <nouvelle valeur 2>
```

— **Modification du DN et/ou du RDN** — Modifier le DN d'une entrée veut dire modifier sa position dans l'arbre. Modifier son RDN veut dire modifier son identifiant. Les syntaxes sont similaires.

```
dn: <distinguished name>
changetype: moddn
newrdn: <nouveau RDN>
deleteoldrdn: <1 ou 0>
newsuperior: <nouveau parent>
```

Le protocole LDAP n'a pas de transaction, donc deux clients peuvent modifier une entrée en même temps.

Abandon

Envoie une requête au serveur pour lui dire d'abandonner une opération en lui fournissant son identifiant. Tous les serveurs ne la prennent pas en charge.

Unbind

Ferme la connexion. Il n'y a pas de réponse.

Les URL LDAP

Les Uniform Resource Locator (URL) LDAP sont une méthode simple pour interroger un annuaire LDAP et qui ne nécessite aucune notion de programmation.

```
ldap[s]://<hostname>:<port>/<base_dn>?<attributes>?<scope>?&<filter>
?<extensions>
```

— **hostname** — Addresse du serveur.
— **port** — Port TCP de la connexion.
— **base_dn** — DN du point de départ de la recherche.
— **attributes** — Attributs que l'on veut récupérer, séparés par des virgules.
— **scope** — Portée, voir 1.1.3.
— **filter** — Filtre, voir 1.1.3. Le filtre par défaut est `(objectClass=*)`.
— **extensions** — Les extensions sont un moyen pour pouvoir ajouter des fonctionnalités aux URL LDAP tout en gardant la même syntaxe. On peut inclure plusieurs extensions dans une URL, en les séparant par des virgules. Les extensions ont le

format suivant : `type=value`. La partie `value` est optionnelle. Si elles sont préfixées par un '!', elles signalent une extension critique. Dans ce cas, le client et le serveur doivent supporter tous les deux l'extension.

Exemples :

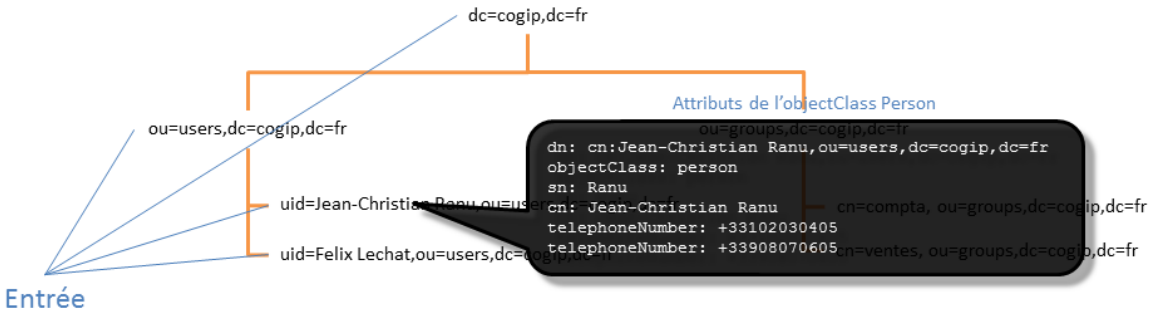
- Par tous les User IDentifier (UID) de la branche users de l'Afpa : `ldap://localhost:389/ou=users,dc=afpa,dc=fr?uid?sub`
- Lecture de toutes les personnes du service vente : `ldap://localhost:389/ou=vente,dc=afpa,dc=fr?cn,tel,mail?scope=sub?(objectclass=person)`
- Lecture des objets personnes d'un annuaire : `ldap://localhost:389/??sub?objectclass=person`
- Recherche de Jean-Christian Ranu : `ldap://localhost:389/uid=Jean-ChristianRanu`
- Recherche approximative d'une personne : `ldap://localhost:389/ou=users,dc=afpa,dc=fr?mail,uid,sub?(sn=Ranu)`

1.1.4 Modèle d'information

Le modèle d'information désigne les données contenues dans l'annuaire (Directory Information Tree (DIT)).

L'arborescence d'informations (DIT)

Les informations sont présentées sous forme d'une arborescence hiérarchique appelée DIT.



- L'*entrée* est l'unité de base de l'annuaire (Directory Service Entry (DSE)). Elle correspond à une branche de l'arborescence.
- Chaque entrée correspond à un objet appelé `objectClass`, par exemple une personne, un objet, des paramètres...
- L'`objectClass` est constitué d'un ensemble de paires clé/valeur appelées *attributs*, qui sont obligatoires ou facultatifs.
- Les *types d'attributs* sont des règles de codage et de correspondance qui déterminent les types de données et les comparaisons à appliquer lors d'une recherche.
- Le *schéma* est l'ensemble des définitions d'objets et d'attributs qu'un serveur LDAP peut gérer. Cela permet de définir si un attribut peut avoir une ou plusieurs valeurs et/ou de les regrouper par objet (`objectclass`) pour définir s'ils sont obligatoires ou pas.

DN et RDN

Un DN se construit en prenant le nom de l'élément, appelé RDN (le chemin de l'entrée par rapport à un de ses parents), et en lui ajoutant l'ensemble des noms des entrées parentes.

On peut comparer le DN et le RDN au chemin d'un fichier dans un répertoire. Le DN, c'est le chemin complet. Il est unique et permet de situer l'entrée dans l'arbre. L'entrée peut être déplacée, et donc le DN redéterminé.

Le RDN peut avoir plusieurs valeurs, par exemple en cas d'homonymie, pour différencier les entrées. Dans ce cas, il faut séparer les valeurs par '+'. Par exemple `dn: cn:Jean Martin+ou:ventes,ou=users,dc=afpa,dc=fr` et `dn: cn:Jean Martin+ou:compta,ou=users,dc=afpa,dc=fr`.

Ces RDN à plusieurs valeurs sont pratiques mais peuvent créer des confusions. Il vaut mieux les utiliser qu'en dernier recours.

Les attributs

Une entrée peut avoir un ou plusieurs attributs. L'attribut est séparé de sa valeur par " :" et, s'il a plusieurs valeurs, est écrit sur plusieurs lignes.

```
dn: cn:Jean Martin+ou:ventes,ou=users,dc=afpa,dc=fr
objectClass: person
cn:Jean Martin
ou:ventes
telephoneNumber: +33 1 02 03 04 05
telephoneNumber: +33 9 08 07 06 05
```

Il existe 2 types d'attributs :

1. *Les attributs normaux* — Ce sont les attributs habituels (nom, prénom, ...) qui caractérisent l'objet.
2. *Les attributs opérationnels* — Ce sont les attributs auxquels seul le serveur peut accéder afin de manipuler les données de l'annuaire (dates de modification, ...).

Les classes d'objets

`objectClass` est un regroupement d'attributs. Il définit une entrée, de la même manière qu'en programmation orientée objet. Ces attributs sont obligatoires ou facultatifs.

`objectClass` est lui-même un attribut obligatoire d'une entrée. C'est dans les schémas que l'on définit le type de classe et le caractère obligatoire ou facultatif d'un attribut.

Il existe 3 types de classe d'objet :

1. *Structurelle* — Représente un objet réel, comme une personne, un domaine, une entité... Chaque entrée doit avoir au moins une classe d'objet structurelle dans l'attribut `objectClass`.
2. *Auxiliaire* — Sert à compléter les informations d'une classe structurelle. Elle ne peut pas être seule.
3. *Abstraite* — Comme en programmation orientée objet, ne peut pas être utilisée directement, mais est plutôt étendue par une autre classe d'objet.

Les schémas

Un schéma est un fichier qui décrit les attributs disponibles et les `objectClass` qui y font appel.

On a vu que les `objectClass` regroupent des attributs. De la même façon, les schémas regroupent les `objectClass`. Un annuaire peut avoir plusieurs schémas.

On peut créer de nouveaux schémas pour définir de nouveaux objets, mais il est parfois plus simple d'étendre un `objectClass` existant pour créer des attributs supplémentaires.

Type d'attribut

Chaque type d'attribut est identifié par un OID.

```
attributetype ( 2.5.4.20 NAME 'telephoneNumber'
DESC 'An integer uniquely identifying a user in a domain'
EQUALITY telephoneNumberMatch
SUBSTR telephoneNumberSubstringMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.50 )
```

Les types d'attribut ont des propriétés :

- **NAME** — Nom du type d'attribut.
- **DESC** — Description.
- **OBSOLETE** — Indique si le type d'attribut est actif.
- **SUP** — Définit le supertype.
- **EQUALITY** — Type d'égalité à mettre en œuvre à la recherche.
- **ORDERING** — Correspondance de tri.
- **SUBSTR** — Correspondance de substring.
- **SYNTAX** — OID de la syntaxe, et nombre maximal de caractères entre accolades.
- **SINGLE-VALUE** — Restreint l'attribut à une seule valeur.
- **COLLECTIVE** — Indique si le type d'attribut est collectif.
- **NO-USER-MODIFICATION** — L'utilisateur ne peut pas le modifier.
- **USAGE** — Indique l'application.

objectClass

Comme pour les attributs, l'`objectClass` est toujours identifié par un OID.

```
( 2.5.6.6 NAME 'person'
 SUP top
 STRUCTURAL
 MUST ( sn $ cn )
 MAY ( userPassword $ telephoneNumber $ seeAlso $ description ) )
```

La classe `person` étend la classe `top` et est une classe structurelle, qui représente un objet réel. Les attributs `sn` et `cn` sont obligatoires. Les attributs `userPassword`, `telephoneNumber`, `seeAlso` et `description` sont facultatifs. Les `objectClass` ont des propriétés :

- **NAME** — Nom de la classe.
- *Type d'objet* — **STRUCTURAL**, **ABSTRACT** ou **AUXILIARY** (voir 1.1.4).

— **DESC** — Description de la classe
— **OBSOLETE** — Cette classe ne doit plus être utilisée.
— **AUX** — Liste des classes auxquelles cet objet peut appartenir.
— **MUST** — Regroupe les attributs obligatoires.
— **MAY** — Liste les attributs facultatifs.
— **NOT** — Liste les attributs qui ne doivent pas être utilisés.
— **SUP** — Classe parent étendue. Toutes les entrées héritent directement de la classe `top`.

Le format LDIF

LDAP Data Interchange Format (LDIF) est un format qui standardise la configuration du serveur LDAP et permet d'importer/exporter les données.

— les commentaires commencent par "#" et ne font qu'une ligne
— une entrée forme un paragraphe, et un fichier LDIF respecte les schémas, qui valident une entrée
— chaque ligne constitue un attribut
— les attributs sont constitués d'une clé et d'une valeur séparées par " :"

Un fichier LDIF pourrait ressembler à ceci :

```
dn: cn=Jean-Christian Ranu,ou=ventes,dc=cogip,dc=fr
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Jean-Christian Ranu
cn: JC Ranu
displayName: Jean-Christian Ranu
sn: Ranu
givenName: Jean-Christian
initials: JCR
title: manager, product development
uid: jcranu
mail: jc.ranu@cogip.fr
telephoneNumber: +33 1 02 03 04 05
mobile: +33 6 78 90 01 02
o: COGIP
ou: ventes
departmentNumber: 2604
employeeNumber: 42
employeeType: full time
preferredLanguage: fr, en-gb;q=0.8, en;q=0.7
labeledURI: http://www.cogip.fr/users/jcranu
```

1.1.5 Modèle de sécurité

Le modèle de sécurité désigne le mécanisme permettant aux clients de s'identifier et d'accéder aux données en fonction de leurs droits.

Le binding

Avant d'interroger l'annuaire, le client doit se connecter au serveur. On appelle cette opération le *binding*. Il peut s'agir d'une authentification simple, mais on peut appliquer des droits particuliers en utilisant des ACL.

1. Le client authentifie l'utilisateur avec son mot de passe à l'aide du DN pour déterminer ses droits. Il peut aussi se connecter en anonyme pour une simple recherche.
2. Le serveur compare le mot de passe saisi et celui de l'entrée (valeur de l'attribut `userPassword`) en chiffrant le tout.

```
dn: cn:Jean-Christian Ranu, ou=personnes,dc=afpa,dc=fr
objectClass: person
cn:Jean-Christian Ranu
sn: Ranu
userPassword: {MD5} Xr4il0zQ4PC0q3aQ0qbuaQ==
```

Méthodes d'authentification

Il existe plusieurs méthodes pour que le client puisse s'identifier.

Authentification anonyme

Établit une connexion en utilisant un DN et un mot de passe vides. Cette authentification est assez courante, entre autre pour les serveurs de messagerie qui recherches des contacts.

Authentification simple

N'utilise aucun chiffrement pour transmettre les données. Envoie un DN et un mot de passe en clair sur le réseau. La valeur pour le mot de passe (dans `userPassword`) peut, elle, être chiffrée.

Chiffrement des communications

Chiffrer la communication entre l'application cliente et l'annuaire permet de garantir la confidentialité des données. Le chiffrement, via Secure Sockets Layer (SSL) ou TLS peut être utilisé via 2 méthodes :

1. LDAP over SSL (LDAPS), qui communique par défaut sur le port 636. Le principe est le même que pour l'authentification simple, sauf que la communication est chiffrée. Il est préférable d'utiliser startTLS.
2. startTLS, qui utilise le port 389 par défaut. Le client doit utiliser la fonction startTLS pour s'authentifier, mais a la possibilité d'envoyer les données chiffrées ou non, en fonction du besoin des requêtes.

Simple Authentication and Security Layer (SASL)

SASL permet d'ajouter des méthodes d'authentification à des protocoles orientés connexion tels que LDAP. Il traite le mécanisme d'authentification avant même la transmission des données utilisateur. D'abord le serveur authentifie la requête, puis valide les données.

Le client et le serveur auront la possibilité de sélectionner la méthode d'authentification utilisée. Il est également possible de mettre en place une couche de connexion sécurisée comme SSL/TLS, indépendamment du chiffrement de données vus ci-dessus.

Les droits

Les ACL interviennent après le *binding*. Une fois l'utilisateur authentifié, toutes les requêtes et manipulations de données sont faites en fonction de ses droits.

Ce paramètre n'est pas dans les schémas ou au niveau de l'entrée mais dans le fichier de configuration du serveur LDAP.

1.1.6 La réplication

Pour assurer un service, il faut non seulement des sauvegardes mais également la *redondance*. Il faut répliquer les serveurs LDAP. Mais il n'y a pas de normalisation pour la réplication. Répliquer un annuaire d'un fournisseur à un autre n'est donc pas forcément possible. Les méthodes peuvent être différentes.

Cela dit, la réplication n'est pas obligatoire. Il y a des cas il faut mettre en place une copie, de tout ou partie de l'annuaire :

— Si un application fait un *usage important*, au point de ralentir les autres.
— Si le serveur atteint ses *limites*.
— Si une entreprise est *multi-site*.
— Si l'*indisponibilité* de l'annuaire pose problème.
— Si l'annuaire est un composant important de l'architecture d'une entreprise (*haute disponibilité*).

On a dans ce cas un annuaire *maître*, qui envoie alors par le biais du format LDIF toutes les modifications effectuées sur un annuaire *esclave*, ce qui permet :

— un équilibrage de charge, en redirigeant le client vers l'un ou l'autre
— une copie conforme de l'annuaire, utile en cas de crash.

On peut répliquer de deux manières :

1. le mode **maître-esclave** — Le plus courant : la réplication est unidirectionnelle. L'annuaire maître envoie toutes les modifications à un annuaire esclave. Ceci n'autorise l'écriture que sur l'annuaire maître. L'esclave n'est alors disponible qu'en lecture seule.
2. le mode **maître-maître** — La réplication est bidirectionnelle, chaque annuaire pouvant être maître de l'autre. On peut alors écrire indifféremment sur l'un ou l'autre.

Il est aussi possible de chaîner les réplications pour en obtenir plusieurs.

1.1.7 La distribution

La distribution permet de faire pointer un lien vers un autre annuaire pour une branche particulière : les *referrals*. Cela permet de déléguer la gestion de cette branche. Par exemple, l'annuaire 1 délègue la branche `ou=groups,dc=afpa,dc=fr` à l'annuaire 2.

Au niveau de l'annuaire 1, c'est une entrée de la classe `referral`, qui contient un attribut `ref` contenant l'adresse de la suite de l'arborescence.

```
dn: ou=groups,dc=afpa,dc=fr
objectClass: referral
ref: ldap://ldap2.afpa.fr/ou=groups,dc=afpa,dc=fr
```

1.1.8 Alias

On peut aussi avoir des liens symboliques au sein du même annuaire : les *alias*. Normée par la RFC 4512, c'est une classe object structurelle qui définit au sein d'un même annuaire un DN qui contient l'information avec l'attribut `aliasedObjectName`.

```
dn: uid=jcranu,ou=users,dc=afpa,dc=fr
objectClass: alias
objectClass: extensibleObject
aliasedObjectName: uid=dauteuil,ou=users,dc=tssr,dc=fr
```

L'entrée `uid=jcranu,ou=users,dc=afpa,dc=fr` référence alors l'entrée `uid=dauteuil,ou=users,dc=tssr,dc=fr`.

Contrairement aux referrals, la suppression d'une données référencée n'a besoin d'aucune action spéciale. Le serveur désactive automatiquement le lien.

Chapitre 2

Windows

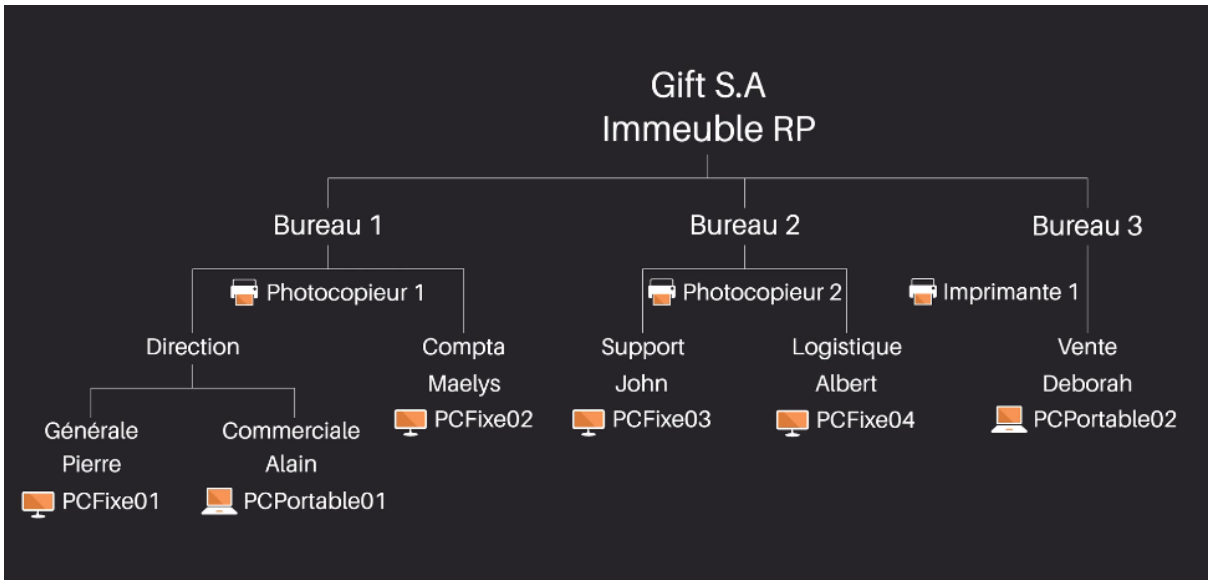
2.1 Active Directory

Active Directory (AD) est une implémentation d'un serveur LDAP dans Windows, utilisé dans 80% des entreprises.

2.1.1 Préparation

Nous allons prendre un exemple d'une entreprise fictive, Gift S.A., qui a 3 bureaux.

— Il nous faut avoir une vue globale de l'entreprise pour en identifier les ressources.
— La représentation hiérarchique permet de simplifier la conception.
— Une cartographie des ressources d'une entreprise permet de préparer l'utilisation d'un annuaire d'entreprise.



Architecture

L'entreprise est représentée par une *forêt* AD. Une *forêt* est un ensemble de domaines AD qui partagent une structure logique, un schéma de données, une configuration d'annuaire et des fonctionnalités identiques.

Il est très courant de n'avoir qu'un seul domaine au sein d'une forêt mais les entreprises de grande taille vont ajouter des domaines pour identifier leurs différentes structures (`fr.gift.sa`, `us.gift.sa`, ...).

Le domaine représente une partition de forêt (on parle de *partition d'annuaire*). Dans le domaine on peut créer des objets identifiés de manière unique sur le réseau.

À l'intérieur du domaine, on retrouve les *unités organisationnelles* (OU). Ce sont des conteneurs qui peuvent représenter les différents services.

Dans ces conteneurs, on trouve les *objets*. Ces objets représentent les utilisateurs, les imprimantes, les postes, etc. On peut les regrouper au sein de *groupes*.

Avoir de multiples domaines au sein d'une forêt permet de segmenter les données, ce qui augmente la sécurité.

AD est hébergé sur un serveur. On l'appelle *contrôleur de domaine*. Les services AD hébergés sur le contrôleur de domaine s'appellent des Active Directory Domain Service (AD-DS). AD-DS s'occupe du stockage des données d'annuaire, de l'authentification et de la réplication si il y a plusieurs contrôleurs de domaine.

Rôles de contrôleur de domaine

Il y a 5 rôles :

1. **Maître de schéma** (Schema Master) — Contrôle les modifications apportées au schéma de données AD. Le schéma définit les différentes informations (attributs) qui sont associées aux types d'objets. Il ne doit y avoir qu'un seul *Schema Master* dans un annuaire.
2. **Maître d'attribution de noms de domaine** (Domain Naming Master) — Contrôle l'ajout et la suppression des noms de domaines dans une forêt. Les noms de domaine au sein d'une forêt doivent être uniques. Il ne faut donc y avoir qu'un seul *Domain Naming Master* dans une forêt.
3. **Émulateur de contrôleur de domaine principal** (Primary Domain Controler Emulator) — Intéressant car permet le support de clients NT4 (Windows NT étant une des premières versions de Windows conçu pour fonctionner en réseau). Fournit également l'*horloge de référence* du domaine via le protocole NTP. Il faut qu'il y ait une instance de ce rôle par domaine.
4. **Maître Registered IDentifier (RID)** (RID Master) — Fournit des tranches d'identifiants uniques aux autres contrôleurs de domaine. Le RID est un identifiant unique relatif à un domaine. Il fait partie du Security IDentifier (SID). Microsoft base l'identification des ressources sur ces identifiants.
5. **Maître d'infrastructure** (Infrastructure Master) — Permet de synchroniser les changements effectués sur les objets au sein des différents domaines en gérant les réplications.

Les types d'objets de l'annuaire

On retrouve 3 types d'objet dans un annuaire AD :

1. **Les OU** — Ce sont les premiers objets de l'annuaire. Il permettent d'organiser et de structurer l'annuaire.
2. **Les groupes** — Ils permettent de simplifier la gestion des ressources en centralisant des objets selon des critères. On peut par exemple regrouper les utilisateurs de la comptabilité devant avoir un accès à des ressources précises.
3. **Les ressources** — Elles sont le cœur de l'annuaire. Elles permettent de lister les utilisateurs, les imprimantes, les postes, avec des informations appelés attributs.

Les types d'installations

On a le choix entre créer une nouvelle forêt avec un ou plusieurs domaines, ou bien créer un nouveau domaine au sein d'une forêt déjà existante. Si l'on crée une nouvelle forêt, on va pouvoir avoir un domaine unique, appelé *domaine racine*. C'est le choix le plus courant.

En gardant notre exemple de *Gift S.A.*, ce domaine sera réparti sur deux contrôleurs de domaine. Cela offre la redondance primordiale pour fournir un service lors des mises à jour, sauvegardes, maintenances, ...

Les groupes

Les groupes permettent non seulement de simplifier la gestion, mais aussi de centraliser les droits d'accès et d'appliquer des paramètres de sécurité.

Voici les différents types de groupes. Les 3 premiers définissent la portée :

— **Groupe Domaine Local** — Contient des membres issus du domaine. Cela peut poser un problème lors de l'ajout d'un second domaine à la forêt. Les groupes locaux d'un domaine ne pourront pas contenir d'objets issus de l'autre domaine.
— **Groupe Global** — Permet de gérer des ressources issues de différents domaines disposant d'une relation d'approbation.
— **Groupe Universel** — Utilisé lorsque les ressources peuvent être issues de tous les domaines d'une forêt. Ce type de groupe autorise des membres issus de n'importe quel domaine.
— **Groupe de sécurité** — Pour gérer les droits d'accès.
— **Groupe de distribution** — Principalement utilisé en lien avec le système de messagerie *Microsoft Exchange* pour créer des listes de contacts pour des listes de diffusion.

Nommer les groupes

De bonnes habitudes pour nommer les groupes aident énormément à long terme.

Il est intéressant de préfixer le nom du groupe par sa portée. Par exemple `GDL`, `GG` et `GU` pour `Groupe de Domaine Local`, `Groupe Global` et `Groupe Universel`. Ainsi, quand on va attribuer des droits d'accès, on retrouvera facilement la portée des objets. On pourra par exemple nommer une imprimante `GG_R_Imprimante_RDC`. Pour les utilisateurs souhaitant accéder à internet, un nom de groupe cohérent serait `GU_U_Accès-Internet`.

Principe des Group Policy Object (GPO)

Les GPO, ou *stratégies de groupe*, permettent d'appliquer des politiques de sécurité sur des objets utilisateurs ou ordinateurs. Les configurations possibles couvrent tous les besoins : de la personnalisation du poste de travail à la sécurisation fine des accès en passant par l'installation de logiciels ou d'applications.

Pour nous y retrouver, il y a des modèles d'administration : plus de 1700 pour un utilisateur Windows 10 et plus de 2400 sur un ordinateur Windows 10. On peut retrouver les modèles avec la commande `gpedit.msc`. Ne l'utiliser que pour la lecture : ce sont des stratégies locales. Pour gérer les stratégies du domaine, il faut un autre éditeur, identique.

Les GPO s'appliquent à l'authentification puis toutes les 60 à 120 minutes par défaut. Une GPO peut être liée à un domaine, un site ou une OU.

2.1.2 Installation

Nous allons utiliser deux serveurs Windows Server 16 et un client Windows 10.

Il faut commencer par installer l'OS sur le premier serveur. Une fois l'installation faite, soit on refait la même chose pour le deuxième, soit on clone la première Virtual Machine (VM). Si on décide de cloner, il faut faire attention à un détail : Le SID sera le même, ce qui n'est pas ce qu'on veut. Il doit être unique. Pour cela :

- Cloner la machine
- Exécuter la commande `sysprep` qui se situe dans le répertoire `C:\Windows/System32`. Sélectionner l'option "Généraliser" puis redémarrer la VM.

On va appeler nos deux serveurs `SRVDC1` et `SRVDC2`, et notre client `PCFIXE01`.

Il faut que les serveurs aient le rôle de serveur DNS en plus du rôle AD-DS.

Pour chaque serveur, on commence par installer les services AD-DS dans le gestionnaire des serveurs :

`Ajouter des rôles et des fonctionnalités > Installation basée sur un rôle ou une fonctionnalité > Service AD-DS`.

Il faut que toutes les machines puissent communiquer. S'assurer de la bonne configuration IP et faire des ping.

Pour l'installation des services AD-DS, peu importe quel serveur est installé en premier. Par contre, il va falloir maintenant promouvoir les serveurs en contrôleurs de domaine. On fait ça un par un.

Le premier contrôleur de domaine

Dans le service AD-DS installé, on va dans la bannière jaune et on clique sur `Promouvoir ce serveur en contrôleur de domaine`. On ajoute une nouvelle forêt et on choisit un nouveau nom de domaine : `gift.sa`.

Ensuite on a le choix du *niveau fonctionnel* du domaine et de la forêt. Cela permet de faire fonctionner le contrôleur de domaine à un niveau inférieur de Windows, compatible avec un équipement plus daté sur le réseau. Dans notre cas, on n'a que des Windows Server 2016 donc on laisse ça.

On nous propose le service DNS puis le *catalogue global*. Cette option permet à un contrôleur de domaine de stocker une copie complète de la base de données AD avec l'ensemble des informations.

On choisit enfin un mot de passe qui nous permettra de réparer l'annuaire s'il le faut.

La partie DNS risque de se plaindre parce qu'elle ne trouve pas d'autorité parente pour le nom de domaine. On peut ignorer ce message.

On choisit un nom de domaine netBIOS, qui sera généralement le nom de domaine racine sans extension.

On laisse les chemins proposés par défaut pour l'emplacement du stockage des informations de l'annuaire.

Au redémarrage, on s'authentifie avec un nom de domaine avant le nom d'utilisateur : `GIFT\Administrateur`.

Le deuxième contrôleur de domaine

Pour le deuxième serveur, les étapes sont quasiment identiques, sauf qu'au début, au lieu de créer une nouvelle forêt, on ajoute le serveur à un domaine existant.

Outils d'administration

Dans le menu *Outils d'administration*, on peut trouver deux outils utiles.

Le **Centre d'administration Active Directory** centralise les possibilités d'administration. Il a été rajouté depuis Windows Server 2012.

L'outil **Utilisateurs et ordinateurs Active Directory** affiche une représentation graphique de l'annuaire. On va l'utiliser pour ajouter nos premiers objets.

On clique sur le nom du domaine et on sélectionne `Nouveau` puis `Unité d'Organisation`, qu'on va nommer "Direction".

Pour ajouter un utilisateur, on clique sur le nom de l'OU que l'on vient de créer et on sélectionne `Nouveau` puis `Utilisateur`. On a alors des champs à remplir et on a la possiblité de faire changer le mot de passe lors de la première connexion de l'utilisateur.

En entreprise, on référence souvent les informations comme le lien de responsabilité (qui est responsable de cet utilisateur dans l'entreprise). Il est aussi courant de spécifier sur quel ordinateur un utilisateur peut s'authentifier.

Nous pouvons maintenant créer l'arborescence de `Gift S.A.`.

Intégration du poste client

Pour intégrer un client à un domaine, il faut que le client ait les contrôleurs de domaines en tant qu'adresses DNS. Le client va s'enregistrer en tant que ressource dans le domaine. Intégrer des postes à un domaine permet de les classer dans la hiérarchie. Les postes sont ainsi en lien permanent avec les contrôleurs de domaine.

Quand on ne spécifie rien, l'objet est créé dans une OU spécifique créé par défaut (*Computers* pour les postes clients). Il est ensuite possible de les déplacer au sein de la hiérarchie.

Par défaut, tout ordinateur se trouve dans un *groupe de travail* (*WORKGROUP*). Dans les propriétés système, on va modifier le type d'adhésion du poste en sélectionnant `Domaine` et en entrant le domaine `gift.sa`.

2.1.3 Stratégies de groupe

Paramètres Dans `Outils > Gestion des stratégies de groupe`, on peut trouver les stratégies de groupes, GPO. Ils permettent d'automatiser certaines tâches et d'agir sur un poste depuis le contrôleur de domaine. Il y en a 2 par défaut :

1. Politique par défaut des contrôleurs de domaine. On y trouve les accès réseau, la liste des comptes utilisateur capables d'ajouter des postes au domaines, les paramètres d'échanges de données entre les membres du domaine.
2. Politique par défaut du domaine. On y trouve des paramètres les stratégies de mot de passe, de verrouillage des comptes, de stockage des empreintes de mot de passes.

Ces paramètres par défaut sont avant tout des paramètres de sécurité pour l'authentification.

Créer une GPO

Pour créer une nouvelle GPO : `Objets de stratégie de groupe > Nouveau`. Pour créer un nouvel objet GPO et le lier à un domaine : `Clic sur le nom de domaine > Créer un objet GPO dans ce domaine, et le lier ici...`. On le nomme de manière identifiable, par exemple `GPO_U_FondEcran`.

Quand on modifie la GPO, on accède à deux configurations :

1. *Configuration ordinateur* — Le paramètre s'applique à la machine, peu importe l'utilisateur connecté.
2. *Configuration utilisateur* — Le paramètre s'applique en fonction de l'utilisateur connecté à la machine, peu importe la machine.

Dans les deux cas, on retrouve :

- **Stratégies**
  - **Paramètres du logiciel** — Permettent de gérer les déploiements de logiciels de façon centralisée. L'installateur du logiciel doit être au format `MSI`.
  - **Paramètres Windows** — Regroupent les différents scripts qu'il est possible d'écrire pour exécuter des actions à certains moments clés. Certains paramètres ne sont disponibles que pour l'ordinateur, d'autres que pour l'utilisateur. C'est aussi qu'on peut configurer une stratégie de mot de passe.
  - **Modèles d'administration** — Fichiers au format `ADMX` qui permettent de gérer la base de registre de Windows. On dispose d'une description du paramètre et de l'impact qu'il aura sur le client.

— **Préférences** — Apparus avec Windows 2008, simplifient des tâches avec une interface graphique proche de celle que l'on trouve sur un poste client.

S'il y a besoin d'un dossier de partage (par exemple pour faire mettre un papier peint sur les postes), il faut soit le créer, soit utiliser le dossier `NETLOGON` qui est créé par défaut.

Vérifier que la stratégie s'applique bien

Si le poste client est éteint, il suffit de se connecter avec un utilisateur préalablement créé pour vérifier que la stratégie de groupe a été appliquée. On peut aussi utiliser deux commandes :

1. `gpresult` — Affiche les informations sur l'application des stratégies à partir d'un poste, pour l'utilisateur courant.
2. `gpupdate` — Applique les paramètres des GPO. La commande seule applique les paramètres modifiés uniquement. On peut forcer l'application de tous les paramètres avec `/force`.

Modèles d'administration

Pour certains composants comme le navigateur ou Office et les nouvelles versions de Windows, on se rendra compte que certains paramètres n'existent pas. Il va falloir les rajouter avec les *modèles d'administration*.

Microsoft a mis en place le *Central Store* qui centralise les modèles d'administration dans le partage `SYSVOL`. Il faut récupérer les fichiers de modèles d'administration au format ADMX et des les copier dans `C:\Windows\System32\SYSVOL\<domain>\Policy` dans un répertoire nommé `PolicyDefinitions`. On y met les fichiers ADMX et le répertoire de la langue des systèmes que l'on souhaite administrer.

Si on retourne dans les `Modèles d'administration`, il nous indique qu'ils ont été récupérés sur le magasin central. On peut maintenant y trouver les paramètres qu'on a ajoutés.

Sécurité

Dans le rôle AD-DS du gestionnaire de serveur, on peut trouver le *Best Practice Analyzer*. Il permet de faire un scan de l'AD et liste les erreurs rencontrées. Les erreurs ont un niveau de gravité : la plupart sont de type `Informations`, ce qui peut indiquer une bonne conformité. Par contre, celles qui sont d'un niveau de gravité de type `Avertissements` ou `Erreurs` doivent être résolues. On peut n'afficher que ces niveaux de gravité avec un filtre. L'explication fourni avec l'erreur devrait aider à la résoudre. Après résolution, l'avertissement correspondant devrait disparaître.

Maintenant, il faut trouver les groupes disposant de privilèges élevés dans l'OU `Users` et dans l'OU `Builtin`. On trouve des groupes qu'il faut utiliser le moins possible, et qui ne doivent pas comporter un grand nombre de membres :

— Administrateurs clés
— Administrateurs clés Entreprise
— Administrateurs de l'entreprise

— Administrateurs du schéma
— Admins du domaine
— DNSAdmins

On y trouve aussi le groupe `Administrateurs`.

Pour aller plus loin dans la sécurisation, on peut mettre en place deux nouveaux rôles :

1. **Services Windows Server Update Services (WSUS)** — Permettent de centraliser la gestion de correctif de sécurité sur les clients et les serveurs. Il spécifie une stratégie d'acceptation et de déploiement.
2. **Services Network Policy Server (NPS)** — Autorisent ou non l'accès au réseau à des postes clients ou à des utilisateurs en fonction de leur identité et de leurs droits d'accès.

2.1.4 Sécurisation

Un annuaire centralisé comme AD qui gère toutes les configurations de l'entreprise est une cible en or pour les attaquants. La sécurité de l'AD est donc prioritaire pour les administrateurs.

Il faut réduire la surface d'attaque pour un assaillant.

— **Les utilisateurs** — Il faut surtout les sensibiliser. Il leur faut des mots de passe complexes qu'ils sauront retenir sans les notes sur un post-it. Cela peut se faire par une GPO. Mais il faut surtout demander aux utilisateurs de changer de mot de passe dès qu'une suspicion de compromission est identifiée.
Il est aussi possible de ne pas afficher le nom du dernier utilisateur à s'être connecté sur un poste : `Paramètres de sécurité > Option de sécurité`, puis le paramètre `Ouverture de session interactive : ne pas afficher le dernier nom d'utilisateur`.
— **Le réseau** — Segmenter un réseau est une bonne pratique de sécurité. Dans cette idée, il est possible de répartir les contrôleurs de domaine sur différents sites. Pour ça, il faut aller dans `Sites et services Active Directory`. On peut créer un site distant qui sera routé vers le site principal (le `Default-First-Site-Name`). On personnalise le calendrier de réplications en cliquant sur le nom du serveur et sur `NTDS settings`. La réplication inter-site peut se faire via IP (conseillé) ou via SMTP.
— **Les systèmes** — Il est prudent de garder la main sur les configurations. AD nous permet d'identifier les postes mal configurés ou apportés par les utilisateurs, en activant les journaux d'audit, via une GPO.
Il est également possible d'interdire l'accès à internet sur les contrôleurs de domaine, via le pare-feu de Microsoft. Grâce à une GPO, on peut bloquer les flux vers les ports 80 et 443 vers tous les hôtes distants. Avec le pare-feu d'entreprise, il est également possible d'autoriser l'accès à internet qui via la plage IP dédiée aux postes clients.
La meilleure pratique est de n'utiliser l'accès à internet que via un serveur proxy qui centralisera les requêtes et pourra filtrer certains sites ou passer un anti-virus avant de renvoyer la réponse des serveurs à visiter.

— **Les logiciels** — Le déploiement de logiciels permet de fournir des logiciels aux utilisateurs sans passer par chaque poste. Il permet aussi de suivre l'évolution des licences.
Au niveau des GPO, deux stratégies :

1. Installer des logiciels en fonction des ordinateurs.
2. Installer des logiciels en fonction de profils d'utilisateurs.

On peut également bloquer le lancement de certaines applications via `Applocker`. Ou plus simplement de mettre en place une stratégie de restriction logicielle, appellée `SRP`.
Pour cela, créer une GPO puis dans `Configuration ordinateur > Paramètres de sécurité > Stratégies de restriction logicielle`. On peut voir 5 éléments :

1. Niveaux de sécurité
2. Règles supplémentaires
3. Contrôle obligatoire
4. Types de fichiers désignés
5. Éditeurs approuvés

On a aussi par défaut deux emplacements à partir desquels il n'y a pas de restriction. On peut rajouter un chemin pour le restreindre, pour éviter par exemple que les utilisateurs ne puissent lancer des exécutables depuis le dossier `Téléchargements`.

On peut aussi configurer un contrôleur de domaine en lecture seule, ce qui ne permet pas d'action d'administration. Pour cela, il faut configurer un nouveau serveur et cocher la case `RODC` lors de sa promotion en contrôleur de domaine. Il authentifiera les utilisateurs et appliquera les GPO mais il ne sera pas possible de prendre le contrôle du domaine et de créer par exemple un nouvel utilisateur Administrateur.

2.1.5 Sauvegarde de la base de données

AD étant une base de données, il faut la sauvegarder à intervalles réguliers. Pour cela, deux façons d'utiliser l'outil de *Sauvegarde Windows Server* :

1. De façon graphique :
   — Dans le `Gestionnaire de serveur`, cliquer sur `Ajouter des rôles et fonctionnalités > Suivant > installation basée sur un rôle ou une fonctionnalité > Suivant`.
   — Sélectionner le serveur, puis `Suivant`.
   — Passer sur les rôles, puis sur l'écran des fonctionnalités sélectionner `Sauvegarde de Windows Server > Suivant > Installer`.

   Dans le gestionnaire de serveur, on peut cliquer sur `Outils > sauvegarde de Windows Server`. Pour effectuer une sauvegarde de l'AD :
   — Lancer l'outil.
   — Sélectionner `Sauvegarde locale`.
   — `Action > Sauvegarde unique > Différentes options > Suivant`
   — On sélectionne l'`état du système`, puis `Suivant`.
   — Après avoir spécifié le type de destination, on peut effectuer la sauvegarde.

2. En ligne de commande. Plus pratique et rapide quand on connaît les arguments. La commande est `wbadmin.exe`.

   Par exemple, pour sauvegarder l'AD sur le lecteur `E:\` :

   ```
   wbadmin start backup -backupTarget:e: -systemstate -vssfull
   ```

   — L'option `-backupTarget:e:` spécifie le dossier de destination, ici `E:\`.
   — L'option `-systemstate` permet de sauvegarder le répertoire `SYSVOL` ainsi que l'AD mais surtout le fichier qui stocke la base de données : `NTDS.DIT`.
   — L'option `-vssfull` permet d'effectuer une copie complète.

2.2 PowerShell

2.2.1 Windows PowerShell ISE

Le plus simple pour écrire les scripts PowerShell est l'IDE *Windows PowerShell ISE*. Il en existe d'autres, comme *PowerGUI*, mais l'avantage de *ISE* est qu'il est intégré dans Windows.

2.2.2 Modes d'exécution

Par défaut, PowerShell est configuré en mode *Restricted*, qui bloque l'exécution des scripts. Nous allons tout d'abord le passer en mode *Unrestricted*.

Les différents modes d'exécution sont les suivants :

— **Restricted** — Aucun script ne peut être exécuté. PowerShell est utilisable uniquement en mode interactif.
— **AllSigned** — Seuls les scripts doivent être signés pour pouvoir être exécutés.
— **RemoteSigned** — Les scripts provenant d'internet doivent être signés avant de pouvoir être exécutés. Les scripts présents sur le poste de travail peuvent être exécutés sans problème.
— **Unrestricted** — Pas de restriction, tous les scripts peuvent être exécutés.

En production, il vaut mieux choisir le mode *AllSigned*.

Pour savoir dans quel mode on est actuellement :

```
PS> Get-ExecutionPolicy
```

Pour changer de mode :

```
PS> Set-ExecutionPolicy Unrestricted
```

2.2.3 Les modules

Certaines fonctions déjà existantes sont présentes dans des modules. Pour obtenir les modules chargés sur la machine :

```
PS> Get-Module
```

Pour en ajouter :

```
PS> Get-Module -ListAvailable
```

Cela liste des modules disponibles. Une fois le module désiré identifié :

```
PS> Import-Module ActiveDirectory
```

On aura des modules différents en fonction du système d'exploitation utilisé.

2.2.4 Types de variables

Voici quelques types de variables courants disponibles avec PowerShell :

- `string` : chaîne de caractères.
- `char` : caractère Unicode sur 16 bits.
- `byte` : caractère non signé sur 8 bits.
- `int` : nombre entier signé sur 32 bits.
- `long` : nombre entier signé sur 64 bits.
- `decimal` : nombre décimal sur 128 bits.
- `bool` : booléen (vrai ou faux).
- `DateTime` : date et heure.
- `array` : tableau de valeurs.

2.2.5 Exemple de script

On souhaite par exemple écrire un script pour nous afficher la liste de tous nos utilisateurs Active Directory dont la dernière connexion remonte à plus de 90 jours afin de vérouiller leur compte pour des raisons de sécurité. En plus, on va envoyer un mail automatique à l'assistance pour les informer de cette désactivation.

```
# Script pour automatiser la désactivation des comptes AD dont la dernière
    connexion > 90 jours
# version 1.0
# Auteur : Tunui Franken

# Force le type d'exécution
Set-ExecutionPolicy Unrestricted

# Importe le module AD
Import-Module ActiveDirectory

# Obtient la liste des comptes inactifs depuis plus de 90 jours
$LockedAccount = Search-ADAccount -UsersOnly -AccountInactive -TimeSpan
    90.00:00:00 -SearchBase "OU=Users,DC=afpa,DC=fr" | Where-Object {$_.enabled}

# Désactive tous les comptes de la liste
$LockedAccount | Set-ADUser -Enabled $false

# Préparation du mail pour prévenir l'assistance
$smtpServer = "mail.afpa.fr"
$from = "DisableADAccount <powershell@afpa.fr>"
$to = "Helpdesk <helpdesk@afpa.fr>"
$subject = "[INFO] Comptes AD last logon > 90 jours"
$body = "
<html>
```

```
    <head></head>
    <body>
        <p>
            Bonjour,<br/>
            Les comptes suivants sont d&eacute;sactiv&eacute;s &agrave; cause d'
                une inactivit&eacute; de plus de 90 jours :<br/>
            $LockedAccount
        </p>
    </body>
</html>
"

# Envoi du mail
Send-MailMessage -smtpserver $smtpServer -from $from -to $to -subject $subject -
    body $body -bodyasHTML -priority High
```

2.2.6 Créer des partages

Il faut avoir un dossier déjà créé, ou bien le créer directement :

```
PS> New-Item -Path "C:\SAUVEGARDE" -ItemType Directory
```

Puis créer le partage et les droits :

```
PS> New-SmbShare -Name Sauvegarde -Path C:\SAUVEGARDE -FullAccess
    Administrateurs
```

On peut par exemple rajouter des droits de lecture à tout le monde :

```
PS> Grant-SmbShareAccess -Name Sauvegarde -AccountName "Tout le monde" -
    AccessRight Read
```

On peut mettre les droits suivants :

- *Full* : contrôle total
- *Change* : modifier
- *Read* : lecture
- *Custom* : personnalisé

Et visualiser le contenu :

```
PS> Get-ChildItem \\Serveur\Sauvegarde -Force
```

`-Force` permet de voir aussi les fichiers et dossiers cachés.

2.2.7 Créer un utilisateur et l'activer

Pour créer un utilisateur avec ces propriétés :

- login : `franken`
- mail : `franken@afpa.fr`
- mot de passe : `Pwd2021`
- pas d'expiration du mot de passe

— ne peut pas changer son mot de passe

```
PS> New-ADUser -Name "FRANKEN Tunui" -SamAccountName franken -
    UserPrincipalName "franken@afpa.fr" -AccountPassword (ConvertTo-
    SecureString -AsPlainText Pwd2021 -Force) -PasswordNeverExpires $true -
    CannotChangePassword $true
```

Par défaut cela crée un nouvel utilisateur dans l'OU `Users` mais il est désactivé. Pour l'activer :

```
PS> Enable-ADAccount franken
```

Si on veut automatiser la création d'utilisateurs, il va falloir faire un script :

```
# Demander les informations de l'utilisateur
$nom = Read-Host "Nom :"
$prenom = Read-Host "Prenom :"
$login = Read-Host "Login:"
$passwd = Read-Host "Mot de passe:"

# Créer l'utilisateur
New-ADUser -Name "$nom $prenom" -SamAccountName $login -UserPrincipalName
    $login@afpa.fr -AccountPassword (ConvertTo-SecureString -AsPlainText $passwd
     -Force) -PasswordNeverExpires $true -CannotChangePassword $true -Enabled
    $true
```

On peut ensuite le lancer avec `powershell .\ajout-utilisateurs.ps1`.

Pour afficher les utilisateurs créés :

```
PS> Get-ADUser -Filter *
```

2.2.8 Créer des groupes

Un script pour ajouter des groupes et y organiser les utilisateurs :

```
# Demander le nom du groupe à créer
$group = Read-Host "Groupe :"
New-ADGroup $group -GroupScope Global

# Demander le nombre d'utilisateurs à insérer dans le groupe
[int] $nombre = Read-Host "Nombre d'utilisateurs à insérer dans le groupe :"

# Demander le nom des utilisateurs à insérer dans le groupe, puis les ajouter
for ($i = 0; $i -lt $nombre; $i++) {
    $nom = Read-Host "Nom de l'utilisateur a insérer dans le groupe $group"
    Add-ADGroupMember -identity $group -Members $nom
    Write-Host "L'utilisateur $nom a bien été inséré dans le groupe $group"
}
```

Pour afficher les groupes créés :

```
PS> Get-ADGroup -Filter *
```

Et pour afficher les utilisateurs d'un groupe :

```
PS> Get-ADGroupMember GROUPE
```

On peut exporter ça vers un fichier CSV :

```
PS> Get-ADGroupMember GROUPE | Export-Csv GROUPE.csv -Encoding UTF8
```

2.2.9 Script de sauvegarde

Imaginons un simple script qui contient cette commande, qui copie le contenu d'un dossier vers un partage sur le réseau :

```
Copy-Item -Path "C:\Users\Franken\DossierImportant\" -Destination "\\SERVEUR
    \Franken\" -Recurse
```

On peut lancer ce script toutes les nuits par exemple, avec le *Planificateur de tâches*. Il faut cliquer sur `Créer une tâche de base`, on peut suivre l'outil graphique pour choisir une fréquence, un horaire, et le chemin vers le script.

Mais le mieux, si on a un domaine AD, serait d'utiliser une GPO pour l'exécution du script.

Sur le serveur, on crée une GPO et on configure une *tâche planifiée* dans la *configuration ordinateur* ou la *configuration utilisateur*. On peut alors mettre les mêmes informations que précédemment, sans oublier d'indiquer à qui la GPO s'applique.

Chapitre 3

Virtualisation

3.1 VirtualBox

3.1.1 Disques virtuels

Pour créer des disques virtuels on a le choix entre deux formats :

— Virtual Disk Image (VDI)
— Virtual Machine Disk File (VMDK)

Pour convertir un disque virtuel VMDK en VDI :

```
VBoxManage clonemedium disk <foo>.vmdk <foo>.vdi --format VDI
```

3.1.2 Modes de réseau

Mode	VM to Host	Host to VM	VM1 to VM2	VM to net/LAN	net/LAN to VM
Host only	+	+	+	−	−
Internal	−	−	+	−	−
Bridged	+	+	+	+	+
NAT	+	port forward	−	+	port forward
NAT service	+	port forward	+	+	port forward

Attention pour le clonage : pour éviter les conflits avec les identifiants de machine, on peut changer l'adresse MAC.

3.1.3 Additions invité

Pour le plein écran en résolution normale, pour laisser la souris dans la machine virtuelle et pour activer la possibilité de dossier de partage entre la VM et l'hyperviseur, dans le menu de la machine vituelle :

```
> Péripheriques > Insérer l'image CD des Additions Invité...
```

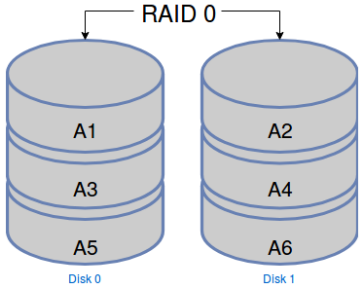
Chapitre 4

RAID

4.1 La théorie

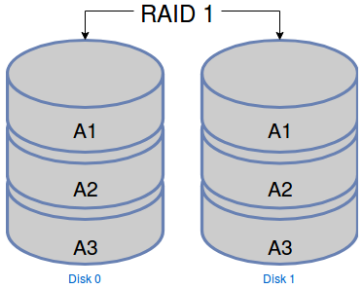
Dans un volume RAID, tous les disques doivent avoir la même capacité. Si ce n'est pas le cas, on peut créer des partitions de la taille du disque le plus petit et construire le RAID avec ces partitions de taille identique.

4.1.1 RAID 0 — Agréger des disques



Chaque fichier est réparti par petits bouts sur plusieurs disques. À chaque lecture ou écriture sur le volume RAID, les disques physiques vont travailler en parallèle et les performances seront meilleures. Autre avantage : pas de place perdue. Avec 3 disques de 10 Go, on a un volume de 30 Go. L'inconvénient, un peut comme avec LVM, c'est que si on perd un seul disque du volume, on perd l'ensemble des données.

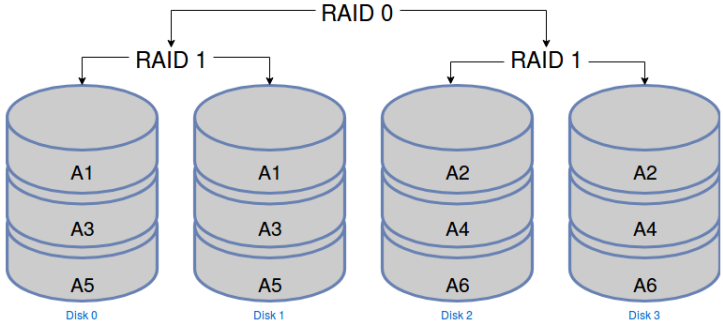
4.1.2 RAID 1 — Stocker des données en miroir



Tous les disques sont des copies exactes. Chaque fois qu'on écrit un fichier, il est écrit sur

chaqun des disques, ce qui est plus long en termes de performances. L'avantage est la fiabilité : tant qu'il reste au moins un disque, aucune donnée n'est perdue. L'inconvénient c'est qu'on perd l'espace de stockage des disques additionnels.

4.1.3 RAID 10 — Compromis entre fiabilité et performances



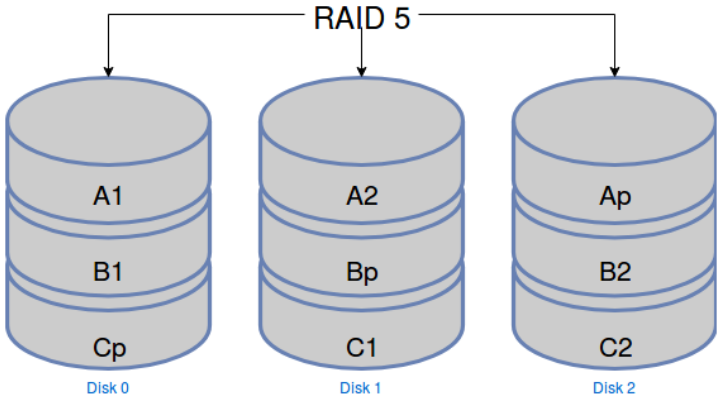
1O veut en fait dire $1 + 0$. C'est un mélange du RAID 0 et du RAID 1. Les disques sont regroupés par grappes en RAID 1 et les fichiers sont répartis sur plusieurs grappes en RAID 0. Il faut donc au moins 4 disques pour faire du RAID 10.

Il offre :

— *fiabilité* : tant qu'il reste au moins un disque dans chaque grappe, aucune donnée n'est perdue.
— *performance* : les lectures/écritures sont réparties sur plusieurs disques.

En termes d'optimisation de stockage : avec 4 disques de 10 Go, on a un espace disponible de 20 Go.

4.1.4 RAID 5 — Beaucoup de calculs



Il faut au moins 3 disques. C'est une techniques plus complexe mathématiquement et répartit les données sur tous les disques en rajoutant des *blocs de parité*. Chaque bloc écrit sur un disque est donc :

— soit un bloc de parité : si on le perd, on ne perd pas de données et on peut recalculer ce bloc à partir des blocs de données des autres disques.
— soit un bloc de données : si on le perd, on peut le retrouver à partir des blocs de données restants et du bloc de parité.

Le RAID 5 améliore les performances puisque les lectures/écritures sont réparties sur plusieurs disques. Il apporte une certaine fiabilité puisqu'on peut perdre un disque sans perte de données. Par rapport au RAID 10, il optimise le stockage puisqu'avec 4 disques de 10 Go, on a un volume utile de 30 Go.

Par contre, avec du RAID 5, peu importe de le nombre total de disques, on ne peut pas se permettre de perdre plus d'un disque. Aussi, le calcul des blocs de parité représente une charge processeur non négligeable.

Il existe d'autres niveaux de RAID mais ils sont moins courants. En tout cas, avec le RAID, il s'agit de compromis entre fiabilité, performance et coût.

4.2 La pratique

Il y a plusieurs façons de gérer un volume RAID :

— **RAID matériel** — Certains serveurs sont équipés d'un contrôleur matériel qui gère le RAID. Le réglage se fait au niveau du BIOS et les performances sont très bonnes. L'OS voit le volume RAID comme un disque dur.
— **fake-RAID ou RAID pseudo-matériel** — Version bon marché du RAID matériel. C'est une puce qui gère le RAID en utilisant le processeur et la mémoire de l'ordinateur. Il vaut mieux éviter.
— **RAID logiciel** — Le RAID est géré par l'OS. Cette méthode est utilisable sur tout matériel mais elle consomme une partie des ressources système.

4.2.1 Mise en place d'un RAID 1 pour sécuriser les données

Avec 3 disques on peut mettre en place un RAID 1 :

1. un disque de données
2. un disque de copie des données
3. un *hot spare*, c'est-à-dire un disque de rechange à chaud qui pourra automatiquement prendre la place d'un des 2 disques précédents en cas de défaillance.

Pour créer le volume RAID :

```
# mdadm --create /dev/md0 --level=raid1 --raid-devices=2 /dev/sdb /dev/sdc
    --space-devices=1 /dev/sdd
```

Cela crée le périphérique `/dev/md0` (les volumes RAID ont des noms de type `/dev/md*`) en tant que RAID 1. Il y a 2 disques en RAID et un disque *spare*.

Il vaut mieux fixer les paramètres du RAID pour éviter que le périphérique ne change de nom au prochain redémarrage. Pour cela, ajouter la ligne suivante dans le fichier `/etc/mdadm/mdadm.conf` après le commentaire `# definitions of existing MD arrays` :

```
ARRAY /dev/md0 level=raid1 num-devices=2 spares=1 UUID=<uuid_du_periphérique
    > devices=/dev/sdb,/dev/sdc,/dev/sdd
```

Pour trouver le `UUID` et d'autres détails sur le RAID :

```
# mdadm --query --detail /dev/md0
```

Le système de démarrage a une copie du fichier `/etc/mdadm/mdadm.conf`. Pour que nos changements soient pris en compte, il faut taper la commande suivante :

```
# update-initramfs -u
```

On peut ensuite formater et utiliser le volume RAID comme n'importe quelle partition :

```
# mkfs -t ext4 /dev/md0
# mount /dev/md0 /var/data
# touch /var/data/un_fichier_sur_raid
```

4.2.2 Reconstuire un RAID suite à un défaut de disque

En cas de perte d'un disque, le RAID permet au système de fonctionner sur la copie valide. Il faut quand même vite remplacer le disque défaillant avant de perdre la dernière copie de données.

Dans l'exemple précédent, on a prévu un disque de rechange, donc le processus sera automatisé, mais il faudra quand même penser à ajouter un nouveau disque de rechange.

Pour simuler une perte du disque `/dev/sdb` et vérifier que le disque `/dev/sdd` prend bien le relai :

```
# mdadm --manage /dev/md0 --fail /dev/sdb
# mdadm --query --detail /dev/md
```

Les données sont copiées du disque restant sur le nouveau disque. C'est une reconstruction du RAID. Elle peut être assez longue s'il y a beaucoup de données à copier. On peut aussi suivre les différentes étapes du processus dans le fichier de log `/var/log/syslog`.

Il faut maintenant retirer le disque défectueux du RAID :

```
# mdadm --manage /dev/md0 --remove /dev/sdb
```

Ensuite on peut changer physiquement le disque. Certains contrôleurs disque permettre de le faire à chaud, d'autres nécessitent d'éteindre la machine. Une fois le disque changé, on réintègre le nouveau disque au volume RAID :

```
# mdadm --manage /dev/md0 --add /dev/sdb
```

Chapitre 5

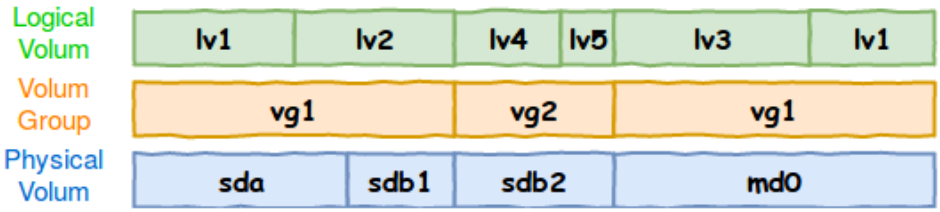
LVM

5.1 Principe

Avec le partitionnement normal, il est dangereux de redimensionner les partitions et d'en ajouter. Si on crée de nouvelles partitions, il faut répartir les fichiers manuellement avec des points de montage supplémentaires.

LVM répond à ces problématiques. Il est basé sur 3 niveaux d'abstraction :

1. **Volumes physiques** (Physical Volume (PV)) — Un volume physique peut être un disque entier, une partition ou un volume Redundant Array of Independant Disks (RAID). On *marque* un PV pour pouvoir l'utiliser avec LVM.
   Pour utiliser LVM avec RAID :
   ```
   # umount /dev/md0
   # pvcreate /dev/md0
   ```
   Pour voir les informations sur les PV, on utilise la commande `pvdisplay`.
2. **Groupes de volumes** (Volume Group (VG)) — On regroupe les PV dans des VG. Pour créer un VG appelé `raid-volume` qui ne contient que notre RAID. On peut plus tard ajouter dynamiquement n'importe quel PV à ce groupe.
   ```
   # vgcreate raid-volume /dev/md0
   ```
   Comme avec les PV, on peut voir les informations sur les VG avec `vgdisplay`.
3. **Volumes logiques** (Logical Volume (LV)) — On découpe les VG en LV. C'est l'équivalent LVM des partitions.
   Pour créer un LV `data1` de 800 Mo et un LV `data2` de 200 Mo sur notre VG :
   ```
   # lvcreate --name data1 --size 800M raid-volume
   # lvcreate --name data2 --size 200M raid-volume
   ```
   Encore une fois, on peut voir les informations sur les LV avec `lvdisplay`.



Les commandes ont une forme dépendant de s'ils s'appliquent sur des PV, des VG ou des LV :

- `{pv,vg,lv}create`
- `{pv,vg,lv}display`
- `{pv,vg,lv}remove`

5.2 Gestion des LV

Les LV fonctionnent comme des partitions. Pour chaque LV, LVM crée un fichier périphérique dans `/dev` sous la forme `/dev/<groupe_de_volume>/<volume_logique>`.

Il faut les formater et les monter :

```
# mkfs -t ext4 /dev/raid-volume/data1
# mount -t ext4 /dev/raid-volume/data1 /var/data1
# mkfs -t ext4 /dev/raid-volume/data2
# mkdir /var/data2
# mount -t ext4 /dev/raid-volume/data2 /var/data2
```

Si au bout d'un certain temps on veut modifier la taille de `/var/data1`, pour lui donner 600 Mo, avec LVM ce sera plus facile qu'avec des partitions classiques.

On a le choix entre :

- réduire la partition `/var/data1` et utiliser l'espace libérer sur ce volume
- rajouter un disque et construire un nouveau RAID 1 avec le disque *spare* du RAID actuel.

5.2.1 Redimentionnement de volumes LVM

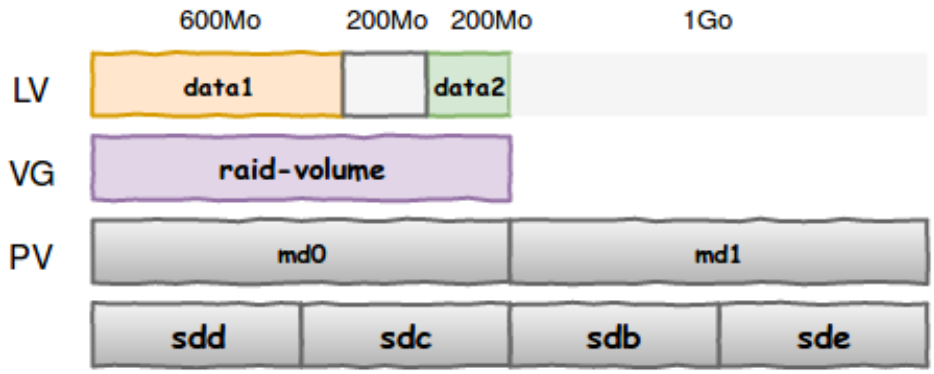
Pour réduire `/var/data1` de 800 à 600 Mo, il faut d'abord réduire le système de fichiers, et ensuite réduire le LV. Avec LVM, on peut regrouper les deux étapes en une seule commande. Deux prérequis cependant : il faut bien sûr avoir suffisamment d'espace disponible et le système de fichier doit supporter l'opération de réduction, ce qui est le cas de `ext4`.

```
# lvreduce --size 600M --resizefs /dev/raid-volume/data1
```

Avec `resizefs`, `lvreduce` se charge de démonter la partition, de lancer un `e2fsch` et un `resize2fs`. Ensuite il réduit la taille du LV et remonte la partition.

5.2.2 Ajout de volumes à LVM de manière flexible

Imaginons que l'on ait un autre volume RAID, `/dev/md1` et nos 200 Mo libérés au milieu de notre VG `raid-volume` :



Avec du partionnement classique, on ne peut pas déplacer les 200 Mo, et on aurait plusieurs partitions, avec une gestion de la répartition des données compliquée.

Avec LVM, on ne s'occupe pas de savoir où sont les données.

On va utiliser le deuxième RAID pour agrandir notre LV :

— On marque le RAID comme PV :

```
# pvcreate /dev/md1
```

— On ajoute ce PV au VG `raid-volume` :

```
# vgextend raid-volume /dev/md1
```

— On étend le LV à la taille souhaitée, sans se soucier de savoir où sont les données.

```
# lvextend --size +1200M --resizefs /dev/raid-volume/data2
```

5.3 Créer des snapshots LVM

En français, *instantané*. C'est une image de l'état du LV à un instant `t`. Il est stocké sur le VG donc il faut un peu d'espace disponible.

```
# lvreduce --size -300M --resizefs /dev/raid-volume/data2
```

On peut créer des fichiers et ensuite créer un snapshot appelé `backup_yyyymmdd` du LV `/dev/raid-volume/data2` en lui allouant 100 Mo :

```
# lvcreate --snapshot --name backup_yyyymmdd --size 100M /dev/raid-volume/
    data2
```

On peut voir les snapshots avec `lvdisplay`.

Lors de sa création, un snapshot a une taille de 0. En fait on a marqué le moment à partir duquel on va enregistrer les changements. Quand on continue de travailler, d'ajouter, modifier et supprimer des fichiers sur le LV, tous les changements sont enregistrés dans le snapshot qui va grossir.

Si le snapshot atteint la taille qu'on lui avait allouée, il devient inutilisable. On peut alors augmenter cette taille avec `lvresize`.

On utilise le snapshot comme n'importe quel LV. Pour récupérer des fichiers par exemple, on monte le volume et on retrouve les fichiers dans l'états où ils étaient au moment où on a fait le snapshot.

On peut bien sûr supprimer le snapshot si on en n'a plus besoin :

```
# lvremove /dev/raid-volume/backup_yyyymmdd
```

Chapitre 6

Sauvegarde des données

Faire du LVM et du RAID n'empêche pas de devoir faire des sauvegardes régulières. Il y a deux types de sauvegardes :

1. les sauvegardes synchrones
2. les sauvegardes asynchrones

6.1 Les sauvegardes synchrones

Les sauvegardes synchrones se font en *temps réel* comme pour le RAID 1. À chaque instant, il y a un double des données sur un autre support.

Pour avoir du "vrai" temps réel, il faut écrire les données sur tous les supports en même temps. On peut alors lire les données aléatoirement sur un support ou un autre.

Il peut être plus simple de faire toutes les écritures sur un support et de synchroniser ou de répliquer les données immédiatement sur d'autres supports. Dans ce cas, on a du "presque" temps réel.

L'avantage des sauvegardes synchrones, c'est qu'on a toujours une copie fraîche des données. Par contre, en cas de bug ou de mauvaise manipulation, on perd toutes les données en même temps. Si on efface par erreur un fichier dans un volume RAID 1, il est effacé sur les deux disques en même temps. Pour éviter ça, on réalise des sauvegardes *asynchrones*.

6.2 Les sauvegardes asynchrones

Ce sont des sauvegardes qu'on fait à intervalle régulier. On peut revenir dans le passé mais on peut aussi perdre tous les changements depuis la dernière sauvegarde. La durée de l'intervalle est donc une affaire de compromis.

Comme les erreurs sont inévitables, il faudra toujours avoir des sauvegardes asynchrones, mais idéalement il faut les deux. En général, on met en place une rotation des sauvegardes.

6.3 Réduction de la taille des sauvegardes

6.3.1 La compression

La compression est la première chose qui vient à l'esprit pour gagner de la place. Pour des fichiers qui sont déjà compressés, comme les `jpeg`, on n'y gagne pas grand chose, mais pour les fichiers texte on peut faire un grand gain de place. Il n'y a pas vraiment de raison de renoncer à la compression.

6.3.2 Les sauvegardes différentielles

Quand on fait une sauvegarde complète des données, on conserve beaucoup de données en double pour rien. Pour un système de 1 Go, si on ne modifie pas de données mais que tous les jours on ajoute 100 Mo de données et qu'on fait une sauvegarde quotidienne, au bout d'une semaine on obtiendrait 9,1 Go de sauvegardes.

Il serait donc plus économe de faire une sauvegarde complète le lundi, puis de faire chaque jour de la semaine une sauvegarde de ce qui a changé seulement. Ce sont des sauvegarde *différentielles*.

Ne pas oublier, lors de la restauration de la sauvegarde le jeudi, de disposer de la sauvegarde du jeudi mais aussi du lundi.

6.3.3 Les sauvegardes incrémentielles

Les sauvegardes différentielles ne sont pas encore ce qu'il y a de plus efficace. Dans l'exemple précédent, les sauvegardes des jours de la semaine contiennent encore beaucoup de données en double. Le plus efficace est de ne sauvegarder chaque jour seulement les données qui ont changé depuis la dernière sauvegarde. Ce sont des sauvegardes *incrémentielles*. Pour restaurer une sauvegarde du jeudi, il faut alors les sauvegardes du lundi, du mardi, du mercredi et du jeudi.

Ce système limite le plus la taille des sauvegardes mais il est plus complexe à gérer.

6.4 Restaurations

Ne pas pouvoir restaurer une sauvegarde rend la sauvegarde inutile. Il faut donc avoir une stratégie de restauration.

Chapitre 7

Partage de fichiers

7.1 NFS

7.1.1 Présentation

NFS ne fonctionne qu'entre machines UNIX. Pour un partage avec des machines Windows, Samba (voir 7.2) est plus approprié.

NFS est conçu pour un réseau local. On ne devrait pas l'utiliser pour partager des fichiers sur internet, où d'autres solutions seraient plus adaptées.

Le port par défaut alloué à NFSv4 est le port TCP 2049 entrant.

On va ici configurer les deux partitions configurées avec RAID et LVM (voir 5.2) :

1. une pour les clients et leurs fichiers personnels (extension de leur `/home` en ligne)
2. une pour un répertoire partagé à tous les utilisateurs

7.1.2 Configuration du serveur

On commence par créer une arborescence de partage `/export`. Dans ce répertoire seront montés tous les répertoires à partager. Ce n'est pas obligatoire mais c'est une bonne pratique.

```
# mkdir -p /export/{home,shared}
```

Nous avons les deux LV `/var/data1` et `/var/data2`. On monte `/var/data1` dans `/export/home` et `/var/data2` dans `/export/shared`. On va monter avec une option *bind*, qui permet de monter plusieurs fois un système de fichiers. `/var/data1` et `/export/home` pointeront vers les mêmes données. On rajoute pour ça dans le fichier `/ect/fstab` :

```
/var/data1 /export/home none rw,bind 0 0
/var/data2 /export/shared none rw,bind 0 0
```

Une fois que l'arborescence est créée, on peut installer le serveur NFS :

```
# apt install nfs-server
```

Les partages NFS sont définis dans le fichier `/etc/exports`. On rajoute les lignes suivantes :

```
/export        192.168.0.0/24(ro,sync,root_squash,no_subtree_check,fsid=0)
/export/home   192.168.0.0/24(rw,sync,root_squash,no_subtree_check)
/export/shared 192.168.0.0/24(rw,sync,all_squash,no_subtree_check)
```

Le réseaux est celui avec lequel on veut partager nos fichiers. On peut en indiquer plusieurs en les séparant par des espaces.

Les options utilisées sont :

- `ro` — Partage en lecture seule.
- `rw` — Partage en lecture-écriture.
- `sync` — Les opérations sont effectuées les unes à la suite des autres. C'est conseillé en `rw`, mais en `ro` on peut utiliser `async`, le contraire, pour indiquer qu'une lecture peut commencer avant la fin d'une autre lecture.
- `no_subtree_check` — Contraire de `subtree_check`, qui fait une vérification supplémentaire pour vérifier qu'un fichier est bien présent dans l'arborescence. C'était l'option par défaut mais elle cause plus de problèmes qu'elle n'en résout. `no_subtree_check` est recommandé sauf pour les partages en `ro` où les renommages de fichiers sont rares.
- `fsid=0` — Identifie un système de fichier grâce à un code. Utile pour des partages qui ne sont pas identifiés par des périphériques dans `/dev`. Le chiffre 0 est un code spécial indiquant que c'est la racine du répertoire de partage.
- `root_squash` — Option de sécurité. Si on partage des fichiers appartenant à `root`, le `root` des clients sera propriétaire aussi. Avec `root_squash`, l'utilisateur `root` des clients NFS sera connu comme l'utilisateur `nobody`, avec l'UID 65534, du groupe `nogroup` avec le GID 65534.
- `all_squash` — Comme pour `root_squash` mais l'utilisateur `nobody` sera appliqué sur tous les utilisateurs avec le groupe `nogroup`.

On peut valider la configuration et vérifier les partages actifs :

```
# exportfs -r
# showmount -e localhost
```

7.1.3 Configuration du client

Sur le client, il faut monter le partage réseau. Pour cela il faut installer le paquet `nfs-common` et créer le point de montage.

```
# apt install nfs-common
# mkdir /mnt/reseau
```

Puis on édite `/etc/fstab` pour monter le partage :

```
vm-serveur:/ /mnt/reseau nfs4 rw,hard,intr,_netdev 0 0
```

Cela indique qu'on veut monter la racine du partage réseau dans `/mnt/reseau` avec le système de fichiers `nfs4` et les options :

- `rw` — Montage en lecture seule.
- `hard` — Montage traité comme un périphérique standard. Il est permanent et ne sera pas démonté automatiquement au bout d'un certain temps d'inactivité. Le contraire est `soft`.

— `intr` — Interruptible : une opération peut être interrompue en attente de réponse du serveur. C'est conseillé puisqu'il peut y avoir des problèmes pour joindre le serveur et on ne voudrait pas que le client ne reste coincé.
— `_netdev` — Montage réalisé une fois le réseau opérationnel.

On peut monter le partage avec `mount -a`. Dans `/mnt/reseau`, on trouvera les deux répertoires `home` et `shared`, mais on n'aura pas les droits d'écriture, même avec `sudo`.

Pour que le client puisse écrire dans son propre `home`, il faudra corriger certaines choses sur le serveur :

```
# mkdir /export/home/etienne
# chown 1000:1000 /export/home/etienne
# chown nobody:nogroup /export/share
```

Le client peut maintenant écrire dans `/mnt/reseau/home/etienne`.

NFS respecte les droits UNIX. Cela veut dire que sur toutes les machines du réseau, les utilisateurs doivent avoir les mêmes UID. Si `etienne` a l'UID 1000 et que `marie` a aussi l'UID 1000, alors `marie` pourra accéder aux fichiers d'`etienne`.

7.2 Samba

7.2.1 Présentation

Windows utilise son propre protocole de partage de fichiers : SMB/CIFS.

Samba est un logiciel qui permet aux systèmes UNIX d'utiliser également ce protocole. Il est plus facile d'utiliser Samba sous Linux que de configurer NFS sous Windows. Dans des environnements hétérogènes, on privilégie donc Samba.

Dans Windows, il y a deux types de réseaux :

1. **Les groupes de travail (`WORKGROUP`)** — Réseaux d'ordinateurs sans gestion centralisée. Pour se connecter à un autre ordinateur du réseau, il faut que tous les utilisateurs existent sur tous les ordinateurs du réseaux avec les mêmes identifiants.
2. **Les domaines de type NT4 ou Active Directory (AD)** — Les réseaux d'ordinateurs sont gérés par une autorité centrale appelée contrôleur de domaine (voir 2.1).

Samba peut faire contrôleur de domaine NT4 ou AD, ou bien simplement serveur de partage de fichiers pour un groupe de travail.

Faisons comme précédemment avec NFS (voir 7.1) :

— un répertoire personnel avec authentification
— un répertoire commun sans authentification

7.2.2 Configuration du serveur

Il faut d'abord installer le serveur samba :

```
# apt install samba
```

Pour configurer le partage, on édite le fichier `/etc/samba/smb.conf`. La configuration globale commence par `[global]`, puis la configuration de chaque partage est dans une section dont le nom est entre crochets `[]`.

— On peut laisser le nom `WORKGROUP` par défaut si les clients Windows n'ont pas changé leur groupe de travail.

```
[global]
workgroup = WORKGROUP
```

— Vu que Samba gère le partage sur un réseau local, il vaut mieux n'écouter que sur l'interface locale.

```
interfaces = 127.0.0.0/8 enp0s3
bind interfaces only = yes
```

— On configure Samba comme un serveur indépendant, pas comme un contrôleur de domaine. Ensuite on définit qu'un échec d'authentification par un utilisateur inconnu connectera l'utilisateur avec le compte `invité`. On peut définir sur chaque partage si on autorise le compte `invité` ou non.

```
server role = standalone server
map to guest = bad user
```

— En fin de fichier, on rajoute la configuration des deux partages.

```
[home]
path = /export/home/%u
read only = no
guest ok = no
force create mode = 0660
force directory mode = 2770
valid users = @sambashare

[shared]
path = /export/shared
read only = no
guest ok = yes
force create mode = 0660
force directory mode = 2770
force user = nobody
force group = nogroup
```

Les paramètres utilisés ici sont :

— `path` — Le chemin du répertoire partagé. On peut utiliser des variables, ici `%u` est remplacé par le nom de l'utilisateur.

— `read only` — `no` pour le mettre en lecture-écriture, `yes` pour lecture seule. On peut mettre en lecture seule et configurer des exceptions pour certains utilisateurs avec le paramètre `write list = user1 user2`.

— `guest only` — Accepte ou non les comptes `invité`.

— `force create mode` et `force directory mode` — Les droits donnés respectivement aux nouveaux fichiers et aux nouveaux répertoires.

— `valid users` — Les noms des utilisateurs autorisés à accéder au partage. On peut indiquer les noms de groupes en commençant par `@`. Sur Ubuntu, le comte `sambashare` est créé automatiquement à l'installation de Samba.

— `force user` — Va en quelque sorte “convertir” l’utilisateur et le groupe lors de la connexion, comme pour NFS. Tous les utilisateurs qui accèdent à `/export/shared` seront reconnus comme `nobody:nogroup`.
— `force group` —

Il faut redémarrer le service `smbd`. Puis rajouter son utilisateur dans le groupe `sambashare`.

Il faut maintenant enregistrer dans Samba les comptes utilisateurs :

```
# pdbedit -a etienne
# pdbedit -a nobody
```

On va créer un nouvel utilisateur `sambatest` avec son partage `[home]` :

```
# useradd -M -s /bin/false -g sambashare sambatest
# pdbedit -a sambatest
# cd /export/home
# mkdir sambatest
# chown sambatest sambatest
```

On peut aussi lister les utilisateurs de la base Samba :

```
# pdbedit -L
```

Ou supprimer un utilisateur de la base Samba :

```
# pdbedit -x -u sambatest
```

7.2.3 Configuration du client

Un client Linux a besoin de certains paquets :

```
# apt install smbclient cifs-utils
```

On a deux moyens d’accéder aux partages :

— le client `smbclient` en ligne de commandes :

```
# smbclient -U etienne //vm-serveur/home
```

— en montant les partages réseaux. Il faut créer les points de montage sur le client avant de monter :

```
# mkdir -p /mnt/samba/{home,shared}
# mount -t cifs -o rw,guest //vm-serveur/shared /mnt/samba/shared
# mount -t cifs -o rw,user=etienne,uid=etienne,gid=etienne //vm-
    serveur/home/ /mnt/samba/home
```

Pour automatiser le montage au démarrage du système, il faut modifier le fichier `/etc/fstab` :

```
//vm-serveur/shared /mnt/samba/shared cifs rw,guest,_netdev 0 0
//vm-serveur/home /mnt/samba/home cifs rw,uid=etienne,git=etienne,
    credentials=/root/.smbcredentials,_netdev 0 0
```

Les identifiants sont à ajouter au fichier `/root/.smbcredentials` :

```
username=<identifiant>
password=<motdepasse>
```

Ne pas oublier les droits :

```
# chmod 600 /root/.smbcredentials
```

Chapitre 8

Base de données relationnelles — UML

8.1 Le modèle relationnel

Les BD collectent et enregistrent les informations. Avant de mettre en place une BD, il faut la conceptualiser : c'est le *domaine fonctionnel*. Il existe différentes natures de bases de données (BD, ou DataBase (DB) en anglais) :

- **Système de Gestion de Base de Données (SGBD) relationnel** — Les données sont représentées dans différents tableaux pouvant être reliés entre eux.
- **SGBD NoSQL** — Les données sont organisées dans d'autres structures :
  - clé-valeur : comme un dictionnaire qui à chaque mot (clé) associe une définition (valeur)
  - orienté graphe : associe à chaque élément les éléments liés (comme les amis d'une personne)
  - etc.

Nous allons nous concentrer sur les SGBD Relationnel (SGBD-R), à l'aide d'une approche orientée objet. Un objet représente un concept ou une entité du monde réel.

Pour modéliser le domaine fonctionnel, on utilise un langage spécifique : Unified Modeling Language (UML). UML est destiné à décrire l'organisation d'un système avec une approche orientée objet, et non une BD relationnelle. Mais il peut quand même servir de base à la conception en modélisant le domaine fonctionnel avec le *diagramme de classes* (voir 8.2).

Grâce au diagramme de classes, on décrit les informations à enregistrer, mais aussi leur structuration.

8.1.1 Les notions manipulées en objet

La notion d'objet

Un objet est un élément autonome. Il peut être un élément physique réel (un chien, une voiture...) ou un élément abstrait (un ensemble, une liste...).

Il a des caractéristiques, comme :

- un nom : nom commun unique qui donne le type de l'objet (par exemple voiture)
- des attributs : les propriétés de l'objet (couleur, masse, longueur...)

La notion de classe

Une classe est un modèle abstrait d'un objet. Elle définit les attributs et les opérations de cet objet et sert de gabarit pour créer les objets.

Une classe définit donc les caractéristiques de tous les objets de cette classe (nom, liste des attributs).

La notion d'instance

L'instanciation est la création d'un objet à partir d'une classe. L'objet créé s'appelle une instanciation de la classe.

8.1.2 Les notions manipulées par la BD

Une BD relationnelle est une ensemble de tableaux appelés *tables*. Ces tables correspondent aux classes en orienté objet.

Chaque colonne d'une table représente alors un attribut. Chaque ligne, appelée *tuple* ou *n-uplet* représente une instance, donnant la valeur de chaque attribut de l'objet.

Une BD relationnelle est composée d'un ensemble de tables pouvant être reliées entre elles. Cette manière d'organiser les éléments s'appelle un *modèle relationnel*.

8.1.3 La description des classes

Le diagramme de classes suivant :

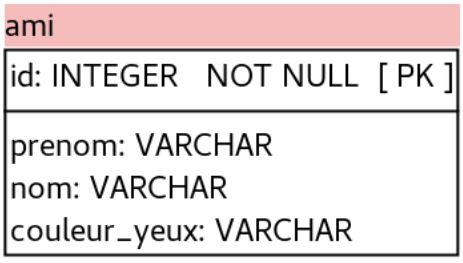
peut se traduire en BD comme ceci :

nom	cuisine	adresse	note
Chez Luigi	Italien	1 Rue d'Issy	4
Burger quid	Fast food	2 Chai Watt	1
La cuisine au beurre	Terroir	Le Faubourg Ville	3
Au lit on dort	Asiatique	10B Rue Seully	3
La mère Michelle	Fruits de mer	Port Salle Hutte	5

Pour passer de l'un à l'autre, on utilise un autre type de schéma : le Modèle Physique de Données (MPD).

8.1.4 Le Modèle Physique de Données

Voici une réprésentation de la table `Ami` dans un Modèle Physique de Données (MPD) :



Cela ressemble beaucoup au diagramme de classes, avec 3 parties :

1. le nom de la table
2. une partie pour l'`id` (qu'on verra plus tard)
3. les attributs (colonnes de la table)

Le type de données de chaque attribut est précisé (`VARCHAR` correspond à du texte par exemple).

8.1.5 La démarche

1. On commence par identifier les concepts ou entités de notre domaine fonctionnel. Cela nous donne les *classes*.
2. Pour chaque classe, on identifie les informations à enregistrer, avec leur type (texte, nombre...). Cela nous donne un *diagramme de classes* avec des *attributs*.
3. On crée le MPD à partir du diagramme de classes :
   (a) en créant une *table* pour chaque classe.
   (b) en créant dans chaque table une *colonne* pour chaque attribut de la classe correspondante.

8.1.6 Le nommage

Si les noms que l'on choisit pour nommer les éléments ne sont pas appropriés, le système peut vite devenir incompréhensible. Il est important que les différents membres de l'équipe projet s'entendent sur les noms choisis.

Un diagramme UML permet de discuter du modèle à développer.

Pour trouver des noms pertinents, on peut se demander si le nom est-il trop général ou trop spécifique.

8.1.7 Conventions de nommage

Il n'y a pas de règle absolue, mais voici les conventions :

- **En UML** :
  - **Pour les classes** :
    - au singulier
    - avec des lettres sans accent
    - utilisant le `PascalCase` (comme `PierrePrecieuse`)
  - **Pour les attributs** :
    - au singulier
    - avec des lettres sans accent
    - utilisant le `camelCase` (comme `couleurYeux`)
- **Dans le MPD** :
  Pour le nom des tables et des attributs :
  - au singulier
  - avec des lettres sans accent
  - utilisant le `underscore_case` (comme `pierre_precieuse`)

8.2 Le diagramme de classes

On a parlé de classes isolées, mais un domaine fonctionnel est souvent composé d'un ensemble de classes liées entre elles. Après avoir défini les classes, on doit donc se demander comment elles forment un système cohérent.

Par exemple, dans le cadre d'un sytème de vente en ligne, on peut se poser les questions suivantes :

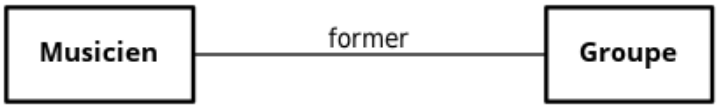
— Qu'est-ce qui est vendu ?
— Est-ce qu'on peut classer les articles en plusieurs catégories ?
— Est-ce qu'on doit prendre en compte le stock des articles ?
— Un client peut-il avoir des bons d'achat sur son compte ?

8.2.1 Décrire les relations

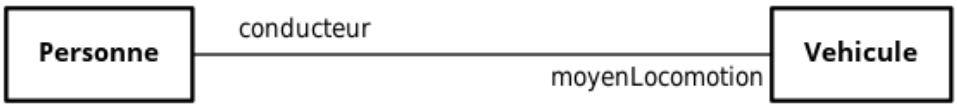
UML permet de décrire les relations entre les classes ainsi que les contraintes que ces relations subissent.

En UML, la relation entre deux classes s'appelle une *association*.

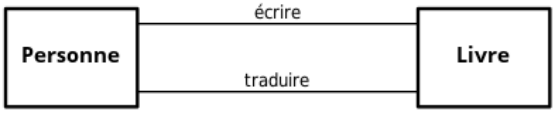
Pour modéliser une association entre deux classes, on trace un trait entre elles. On peut, si besoin, préciser un nom pour cette association au milieu du trait :



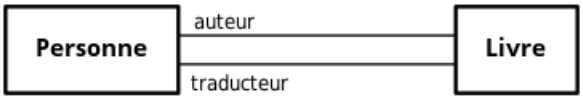
On peut aussi donner un rôle à chaque classe en le notant du côté des classes concernées :



Il est possible qu'il y ait plusieurs associations entre deux classes. Dans ce cas, il devient impératif de noter la relation :



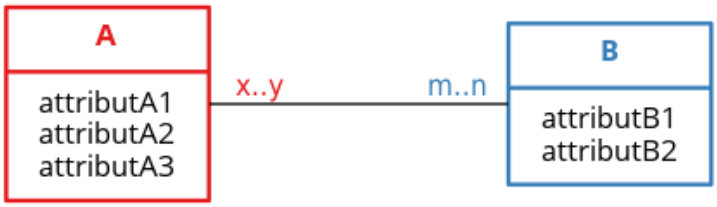
ou bien :



8.2.2 Les multiplicités

Les multiplicités servent à apposer des contraintes numériques sur l'association, c'est-à-dire combien d'instances d'une classe peuvent être liées à une instance de la classe de l'autre côté de l'association. Elles bornent les *cardinalités* des instances liées entre elles.

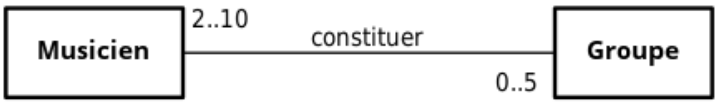
Avec une classe `Groupe` et une `Musicien`, on pourrait définir combien de musiciens minimum et maximum peuvent composer un groupe. On peut aussi définir dans combien de groupes différents un même musicien peut jouer.



Cela donne :

— combien de `A` pour un `B` : entre `x` et `y`
— combien de `B` pour un `A` : entre `m` et `n`

Avec les musiciens et les groupes :



— Un musicien joue dans 5 groupes maximum et peut ne jouer dans aucun groupe.
— Un groupe est constitué d'au moins 2 musiciens et 10 au maximum.

Il n'y a pas de notion d'historique dans les associations. À tout moment, un musicien doit respecter la cardinalité précédente, mais il peut avoir joué dans 50 groupes différents au cours de sa vie. Cela veut dire que nous avons les liens pour les groupes actuels mais nous n'avons aucune information sur les associations antérieures.

Il existe des abréviations pour certaines plages de cardinalités :

Multiplicité	Abréviation	Cardinalités
`0..0`	`0`	Aucune instance
`0..1`		Aucune ou une seule instance
`1..1`	`1`	Exactement une instance
`0..*`	`*`	Aucune, une ou plusieurs instances (aucune limite)
`1..*`		Au moins une instance (aucune limite)
`x..x`	`x`	Exactement `x` instance(s)
`m..n`		Au moins `m` et au plus `n` instances

8.2.3 Les 3 catégories d'association

1. **one-to-one (un à un)** — Un véhicule conduit par un conducteur.
2. **one-to-many (un à plusieurs)** ou **many-to-one (plusieurs à un)** Plusieurs articles dans un journal.
3. **many-to-many (plusieurs à plusieurs)** Plusieurs musiciens dans plusieurs groupes.

C'est en fonction de ces catégories que l'on traduit les associations en relations dans la BD.

8.3 Les clés primaires

En BD, à chaque *instance* de classe correspond un *tuple* (une ligne) dans la table. Si on veut relier les tuples, il faut pouvoir les identifier de manière unique. On ne peut pas identifier une entrée de manière sûre par un de ses attributs, puisqu'il risque d'y avoir des doublons.

Il vaut mieux les identifier avec un attribut supplémentaire : `id`. Parfois on ajoute le nom de la table en préfixe : `voiture_id`.

Lors de la création de la table, l'`id` sera indiqué comment l'identifiant des tuples. On parle alors de *clé primaire*, ou Primary Key (PK) en anglais. C'est l'indication de *clé primaire* qui permet à la BD de s'assurer qu'il n'y ait aucun doublon dans cette colonne. Il est également possible de remplir la valeur de la clé primaire automatiquement en la qualifiant d'*auto-incrémentée*.

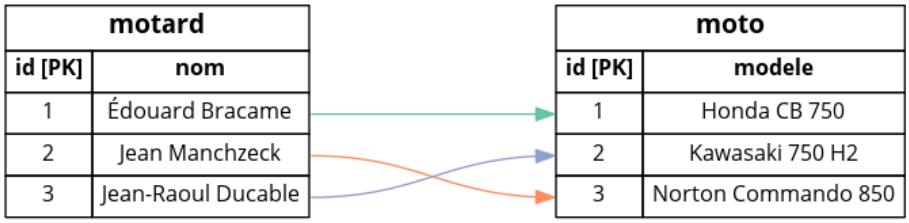
Dans le MPD, la clé primaire porte l'indication `PK`.

Pour que le modèle relationnel soit cohérant il faut donc que la clé primaire soit unique, mais aussi qu'elle ne change pas.

8.4 Les clés étrangères

Pour créer les relations entre les tables avec les clés primaires, il faut une référence vers le tuple lié dans un autre attribut : la *clé étrangère* (Primary Key (PK) en anglais).

La *clé étrangère* d'un tuple est donc la valeur de la *clé primaire* du tuple lié.



Troisième partie

Sécurité

Chapitre 1

Introduction

1.1 Quoi protéger ?

- **Les données** — Les données ont 3 caractéristiques essentielles :
  - *Le secret* : ne pas diffuser les données vers l'extérieur.
  - *L'intégrité* : ne pas pouvoir modifier les données quand on est étranger à l'entreprise.
  - *La disponibilité* : pouvoir utiliser les données au moment opportun.
- **Les ressources informatiques**
  - machines serveurs
  - postes utilisateurs
  - équipements réseau
- **La réputation**
  - perte de confiance des clients
  - mauvaise publicité des médias
  - problème avec la justice après vol d'identité

1.2 Contre quoi se protéger ?

- Attaques pour s'introduire (grâal).
- Attaques de MitM
  - ARP spoofing
- Attaques pour DoS
  - SYN flooding
  - Smurf (broadcast dirigé)
    Charlie prend comme adresse IP source l'Afpa pour faire un broadcast sur le réseau de HP. Maintenant il fait ça plein de fois, HP est en déni de service parce qu'il passe son temps à répondre. Dans les routeurs, il y a `no ip directed broadcast`.
  - Distributed Denial of Service (DDOS)
    Prend d'autres serveurs comme relai pour faire des attaques groupées.
- Attaques par erreur humaine
  - STUXNET
    Virus sur clé USB qui a détruit des centrifugeuses nucléaires iraniennes.

1.3 Contre qui se protéger ?

La majorité (60%) des attaques sont internes à l'entreprise (salariés ou ex-salariés). Tout n'est pas confidentiel, mais beaucoup d'informations sont trop facilement accessibles.

1.4 Comment se protéger

1.4.1 Politique de sécurité globale

Elle doit adresser :

— **Le comportement des utilisateurs** — Procédures, choix de mots de passe, confidentialité des informations, etc.
— **Les systèmes** — Configuration, mises à jour, outils de filtrage, de suivi et de contrôle d'identité.
— **L'architecture du réseau** — Définition de zones de sécurité, mise en place d'équipements firewall, etc.
— **Les communications** — Procédures d'authentification et de chiffrement des données (VPN).

Elle doit intégrer :

— **Une phase de sensibilisation** — Permet de justifier les contraintes imposées par les procédures mises en place.
— **Une procédure de gestion des mots de passe** — Choix des mots de passe, période de renouvellement, fréquences des audits.
— **Des règles d'exploitation** — Définition des droits des utilisateurs, des procédures de gestion des modifications, du suivi d'activité et de la vérification.

1.4.2 Veille technologique

L'évolution est extrêmement rapide. On découvre environ 5 nouvelles failles par jour sur les OS, applications, équipements de filtrage, etc. De trouve de nouvelles techniques de collecte, d'identification d'OS, de scan...

Que ce soit l'activité de la plateforme de sécurisation ou de la veille technologique, c'est un travail quotidien.

1.4.3 Filtrage IP

Beaucoup de services passent par HTTP : si on bloque le port 80 on laisse passer plusieurs attaquent utilisant le web comme point d'entrée.

On ne peut pas rester sur la couche Transport.

Il y a deux types de filtrages IP :

1. **Stateless (sans état)** — Le filtrage dans le sens "sortie" peut être constant, cela ne pose pas de problème. Par contre, la réponse va être dans le sens "entrée". En stateless, le trou laissé dans le firewall va rester même quand il n'y a plus de communication.

2. **Stateful (avec état)** — Même exemple que précédemment, mais en stateful, après une communication, la règle dans le firewall sera enlevée. Le firewall est donc par défaut fermé.

   Il s'agit de créer une règle temporaire suite à une règle ouverte en sortie. Exemple : PC1 est dans mon réseau et PC2 quelque part sur internet. Une règle de sortie (PC1 vers PC2) existe mais aucune règle n'autorise PC2 à communiquer. Si PC2 initialise une connexion, elle sera bloquée. Par contre si PC1 initialise la connexion, comme une règle sortante existe, elle sera autorisée. Une règle temporaire sera alors créée pour autoriser PC2 à communiquer, avec un TTL.

   Les règles stateful sont aussi plus précises : elles permettent d'inclure des choses connues seulement au moment de la connexion (comme le port client), et d'ajouter le protocole (ICMP, Telnet, etc.). Elles vont donc autoriser moins de choses, moins longtemps.

Un exemple ou le stateful sera nécessaire, c'est pour les connexions FTP en mode passif (voir 7.2.2). Puisque le port du serveur pour les données est aléatoire, on crée des règles dynamiques lors de la connexion. En stateless, on serait obligé d'autoriser beaucoup trop de ports.

Chapitre 2

Bonnes pratiques

2.1 Une liste non exhaustive

— Pas de shell en cours d'exécution sur les services (`/bin/nologin`).
— Mises à jour avec les paquets de la distribution.
— Pas de dépôt tiers.
— Pas de mot de passe de groupes.
— Désinstaller les services réseaux non utilisés.
— Les services doivent n'écouter que sur les interfaces et IP nécessaires.
— Ne pas laisser apparaître de numéro de version des logiciels.
— `root` doit être inaccessible à tous (pas de groupe `sudo`).
— Ne pas se connecter en `root` et l'utiliser au minimum (avec `su` et `sudo` donc).
— Faire des logs des connexions et les contrôler.
— Créer des comptes `admin` limités, mais attention à quoi ils ont accès (`cat` très dangereux).
— Faire attention au `PATH` : pas de `.` ou de `:` au début.
— Penser aux installation sandbox (`flatpak`, `snap`, `appimage`).
— Ne pas se balader sur le web en `root`.

Chapitre 3

La cryptographie

3.1 Cryptographie symétrique

Les algorithmes sont Data Encryption Standard (DES), 3-DES, International Data Encryption Algorithm (IDEA), Rivest Cipher (RC)2, RC4...

En cryptographie symétrique, la même clé sert à chiffrer et à déchiffrer les données. Se pose donc le problème de la transmission de la clé secrète.

Son avantage est cependant qu'elle est rapide. Elle est donc souvent utilisée pour chiffrer les paquets IP par exemple.

3.2 Cryptographie asymétrique

Préconisée pour le chiffrement et le déchiffrement de documents. Elle utilise deux clés : une clé *publique* pour chiffrer et une clé *privée* pour déchiffrer.

Comme son nom l'indique, la clé privée doit rester secrète. La clé publique, quant à elle, est divulguée à tout le monde sous forme de certificat.

Les deux clés sont liées, mais la clé privée ne peut pas être déduite de la clé publique en un temps raisonnable.

La cryptographie asymétrique permet d'éviter le partage de secret.

Les algorithmes sont Rivest-Shamir-Adleman (RSA), Digital Signature Algorithm (DSA), ElGamal...

Les algorithmes asymétriques sont 100 à 10 000 fois plus lents que les algorithmes symétriques.

3.3 Fonction de hachage

Une fonction de hachage transforme une chaîne de caractères de longueur quelconque en une chaîne de caractères de longueur fixe. Elle sert à vérifier l'authenticité d'un document.

Les algorithmes sont MD5, Secure Hash Algorithm (SHA)1, SHA2...

La résultante de la fonction de hachage s'appelle l'empreinte.

On passe le document à la fonction de hachage et on compare l'empreinte obtenue avec l'empreinte annoncée par la source du document. Si les deux empreintes sont identiques, l'authenticité du document est avérée.

Chapitre 4

Sécurité Physique

4.1 Les Runlevels UNIX

Un runlevel est l'un des modes dans lequel un système d'exploitation basé sur UNIX fonctionnera. Dans le noyau Linux, il y a 7 niveaux d'exécution (runlevels), de 0 à 6. On ne peut démarrer le système que dans un seul runlevel à la fois. Par défaut, on démarre soit au runlevel 3 (Command Line Interface (CLI)), soit au runlevel 5 (Graphical User Interface (GUI)).

La liste des runlevels par défaut dans les distributions Linux :

0. Arrêt (halt)
1. Single user mode
2. Non utilisé (défini par l'utilisateur)
3. Full multi-user text-mode (CLI)
4. Non utilisé (défini par l'utilisateur)
5. Full multi-user graphical-mode (GUI avec écran de connexion basé sur `X`)
6. Redémarrage

4.2 GRUB2

4.2.1 Rescue Mode (Mode Single User) et Emergency Mode

Avec `systemd`, les runlevels sont remplacés par des cibles (targets). On va pouvoir démarrer en mode de secours ou en mode d'urgence.

Rescue Mode

Le mode de secours est équivalent au mode Single User avec `SysV`. Dans ce mode, tous les systèmes de fichiers locaux sont montés et seuls certains services importants sont démarrés. Aucun service normal (comme les services réseaux) n'est démarré. On l'utilise quand le système ne peut pas démarrer normalement.

Ce mode permet également (et c'est là qu'il devient dangereux) d'effectuer certaines opérations de sauvetage importantes, comme la réinitialisation du mot de passe `root`.

Emergency Mode

Le mode d'urgence ne lance rien, contrairement au rescue mode. Aucun service, aucun point de montage, aucune prise. Ce mode est adapté au débogage.

4.2.2 Utiliser les modes Rescue et Emergency à partir de GRUB

1. On commence par entrer dans le GRUB avec `Esc`.
2. On peut taper `e`
3. **Rescue Mode** :
   On a le choix, à la ligne qui commence par `linux`, entre taper `rescue` arpès `ro` ou ajouter `systemd.unit=rescue.target` à la fin. Dans cette ligne, si le mot `$vt_handoff` existe, il faut le supprimer.
4. **Emergency Mode** :
   À la ligne commençant par `linux`, on ajoute `systemd.unit=emergency.target`. S'il faut monter le système : `# mount -o remount,rw /`

Chapitre 5

Système de fichiers

5.1 Chiffrement d'une partition

— **device mapper crypt** (`dm-crypt`) — Chiffrement transparent des données. Il est possible de spécifier les fichiers ou dossiers que l'on souhaite chiffrer, et certains sytèmes acceptent le chiffrement de fichiers `root`.
— **Linux Unified Key Setup (LUKS)** — Front-end de `dm-crypt`. Peut chiffrer l'intégralité du disque. Supporte l'accès avec différents mots de passe pour de multiples utilisateurs. Peut chiffrer les supports mobiles.
— **cryptsetup** — Manage les volumes LUKS et `dm-crypt`. Il vaut mieux préférer LUKS, à moins de bien connaître `dm-crypt`.
— **shred** — Efface le disque sans récupération possible.

5.2 Les permissions

On modifie le `umask` dans les fichiers `/etc/login.defs` et `/etc/profile`.

Chapitre 6

Les ACL

6.1 Sur Cisco

6.1.1 Présentation et intérêt

Une ACL est une série de commandes qui sont utilisées pour filtrer des paquets en fonction d'informations contenues dans l'en-tête. Par défaut, un routeur n'a pas d'ACL de configuré.

Une ACL utilise une liste séquencielle de *deny* et de *permit*, que l'on appelle des Access Control Entry (ACE).

Voici certains examples où l'utilisation d'ACL est nécessaire :

- **Limiter le trafic pour améliorer la performance du réseau** — Par exemple bloquer le flux vidéo pour alléger le trafic.
- **Contrôler le flux de trafic** — Le trafic d'un protocole de routage peut être limité à certains liens.
- **Fournir un niveau de sécurité de base pour l'accès au réseau** — Certaines parties du réseau peuvent être autorisés à certains utilisateurs.
- **Filtrer le trafic en fonction de son type** — On peut autoriser le trafic e-mail, mais bloquer Telnet.
- **Permettre ou bloquer l'accès au réseau à des hôtes** — Certains protocoles (FTP, HTTP...) peuvent être limités à certains groupes d'utilisateurs.
- **Fournir la priorité à certaines classe de trafic réseau** — La QoS peut prioriser la VoIP.

6.1.2 Les types d'ACL

Le filtrage de paquet contrôle l'accès au réseau en analysant les paquets entrants et sortants. Il peut intervenir à la couche OSI 3 ou 4.

Les routeurs Cisco permettent 2 types d'ACL :

1. **Les ACL standards** — Le filtrage ne s'opère qu'à la couche OSI 3 et utilise l'adresse IP source du paquet seulement.
   Une ACL avec un numéro de 1 à 99 ou de 1300 à 1999 sera une ACL standard :

```
R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255
```

2. **Les ACL étendues** — Le filtrage s'opère à la couche OSI 3 en utilisant l'adresse IP source et/ou destination. Il peut aussi filtrer à la couche 4 en utilisant des ports TCP et UDP, ou bien des informations de protocole pour un contrôle plus fin.
   Une ACL avec un numéro de 100 à 199 ou de 2000 à 2699 sera une ACL étendue :

```
R1(config)# access-list 100 permit tcp 192.168.10.0 0.0.0.255 any eq www
```

6.1.3 Les ACL nommées

On peut également définir une ACL avec un nom. C'est la façon préférée puisqu'elle est plus descriptive et qu'on indique le type (standard ou étendu) de manière explicite.

Pour créer une ACL nommée, il faut ajouter le mot `ip` avant la commande :

```
R1(config)# ip access-list extended FTP-FILTER
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp-data
```

Les règles à suivre pour les ACL nommées sont :

— donner un nom pour identifier la fonction de l'ACL
— les noms peuvent contenir des caractères alphanumériques
— les noms ne peuvent pas contenir d'espace ou de ponctuation
— il est de coutûme d'utiliser du tout majuscule
— les ACE sont ajoutées ou supprimées de dedans l'ACL

6.1.4 Les opérations ACL

Il faut distinguer le flux entrant et le flux sortant. De plus, les ACL n'agissent pas sur les paquets provenant du routeur lui-même.

Enfin, une règle implicite comporte `Deny all`. Par défaut, un paquet qui ne trouve pas de correspondance dans l'ACL sera donc bloqué. Il faut donc au minimum une permission dans l'ACL, sinon le `Deny` implicite bloquera tout le réseau.

Le filtrage entrant

Une ACL entrante filtre les paquets avant qu'ils ne soient routés vers l'interface de sortie. Elle est donc efficace, puisqu'elle évite le calcul non nécessaire de routage sur des paquets qui finiraient pas être rejetés. Si le paquet est accepté, il est ensuite routé.

Le filtrage sortant

Une ACL sortante filtre les paquets après qu'ils aient été routés, peu importe l'interface d'arrivée du paquet. Le paquet est possiblement rejeté après avoir été routé vers son interface de sortie.

6.1.5 Les masques génériques

Les masques génériques sont utilisés pour déterminer quels bits de l'adresse à examiner font une correspondance. Ils sont aussi utilisés par OSPF (voir 5.5.8).

Ils sont l'inverse d'un masque de sous-réseau : la où le masque IPv4 voit le 1 binaire comme une correspondance, le masque générique voit le 0 comme une correspondance.

Pour le calculer, on soustrait le masque de sous-réseau à `255.255.255.255`.

On peut calculer 3 types de masques génériques :

1. **Pour une correspondance avec un hôte :**
   Pour faire correspondre une adresse IPv4 précise, on va utiliser un masque générique de `0.0.0.0`. Cela signifie que tous les bits de l'adresse doivent correspondre pour établir la correspondance.
   La commande pour une telle ACE serait :

   ```
   Router(config)# access-list 10 permit 192.168.1.1 0.0.0.0
   ```

2. **Pour une correspondance avec un sous-réseau IPv4 :**
   Pour faire correspondre un sous-réseau, on utilise le masque générique correspondant au masque de ce sous-réseau (son inverse donc). Si on doit permettre un accès à tous les hôtes du réseau `192.168.1.0/24`, on utilisera le masque générique `0.0.0.255`.
   La commande pour une telle ACE serait :

   ```
   Router(config)# access-list 10 permit 192.168.1.0 0.0.0.255
   ```

3. **Pour une correspondance avec une plage d'adresses IPv4 :**
   Pour une plage d'adresse, le principe est le même que pour un sous-réseau, mais en faisant un sur-réseau (résumé de route). Par exemple, pour autoriser les sous-réseaux `192.168.16.0/24`, `192.168.17.0/24`, ...jusqu'à `192.168.31.0/24`, on utiliserait le masque générique `0.0.15.255`.

Les masques génériques peuvent aussi prendre des mots clé, ce qui permet d'avoir des listes plus lisibles. Il y en a 2 :

1. `host` — Correspond au masque générique `0.0.0.0`.
2. `any` — Correspond au masque générique `255.255.255.255`.

```
Router(config)# access-list 10 permit host 192.168.10.10
Router(config)# access-list 11 permit any
```

6.1.6 Bonnes pratiques de mise en place d'ACL

Une interface ne peut pas accueillir plus de 4 ACL : 2 pour l'IPv4 (en entrée et en sortie) et 2 pour l'IPv6 (en entrée et en sortie).

L'ACL est définie en mode de configuration global, sans direction. C'est au niveau de l'interface que l'on active tel ou tel ACL, et qu'on définit dans quelle direction les ACE se lisent (entrée ou sortie).

L'utilisation des ACL nécessite du soin et de l'attention : une erreur peut être coûteuse en perte de connectivité et en déboggage.

Bonne pratique	Avantage
Baser les ACL sur les politiques de sécurité de l'entreprise	Permet de s'assurer de les avoir implémentées
Écrire ce que l'ACL doit faire	Prévient des problèmes d'accès inattendus
Créer, éditer et sauvegarder les ACL dans un éditeur de texte	Permet de réutiliser les ACL
Documenter les ACL avec la commande `remark`	Aide à comprendre la fonction d'une ACL
Tester les ACL sur un réseau de développement avant de passer en production	Évite les erreurs coûteuses

Où placer les ACL ?

Les ACL étendues devraient être placées le plus près possible de la source du trafic à filtrer. Cela permet de ne pas encombrer le réseau inutilement.

Les ACL standard devraient être placées le plus près possible de la destination. Si l'ACL standard est placé à la source du trafic, le `permit` ou `deny` s'appliquera sur l'adresse source, peu importe où le trafic est destiné. En effet, le type `standard` donne moins de contrôle (puisqu'il ne lit que l'IP source).

Mais le placement de l'ACL et son type dépend de multiples facteurs :

— **L'étendue du contrôle de l'entreprise** — Le placement de l'ACL peut dépendre de si l'entreprise a la main sur le réseau source et le réseau destination.
— **La bande passante des réseaux impliqués** — Pour filtrer du trafic gourmand en bande passante, on peut vouloir filtrer à la source.
— **La facilité de configuration** — L'implémentation d'une ACL peut être plus pratique, même si elle va utiliser de la bande passante inutilement. Filtrer du trafic à la source nécessite de créer d'avantage d'ACL.

6.1.7 Configuration des ACL IPv4 standard

ACL standard numérotée

```
Router(config)# access-list <numéro> {deny | permit | remark <text>} <source> [<
    source_wildcard>] [log]
```

— `<numero>` : C'est le numéro assigné à l'ACL. Les ACL standards vont de 1 à 99 ou de 1300 à 1999.
— `deny` : Le paquet est rejeté si la condition correspond.
— `permit` : Le paquet est accepté si la condition correspond.
— `remark <text>` : Optionnel, ajoute une description à l'ACL. Les remarques sont limitées à 100 caractères.
— `<source>` : Identifie l'adresse réseau ou hôte de la source à filtrer. Pour spécifier tout réseau, on utilise le mot clé *any*. Pour spécifier un hôte, on peut soit utiliser le mot clé *host* avec l'adresse IP, ou simplement l'adresse seule sans le mot clé.
— `[<source_wildcard>]` : Optionnel, correspond au masque générique appliqué à la `<source>`. Si on l'omet, un masque à `0.0.0.0` par défaut est utilisé.

— `[log]` : Optionnel, génère et envoie un message informationnel à chaque correspondance avec une ACE. Le message contiendra l'ACL, la condition, l'adresse source, et le nombre de paquets. Il est généré pour le premier paquet correspondant. Il devrait seulement être implémenté pour le déboggage ou la sécurité.

ACL standard nommée

```
Router(config)# ip access-list standard <nom>
```

Contrairement aux ACL numérotés, cette commande nous fait entrer dans le mode de configuration standard nommé. Ce mode nous permet d'entrer les ACE de cet ACL dans un sous mode de configuration.

```
R1(config)# ip access-list standard NO-ACCESS
R1(config-std-nacl)# ?
Standard Access List configuration commands:
  <1-2147483647> Sequence Number
  default        Set a command to its defaults
  deny           Specify packets to reject
  exit           Exit from access-list configuration mode
  no             Negate a command or set its defaults
  permit         Specify packets to forward
  remark         Access list entry comment
R1(config-std-nacl)#
```

Appliquer une ACL standard

L'ACL est maintenant configuré, mais il doit maintenant être lié à une interface :

```
Router(config-if)# ip access-group {<numéro> | <nom>} {in | out}
```

Pour vérifier si une interface a une ACL appliquée :

```
R1# show ip interface Serial 0/1/0 | include access-list
  Outgoing Common access list is not set
  Outgoing access list is 10
  Inbound Common access list is not set
  Inbound access list is not net
```

Modifier une ACL standard

Il y a deux méthodes pour modifier une ACL : un éditeur de texte, et des numéros de séquence.

Avec un éditeur de texte

Les ACL qui contiennent de multiples ACE devraient être créées dans un éditeur de texte. Imaginons que l'on ait entré une erreur dans la configuration d'un routeur :

```
R1# show run | section access-list
access-list 1 deny 19.168.10.10
access-list 1 permit 192.168.10.0 0.0.0.255
R1#
```

On a bloqué l'hôte `19.168.10.10` au lieu de `192.168.10.10`.

Pour corriger l'erreur :

- copier l'ACL à partir de la `running-config` et la copier dans l'éditeur de texte
- faire les changements nécessaires
- supprimer l'ACL précédemment configurée sur le routeur, sinon coller l'ACL modifié ne fera qu'ajouter à la configuration existante
- coller l'ACL modifié sur le routeur

Avec un numéro de séquence

Une ACE peut aussi être supprimée ou ajoutée avec les numéro de séquence d'ACL. Quand une ACE est entrée, des numéros de séquence lui sont automatiquement assignés. On peut les voir avec `show access-lists`.

Pour modifier une ACL, on utilise la commande `ip access-list standard <numéro>`. Les ACE ne peuvent pas être écrasées en utilisant le même numéro de séquence. L'ACE courante doit donc d'abord être supprimée. On peut ensuite ajouter une nouvelle ACE avec ce même numéro de séquence :

```
R1# conf t
R1(config)# ip access-list standard 1
R1(config-std-nacl)# no 10
R1(config-std-nacl)# 10 deny host 192.168.10.10
R1(config-std-nacl)# end
R1# show access-lists
Standard IP access list 1
    10 deny  192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

Insérer une ACE entre deux autres ACE

Par défaut, quand on ajoute des ACE, elles sont ajoutés à la fin de l'ACL, avec un numéro de séquence croissant de 10 en 10.

Prenons une ACL en exemple :

```
R1# show access-lists
Standard IP access list NO-ACCESS
    10 deny  192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
```

On aimerait aussi interdire l'accès à l'hôte `192.168.10.5`, mais si on ajoute l'ACE correspondant, elle sera ajoutée à la fin, avec le numéro 30. Cette ACE ne sera donc jamais lue, puisque le numéro 20 aurait une correspondance. Il faut ajouter la nouvelle ACE entre les deux existantes, avec par exemple un numéro de séquence de 15 :

```
R1# conf t
R1(config)# ip access-list standard NO-ACCESS
R1(config-std-nacl)# 15 deny 192.168.10.5
R1(config-std-nacl)# end
R1#
```

```
R1# show access-list
Standard IP access list NO-ACCESS
    15 deny  192.168.10.5
    10 deny  192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

On pourrait s'attendre à ce que le numéro 15 apparaisse au milieu, mais l'IOS affiche les hôtes en fonction d'une fonction de hachage.

Statistiques de correspondance des ACL

La commande `show access-lists` affiche des statistiques sur le nombre de correspondances pour chaque ACE. Cependant, l'instruction implicite `deny any` n'est pas affichée, et donc on ne sait pas combien de fois elle a été rencontrée. Pour cela il faut ajouter une ACE `deny any` explicite à la main à la fin de l'ACL.

Pour remettre les compteurs à zéro, on utilise la commande `clear access-list counters`.

```
R1# show access-lists
Standard IP access list NO-ACCESS
    10 deny  192.168.10.10 (20 matches)
    20 permit 192.168.10.0, wildcard bits 0.0.0.255 (64 matches)
R1# clear access-list counters NO-ACCESS
R1# show access-lists
Standard IP access list NO-ACCESS
    10 deny  192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

Sécuriser des ports VTY

La plupart du temps, les ACL filtrent le trafic entrant ou sortant sur une interface. Mais une ACL peut aussi servir à sécuriser l'accès distant en utilisant les lignes VTY.

Pour appliquer une ACL à un VTY :

```
R1(config-line)# access-class {<numéro> | <nom>} {in | out}
```

On utilise rarement `out` puisqu'on veut sécuriser l'accès distant, qui est entrant. Il faut mettre les mêmes restrictions sur toutes les lignes VTY, puisqu'un utilisateur pourrait essayer de se connecter à n'importe laquelle d'entre elles.

6.1.8 Configuration des ACL IPv4 étendues

Tout comme les ACL standards, les ACL étendues peuvent être numérotées ou nommées. Le procédé pour ajouter et activer des ACL étendues est le même que pour les ACL standards, sauf que les commandes sont plus riches pour fournir les caractéristiques additionnelles des ACL étendues.

ACL étendue numérotée

```
Router(config)# access-list <numéro> {deny | permit | remark <text>} <protocol>
   <source> <source_wildcard> [<opérateur> {<port>}] <destination> <
   destination_wildcard> [<opérateur> {<port>}] [established] [log]
```

Cela fait beaucoup de paramètres, mais tous ne sont pas nécessaires.

- `<numéro>` : Les ACL étendues vont de 100 à 199 et de 2000 à 2699.
- `deny` : Bloque l'accès si la condition correspond.
- `permit` : Autorise l'accès si la condition correspond.
- `remark <texte>` : Optionnel, ajoute un commentaire pour la documentation. Les remarques sont limitées à 100 caractères.
- `<protocol>` : Nom ou port d'un protocole internet. On utilise souvent `ip`, `tcp`, `udp` et `icmp`. Le mot clé `ip` correspond à tous les protocoles IP.
- `<source>` : Identifie l'hôte ou le réseau source à filtrer. Pour spécifier tout réseau, on utilise le mot clé *any*. Pour spécifier un hôte, on peut soit utiliser le mot clé *host* avec l'adresse IP, ou simplement l'adresse seule sans le mot clé.
- `<source_wildcard>` : Optionnel, correspond au masque générique appliqué à la source.
- `<destination>` : Identifie l'hôte ou le réseau destination à filtrer. Pour spécifier tout réseau, on utilise le mot clé *any*. Pour spécifier un hôte, on peut soit utiliser le mot clé *host* avec l'adresse IP, ou simplement l'adresse seule sans le mot clé.
- `<destination_wildcard>` : Optionnel, correspond au masque générique appliqué à la destination.
- `<opérateur>` : Optionnel, compare les ports source et destination. On peut utiliser les mots clé `lt`, `gt`, `eq` et `neq`.
- `<port>` : Optionnel, un port TCP ou TCP.
- `established` : Optionnel, pour le protocole TCP seulement.
- `log` : Optionnel, génère et envoie un message informationnel à chaque correspondance avec une ACE. Le message contiendra l'ACL, la condition, l'adresse source, et le nombre de paquets. Il est généré pour le premier paquet correspondant. Il devrait seulement être implémenté pour le déboggage ou la sécurité.

Pour appliquer l'ACL à une interface, on utilise la même commande que pour les ACL standards.

```
Router(config-if)# ip access-group {<numéro> | <nom>} {in | out}
```

Protocoles et ports

Protocoles

```
R1(config)# access-list 100 permit ?
  <0-255>      An IP protocol number
  ahp          Authentication Header Protocol
  dvmrp        dvmrp
  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          K19Q NOS compatible IP over IP tunneling
  object-group Service object group
  ospf         OSPF routing protocol
  pcp          Payload Compression Protocol
  pim          Protocol Independent Multicast
  tcp          Transmission Control Protocol
  udp          User Datagram Protocol
R1(config)# access-list 100 permit
```

Mots clé de ports

La sélection d'un protocole influence la sélection d'un port. Il est possible d'utiliser les numéros de port comme leur nom, bien que le nom soit plus lisible.

```
R1(config)# access-list 100 permit tcp any any eq ?
  <0-65535>   Port number
  bgp         Border Gateway Protocol (179)
  chargen     Character generator (19)
  cmd         Remote commands (rcmd, 514)
  daytime     Daytime (13)
  discard     Discard (9)
  domain      Domain Name Service (53)
  echo        Echo (7)
  exec        Exec (rsh, 512)
  finger      Finger (79)
  ftp         File Transfer Protocol (21)
  ftp-data    FTP data connections (20)
  gopher      Gopher (70)
  hostname    NIC hostname server (101)
  ident       Ident Protocol (113)
  irc         Internet Relay Chat (194)
  klogin      Kerberos login (543)
  kshell      Kerberos shell (544)
  login       Login (rlogin, 513)
  lpd         Printer service (515)
  msrpc       MS Remote Procedure Call (135)
  nntp        Network News Transport Protocol (119)
  onep-plain  Onep Cleartext (15001)
  onep-tls    Onep TLS (15002)
  pim-auto-rp PIM Auto-RP (496)
  pop2        Post Office Protocol v2 (109)
  pop3        Post Office Protocol v3 (110)
  smtp        Simple Mail Transport Protocol (25)
  sunrpc      Sun Remote Procedure Call (111)
  syslog      Syslog (514)
  tacacs      TAC Access Control System (49)
  talk        Talk (517)
  telnet      Telnet (23)
  time        Time (37)
  uucp        Unix-to-Unix Copy Program (540)
```

```
  whois       Nicname (43)
  www         World Wide Web (HTTP, 80)
```

TCP `established`

Le mot clé `established` permet de faire du pare-feu TCP stateful basique. Il permet au trafic interne de quitter le réseau privé en permettant temporairement le trafic de réponse en retour d'entrer le réseau privé.

Sans ce mot clé `established`, le trafic serait autorisé dans les deux sens tout court. N'importe quel hôte à l'extérieur du réseau pourrait entrer dans le réseau privé.

ACL étendue nommée

On entre en mode de configuration d'ACL comme pour les ACL standards, mais avec le mot clé `extended` :

```
Router(config)# ip access-list extended <nom>
```

Un exemple avec deux ACL, `SURFING`, qui autorise le trafic HTTP et HTTPS interne, et `BROWSING`, qui autorise le trafic HTTP et HTTPS retour. On met l'ACL `SURFING` en `in` et l'ACL `BROWSING` en `out` :

```
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# remark Autorise le trafic HTTP et HTTPS interne
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config-ext-nacl)# exit
R1(config)# ip access-list extended BROWSING
R1(config-ext-nacl)# remark Autorise seulement les réponses HTTP et HTTPS
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0/0
R1(config-if)# ip access-group SURFING in
R1(config-if)# ip access-group BROWSING out
R1(config-if)# end
R1# show access-lists
Extended IP access list SURFING
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443 (124 matches)
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established (369 matches)
R1#
```

Modifier une ACL étendue

Même chose que pour les ACL standards, on utilise soit un éditeur de texte (quand les modifications sont nombreuses), soit les numéros de séquence en supprimant l'entrée au numéro en question puis en l'ajoutant à ce numéro.

Vérification d'ACL étendues

Pour voir l'application d'une ACL sur une interface ainsi que sa direction :

```
R1# show ip interface g0/0/0 | include access-list
```

Pour afficher les ACE dans les ACL ainsi que leurs statistiques :

```
R1# show access-lists
```

Pour valider toute la configuration, ainsi que les `remarks` :

```
R1# show running-config | begin ip access-list
```

6.2 Sur Linux

Avec les permissions classiques, chaque fichier ne peut indiquer des permissions que pour un seul utilisateur et un seul groupe. Les ACL permettent d'étendre le nombre d'utilisateurs et de groupes ayant des droits sur un même fichier. On peut donc ajouter d'autres utilisateurs et groupes et définir leurs droits *séparément*.

Les ACL surchargent les droits POSIX. Les fichiers qui reçoivent une ACL ont le marqueur "+". Cependant, si on utilise la commande `setfacl` au lieu de `chmod` pour mettre uniquement des permissions classiques, le fichier en question ne reçoit pas d'ACL. Il ne reçoit pas le marqueur "+" avec la commande `ls -l`.

Chapitre 7

NAT

7.1 Les adresses IPv4 privées

Comme on l'a vu (voir 5.3.1), il n'y a plus assez d'adresses IPv4 pour assigner une adresse unique à tous les périphériques connectés à l'Internet. La RFC 1918 définit des plages d'adresses IPv4 privées, qui ne sont pas routables sur Internet (voir 5.2.2). En effet, ces adresses servent pour l'adressage (et le routage) interne aux entreprises mais ne permettent pas d'identifier une entité.

Pour qu'un périphériques ayant une adresse privée puisse joindre l'extérieur, il faut translater son adresse en adresse publique. Avec la NAT, une seule adresse IPv4 publique peut être partagée par des centaines voire des milliers de périphériques ayant chacun une adresse IPv4 privée.

7.2 Fonctionnement de la NAT

La NAT a plusieurs intérêts, mais le principal est de conserver des adresses IPv4. Le princpe est que les machines utilisent des adresses privées en interne et ne prennent une adresse publique seulement quand c'est nécessaire. Cela donne également le bénéfice d'offir d'avantage de confidentialité et de sécurité, puisque les réseaux privés ne sont pas visibles de l'extérieur.

Les routeurs faisant de la NAT peuvent être configurées avec une ou plusieurs adresses publiques valides. Ces adresses constituent le *pool NAT*. Quand un périphérique interne envoie du trafic vers l'extérieur, le routeur traduit l'adresse privée interne en une adresse publique du pool. Du point de vue des périphériques externes, tout le trafic entrant et sortant du réseau privé semble avoir une adresse IPv4 publique provenant du pool.

La fonction NAT est en général activée sur une routeur d'extrémité, connecté à internet. Ce routeur constitue alors la porte de sortie du réseau.

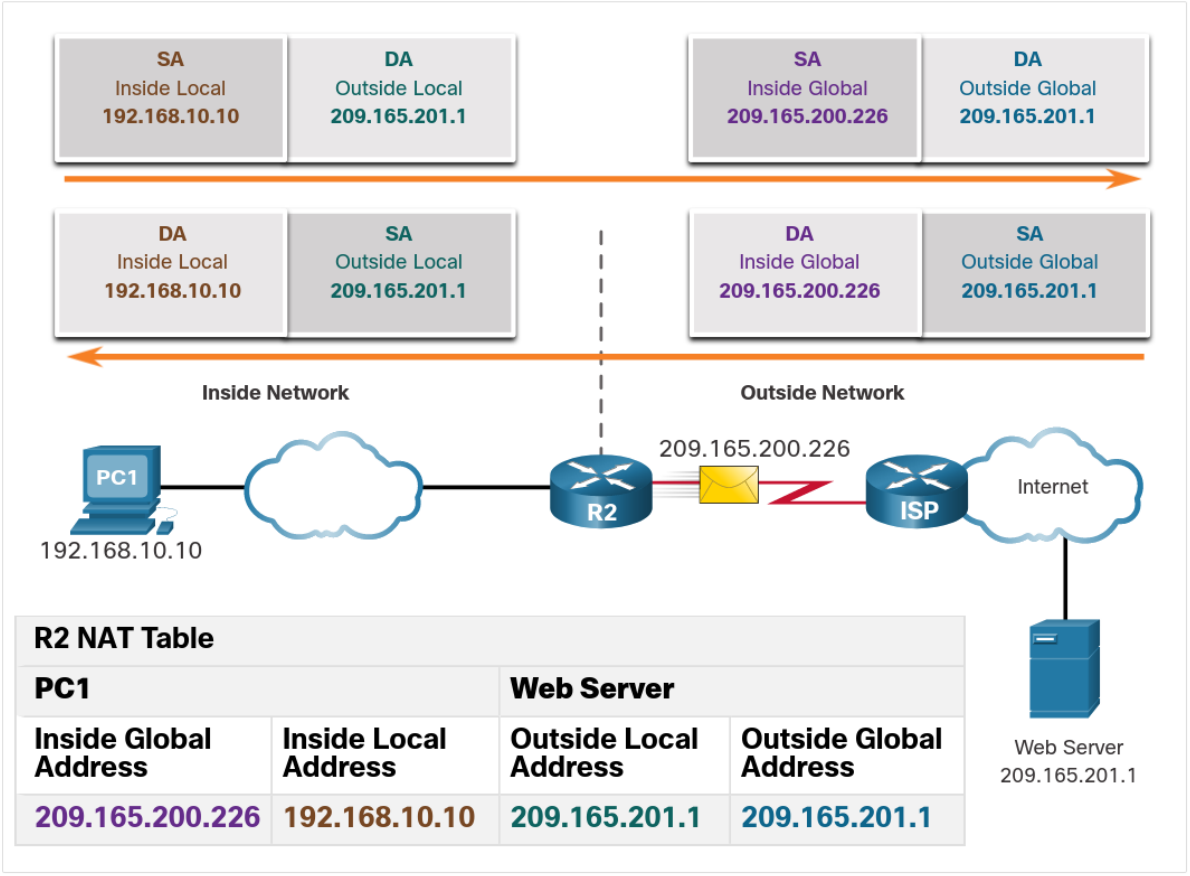
Le routeur maintient une table NAT de correspondances entre adresses privées et adresses publiques.

7.3 Terminologie de NAT

Le *réseau interne* désigne les réseaux qui sont susceptibles d'êtres traduits. Le *réseau externe* désigne tous les autres réseaux.

La NAT inclue 4 types d'adresses. Pour déterminer quel terme utiliser, il faut se placer du côté du périphérique qui a l'adresse traduite.

— **Adresse interne** : l'adresse du périphérique qui subit la traduction NAT.
— **Adresse externe** : l'adresse du périphérique destination.
— **Adresse locale** : toute adresse qui apparaît sur la portion interne du réseau.
— **Adresse globale** : toute adresse qui apparaît sur la portion externe du réseau.



R2 NAT Table

PC1		Web Server	
Inside Global Address	**Inside Local Address**	**Outside Local Address**	**Outside Global Address**
209.165.200.226	192.168.10.10	209.165.201.1	209.165.201.1

1. **Adresse interne locale** — C'est l'adresse de la source, vue de l'intérieur du réseau. C'est donc une adresse privée.
2. **Adresse interne globale** — C'est l'adresse de la source, vue de l'extérieur du réseau. C'est donc une adresse publique.
3. **Adresse externe locale** — C'est l'adresse de la destination, vue de l'intérieur du réseau. C'est rare, mais cette adresse pourrait être différente de l'adresse routable de la destination.
4. **Adresse externe globale** — C'est l'adresse de la destination, vue de l'extérieur du réseau. C'est une adresse publique routable. La plupart du temps, l'adresse externe locale et l'adresse externe globale sont les mêmes.

7.4 Types de NAT

7.4.1 NAT statique

La NAT statique mappe les adresses locales et globales une à une. Ce sont des correspondances définies par l'administrateur du réseau et elles restent constantes.

Static NAT Table	
Inside Local Address	**Inside Global Address - Addresses reachable via R2**
192.168.10.10	209.165.200.226
192.168.10.11	209.165.200.227
192.168.10.12	209.165.200.228

La NAT statique est surtout intéressante dans les cas où des périphériques, comme des serveurs web, ont besoin d'être joignables depuis internet en ayant une adresse consistante. Elle peut aussi être utilisée pour des périphériques qui ont besoin d'être accessibles par certaines personnes hors du site, mais pas par le public général sur internet (par exemple en SSH).

La NAT statique requiert un nombre suffisant d'adresses IPv4 publiques pour accomoder les sessions simultanées.

7.4.2 NAT dynamique

La NAT dynamique utilise un pool d'adresses publiques et les assigne au premier arrivé premier servi. Quand un périphérique interne a besoin de sortir du réseau interne, la NAT dynamique lui assigne une adresses publique provenant du pool.

IPv4 NAT Pool	
Inside Local Address	**Inside Global Address Pool - Addresses reachable via R2**
192.168.10.12	209.165.200.226
Available	209.165.200.227
Available	209.165.200.228
Available	209.165.200.229
Available	209.165.200.230

Tout comme la NAT statique, la NAT dynamique requiert un nombre suffisant d'adresses IPv4 publiques pour accomoder les sessions simultanées.

7.4.3 PAT

La PAT s'appelle aussi *NAT overload*. Elle mappe de multiples adresses IPv4 privées vers une seule adresse IPv4 publique, ou bien quelques unes. C'est ce que font la plupart des routeurs de particuliers (box internet). Le FAI assigne une adresse publique au routeur mais plusieurs membres du foyer peuvent accéder à internet.

La PAT est la forme de NAT la plus commune, à la fois pour les particuliers et pour les entreprises.

Pour mapper de multiples adresses à une seule, la PAT suit le numéro de port. Quand un périphérique initie une session TCP/IP, il génère une valeur de port TCP ou UDP source. En ICMP, il génère un identifiant de requête pour identifier la session. Quand le routeur faisant la NAT reçoit un paquet du client, il utilise ce port source pour identifier la traduction NAT de manière unique.

La PAT s'assure que chaque client utilise un numéro de port unique pour chaque session sur internet. Quand la réponse revient du serveur, le port source, qui devient port destination, détermine à quel périphérique le routeur doit renvoyer les paquets. Ce processus de NAT vérifie aussi que les paquets répondent bien à une requête, ce qui ajoute une couche de sécurité.

NAT Table with Overload

Inside Local IP Address	Inside Global IP Address	Outside Local IP Address	Outside Global IP Address
192.168.10.10:1555	209.165.200.226:1555	209.165.201.1:80	209.165.201.1:80
192.168.10.11:1331	209.165.200.226:1331	209.165.202.129:80	209.165.202.129:80

Prochain port disponible

Si le port source du client, en arrivant au routeur, n'est pas modifié, on risque d'utiliser des ports qui sont déjà en utilisation par d'autres sessions. Dans un réseau local, ceci ne serait pas un problème, parce que les hôtes ont des adresses IPv4 uniques. Mais dans le cadre de la NAT, ce cas de figure entraînerait une situation ou des paquets provenant de différents hôtes auraient la même adresse avec le même port.

La PAT essaie de préserver les ports originaux, mais si le port est déjà utilisé, elle assigne le premier port disponible dans la plage adéquate (0–511, 512–1023 ou 1024–65535). S'il n'y a plus de port disponible mais qu'il y a plus d'une adresse externe dans le pool, la PAT passe à l'adresse suivante du pool et essaie d'allouer le port originel.

Quand le trafic revient dans le réseau interne, si le port source a été modifié, le routeur remodifiera le port destination à la valeur d'origine.

Paquets sans segment de couche 4

Il y a des paquets qui transportent des données autres que TCP ou UDP. Ces paquets ne contiennent pas de port de couche 4. Le cas le plus commun est ICMPv4, mais chaque type de protocole est géré de manière différente par la PAT.

En ICMP, les *query messages*, les *echo requests* et les *echo replies* ont un `Query ID`. Ce `Query ID` est incrémenté à chaque *echo request*. La PAT l'utilise à la place du numéro de port de couche 4.

7.5 Avantages et inconvénients

7.5.1 Avantages de la NAT

— En permettant la privatisation d'intranets, la NAT conserve l'adressage légalement alloué. Avec la PAT, très peu d'adresses externes sont nécessaires pour gérer beaucoup d'hôtes internes.
— La NAT augmente la flexibilité des connexions publiques. Avec de multiples pools, des pools de renfort, et des pools d'équilibrage de charge, on peut assurer des connexions publiques fiables.
— La NAT permet un adressage interne consistant. Si on n'utilise pas de NAT et d'adresses privées, changer l'adressage oblige à changer toutes les adresses des hôtes du réseau. La NAT permet à l'adressage privé de rester identique tout en changeant l'adressage publique. Une entreprise peut par exemple changer de FAI sans toucher aux adresses privées internes.
— Avec les adresses privées de la RFC 1918, la NAT cache les adresses des utilisateurs et autres périphériques. Certains disent que c'est une caractéristique de sécurité, mais ce qui apporte la sécurité est un firewall stateful.

7.5.2 Inconvénients de la NAT

Le fait que les hôtes sur internet ont l'air de communiquer avec le routeur faisant la NAT plutôt qu'avec le véritable hôte du réseau privé pose quelques problèmes.

— La traduction d'adresses prend du temps. Cela augmente les délais de transfert, ce qui peut être gênant pour les protocoles en temps réel comme la VoIP. En effet, le routeur doit examiner chaque paquet pour déterminer s'il doit êtré NATté, puis il doit altérer l'en-tête IPv4 et possiblement l'en-tête TCP ou UDP. Ensuite, le *checksum* de l'en-tête IPv4 et le *checksum* TCP ou UDP doit être recalculé à chaque traduction.
Le premier paquet prend la route longue avec tous ces calculs. Si un cache existe, les paquets suivants iront plus vite, sinon le délai s'appliquera à tous les paquets.
— La déplétion d'adresses IPv4 conduit de plus en plus de FAI à fournir à leurs clients des adresses privées. Dans ce cas, le client NATte son adresse privée vers l'adresse privée fournie par le FAI, puis le FAI refait une deuxième NAT pour traduire l'adresse privée assignée au client vers une adresse publique.
Ce procédé s'appelle le Carrier Grade NAT (CGN).
— L'adressage de bout-en-bout est perdu. Beaucoup de protocoles et applications dépendent de l'adressage de bout-en-bout de la source à la destination. Certaines applications ne fonctionnent donc pas avec la NAT. Par exemple, certaines applications de sécurité, comme les signatures numériques, échouent parce que l'adresse source change avant d'arriver à destination. Les applications utilisant des adresses physiques à la place de noms de domaines ne peuvent pas rejoindre une destination qui est NATtée. Parfois, on peut contourner le problème avec la NAT statique.
— La traçabilité de bout-en-bout est aussi perdue. Cela complique le déboggage, quand les paquets subissent de multiples traductions d'adresse.
— La NAT complique certains protocoles de tunneling, comme IPSec. Les modifications d'en-tête cause des échecs de vérification d'intégrité.
— Des services qui dépendent de l'initiation d'une session TCP à partir de l'extérieur,

ou bien des protocoles stateless comme ceux utilisant UDP peuvent aussi échouer. À moins que le routeur NAT ait été configuré pour gérer ces protocoles, les paquets entrants ne peuvent pas atteindre leur destination. Certains protocoles, comme le FTP en mode *passive*, peuvent gérer une instance de NAT mais écouent quand les deux extrémités sont séparés d'internet par la NAT.

7.6 Configuration de NAT statique

7.6.1 Configuration

Il y a deux tâches pour configurer des traductions de NAT statique :

1. Il faut créer un mappage entre l'adresse locale interne et l'adresse globale interne. Par exemple, avec une adresse locale interne `192.168.10.254` et une adresse globale interne `209.165.201.5` :

```
R1(config)# ip nat inside source static 192.168.10.254 209.165.201.5
```

2. Une fois le mappage configuré, les interfaces qui participent à la traduction doivent être configurées en tant que `inside` ou `outside`, relativement à la NAT. Avec l'interface `s0/1/0` en `inside` et `s0/1/1` en outside :

```
R1(config)# interface s0/1/0
R1(config-if)# ip address 192.168.1.2 255.255.255.252
R1(config-if)# ip nat inside
R1(config-if)# exit
R1(config)# interface s0/1/1
R1(config-if)# ip address 209.165.200.1 255.255.255.252
R1(config-if)# ip nat outside
```

Avec cette configuration, les paquets arrivant sur l'interface interne de `R1` (`s0/1/0`) à partir de l'adresse locale interne `192.168.10.254` seront traduits et transférés vers le réseau externe. Les paquets arrivant sur l'interface externe de `R1` (`s0/1/1`) à destination de l'adresse globale interne (`209.165.201.5`) seront traduits vers l'adresse locale interne (`192.168.10.254`) et transférés au réseau interne.

7.6.2 Vérification

Pour vérifier l'opération de la NAT sur le routeur :

```
R1# show ip nat translations
```

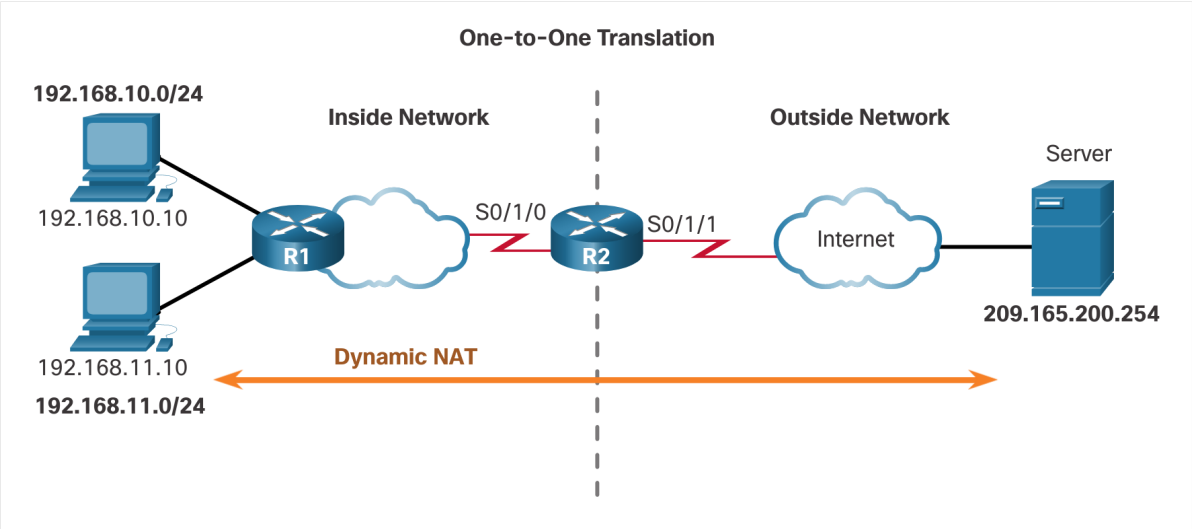
Cette commande montre les traductions NAT actives. Dans le cas d'une NAT statique, la traduction sera toujours présente dans la table, peu importe s'il y a des communications actives. Quand on tape la commande alors qu'il y a une session active, la sortie montre aussi l'adresse du périphérique externe.

Pour afficher les statistiques, c'est-à-dire des informations sur le nombre total de traductions actives, de paramètres de configuration, du nombre d'adresses dans le pool et du nombres d'adresses allouées :

```
R1# show ip nat statistics
```

7.7 Configuration de NAT dynamique

Prenons le cas de figure suivant :



Le routeur `R2` fait la NAT avec un pool d'adresses publiques de `209.165.200.226` à `209.165.200.240`. Ce pool (adresses globales internes) est disponible à tout périphérique du réseau interne (premier arrivé premier servi).

7.7.1 Configuration

Nous allons configurer la NAT pour tous les hôtes du réseau privé `192.168.0.0/16`, ce qui inclut les deux LAN.

1. Il faut définir le pool d'adresses utilisés dans la traduction :

   ```
   R2(config)# ip nat pool NAT-POOL1 209.165.200.226 209.165.200.240 netmask
       255.255.255.224
   ```

   Le `netmask` indique le masque utilisé pour ces adresses publiques.

2. Il faut créer une ACL standard pour identifier (`permit`) les adresses à traduire. Une ACL trop permissive peut mener à des comportements inattendus. Ne pas oublier qu'il y a un `deny all` implicite à la fin de l'ACL.

   ```
   R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
   ```

3. Il faut lier l'ACL au pool, avec la syntaxe suivante :

   ```
   Router(config)# ip nat inside source list {<acl_number> | <acl_name>}
      pool <pool_name>
   ```

   Dans notre cas :

   ```
   R2(config)# ip nat inside source list 1 pool NAT-POOL1
   ```

4. Il faut identifier les interfaces internes, par rapport à la NAT : il s'agit de toute interface connectée au réseau interne.

   ```
   R2(config)# interface s0/1/0
   R2(config-if)# ip nat inside
   ```

5. Il faut identifier les interfaces externes, par rapport à la NAT : il s'agit de toute interface connectée au réseau externe.

```
R2(config)# interface s0/1/1
R2(config-if)# ip nat outside
```

7.7.2 Vérification

La commande `show ip nat translations` montre les traductions statiques et toute traduction dynamique créée par le trafic. L'ajout du mot clé `verbose` permet d'avoir plus d'informations sur chaque traduction.

Par défaut, les traductions expirent au bout de 24 heures, mais on peut reconfigurer ça :

```
Router(config)# ip nat translation timeout <secondes>
```

Pour effacer les entrées dynamiques :

```
Router# clear ip nat translation *
```

L'étoile indique d'effacer toutes les entrées. Effacer les entrées est utile pour le déboggage, mais pour éviter d'effacer des sessions actives, on peut ajouter des mots clé pour choisir quoi effacer :

```
Router# clear ip nat translation inside <global_ip> <local_ip> [outside <
    local_ip> <global_ip>]
Router# clear ip nat translation <protocol> inside <global_ip> <global_port> <
    local_ip> <local_port> [outside <local_ip> <local_port> <global_ip> <
    global_port>]
```

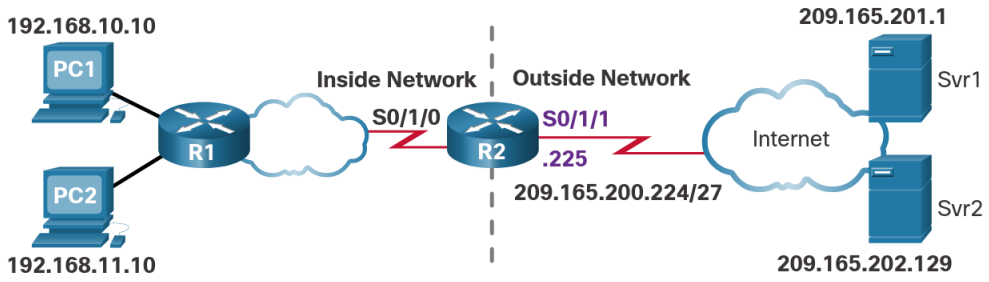
Seules les traductions dynamiques sont effacées. Les traductions statiques ne peuvent pas être effacées de la table.

On peut là aussi utiliser `show ip nat statistics`.

7.8 Configuration de PAT

Il y a 2 façons de configurer la NAT overload, dépendant de la façon dont le FAI alloue des adresses IPv4 publiques. Le FAI peut soit allouer une seule adresse soit plusieurs.

Prenons le scénario suivant :



NAT Table			
Inside Local Address	**Inside Global Address**	**Outside Global Address**	**Outside Local Address**
192.168.10.10:1444	209.165.200.225:1444	209.165.201.1:80	209.165.201.1:80
192.168.11.10:1444	209.165.200.225:1445	209.165.202.129:80	209.165.202.129:80

7.8.1 Configuration pour une adresse publique

Pour configurer la PAT avec une seule adresse globale interne, il suffit d'ajouter `overload` à la commande de NAT dynamique :

```
R2(config)# ip nat inside source list 1 interface s0/1/0 overload
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# interface s0/1/0
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)# interface s0/1/1
R2(config-if)# ip nat outside
```

Tous les hôtes du réseau `192.168.0.0/16`, correspondant à l'ACL 1, qui envoient du trafic sur internet en passant par le routeur `R2`, verront leurs adresses traduites en `209.165.200.225`. Cette adresse est celle de l'interface `s0/1/1`. Les paquets seront identifiés par leur port grâce au mot clé `overload`.

7.8.2 Configuration avec un pool d'adresses

Le principe est le même, sauf qu'au lieu d'utiliser une seule adresse globale interne, les hôtes du réseau interne vont se partager un petit nombre d'adresses publiques.

En prenant la même topologie que précédemment (voir 7.8) :

```
R2(config)# ip nat pool NAT-POOL2 209.165.200.226 209.165.200.240 netmask
    255.255.255.224
R2(config)# access-list 1 permit 192.168.0.0 0.0.255.255
R2(config)# ip nat inside source list 1 pool NAT-POOL2 overload
R2(config)# interface s0/1/0
R2(config-if)# ip nat inside
R2(config-if)# exit
R2(config)# interface s0/1/1
R2(config-if)# ip nat outside
```

Le pool `NAT-POOL2` est lié à une ACL pour permettre le réseau `192.168.0.0/16` d'être traduit. Encore une fois, ces hôtes peuvent partager des adresses grâce au mot clé `overload`.

7.8.3 Vérification

Les commandes de vérification pour la PAT sont les mêmes que pour les NAT statique et dynamique.

```
R2# show ip nat translations
```

Les numéros de port apparaissent, permettant de différencier les adresses globales internes, qui sont les mêmes.

```
R2# show ip nat statistics
```

La sortie nous renseigne sur le nombre et le type de traductions actives, les paramètres de configuration, le nombre d'adresses dans le pool, et combien ont été allouées.

7.9 NAT64

Beaucoup de réseaux utilisent à la fois l'IPv4 et l'IPv6. Il faut donc que l'IPv6 soit utilisable avec la NAT.

L'IPv6 a été développée avec l'intention de rendre inutile la NAT pour IPv4, avec la traduction d'adresses publiques et privées. Cependant, l'IPv6 inclut son propre adressage privé : les Unicast Local Address (ULA) (voir 5.3.5).

Ces adresses sont similaires aux adresses IPv4 privées de la RFC 1918, mais elles ont un autre but. Les ULA sont prévues seulement pour la communication locale à l'intérieur d'un site. Elles n'ont pas pour but de proposer un adressage IPv6 additionnel, ni d'ajouter un niveau de sécurité.

Par contre l'IPv6 propose une traduction de protocole entre l'IPv4 et l'IPv6. Cette traduction s'appelle la NAT64.

La NAT64 est utilisé dans un contexte très différent de la NAT pour l'IPv4. Elle sert à faire communiquer de manière transparente des réseaux IPv6 seuls avec des réseaux IPv4 seuls. Ce n'est pas une traduction IPv6 locale vers globale.

Pour aider la transition vers l'IPv6, l'IETF a développé plusieurs techniques, accomodant une variété de scénarios hybrides :

- **Dual-stack** — Les périphériques font tourner des protocoles associés à la fois à l'IPv4 et à l'IPv6.
- **Tunneling** — Encapsulation d'un paquet IPv6 dans un paquet IPv4. Cela permet au paquet IPv6 de traverser un réseau IPv4 seul.
- **Traduction** — La NAT64 ne devrait pas être utilisée à long terme. Il y a eu différentes types de NAT pour l'IPv6, parmi lesquelles la Network Address Translation — Protocol Translation (NAT-PT), qui est dépréciée par l'IETF. On utilise plutôt sa remplaçante, la NAT64.

Chapitre 8

Proxy

8.1 Définition

Le proxy est un relayage applicatif. On interpose une application entre le client et le serveur. Le proxy refait la requête auprès du serveur à la place du client, et transmet au client la réponse du serveur. Cette application doit donc comprendre le protocole sous-jacent.

Les relais applicatifs permettent :

- de palier les inconvénients du filtrage simple (voir l'exemple du FTP : 2)
- de protéger les applications dangereuses (comme Sendmail) en accès entrant
- de dissimuler la véritable machine source en accès sortant :
  - la connexion semble provenir de la machine relais et non de la machine réelle
  - la machine réelle n'a pas besoin d'être connue (routable) sur Internet
- d'affiner les droits d'accès et les traces en fonction du protocole sous-jacent :
  On peut alors créer des règles dépendant du service, des adresses IP source et destination, du moment de la connexion, de l'utilisateur, des fonctions à l'intérieur d'un protocole.
- de centraliser l'authentification des accès

8.2 Relayage applicatif simple (non transparent)

Les applications clientes se connectent sur un port donné de la machine relais pour obtenir le service demandé. Elles doivent donc être configurées en conséquence.

8.3 Relayage applicatif avec SOCKS

SOCKS est un protocole générique de proxy pour des applications réseaux basées sur TCP/IP. Le serveur SOCKS est implémenté au niveau de la couche Application (osi 7). Le client SOCKS est implémenté entre la couche Application (osi 7) et la couche Transport (osi 4).

Le protocole SOCKS encapsule les applications clientes.

8.4 Relayage applicatif transparent

En mode transparent, la machine relais est vue comme une routeur sur le réseau. Elle intercepte les demandes de connexion et lance automatiquement le relais applicatif correspondant vers la machine destinataire.

Avec ce mode transparent, les applications clientes n'ont pas à être modifiées tant que l'on utilise des numéros de port standards.

Chapitre 9

Authentification RADIUS

RADIUS permet la mise en place d'un moyen centralisé pour authentifier les utilisateurs accédant au réseau.

C'est un protocole qui centralise toutes les authentifications d'utilisateurs et toutes les informations d'accès aux services du réseau à partir d'un serveur d'authentification.

Typiquement, le serveur RADIUS est une machine autonome qui gère une Base de Données contenant des informations sur chaque utilisateur, ainsi que des informations de configuration (types de services à délivrer...).

L'authentification est basée sur le port UDP 1812 et l'autorisation sur le port UDP 1813.

Chapitre 10

Les VPN

10.1 Présentation

Un VPN est utilisé pour sécuriser le trafic entre sites et utilisateurs. Il crée une connexion privée de bout-en-bout.

Le VPN est une connection :

— **Virtuelle** — Les informations d'un réseau privé sont transportées dans un *tunnel* sur un réseau public.
— **Privée** — Le trafic est chiffré pour la confidentialité des données pendant le transport sur le réseaux public.

Un tunnel n'est rien d'autre que l'encapsulation d'un protocole par un autre.

Comme méthodes de *tunneling*, on peut citer Generic Routing Encapsulation (GRE) et Layer 2 Tunneling Protocol (L2TP). Ces deux méthodes ne fournissent pas de chiffrement. Ils se contentent d'encapsuler un protocole dans un autre.

Par contre, IPSec, lui, permet de chiffrer les données encapsulées. Cela offre la même sécurité que sur un réseau privé.

Le réseau privé doit assurer 3 choses :

1. **Confidentialité** (chiffrement)
2. **Intégrité** (hachage)
3. **Authentification**

Il faut noter qu'IPSec ne gère que l'IP unicast. S'il faut encapsulées des données sur autre chose que sur IP, ou bien si on a besoin de multicast (pour OSPF par exemple), il faudra utiliser GRE ou L2TP. On peut bien sûr, s'il y a besoin de chiffrement, encapsuler de l'IPSec dans du GRE ou du L2TP.

Les VPN sont utilisés de deux façons :

1. **VPN site à site** — Les périphériques VPN terminaux (passerelles VPN) sont préconfigurés pour établir un tunnel sécurisé. Le trafic est seulement chiffré entre ces deux périphériques. Les hôtes internes ne savent alors pas qu'il y a un VPN.

2. **VPN à accès distant** — Un VPN à accès distant est créé de manière dynamique pour établir une connexion sécurisée entre un client et un périphérique VPN terminal. Un employé physiquement à l'extérieur du réseau d'entreprise peut par exemple se connecter avec un VPN à accès distant au réseau privé de l'entreprise.

Les VPN peuvent être déployés de plusieurs façons en fonction de qui gère le VPN :

— **VPN d'entreprise** — C'est une solution commune pour sécuriser le trafic d'une entreprise à travers internet. Des VPN site à site ou à accès distant sont créés et gérés par l'entreprise en utilisant des VPN SSL et IPSec.
— **VPN de prestataire de service** — Les VPN gérés par un prestataire de services sont créés et gérés sur le réseau du prestataire. Il utilise MultiProtocol Label Switching (MPLS) en couche 2 ou 3 pour créer des canaux sécurisés entre les sites d'une entreprise. MPLS est une technologie de routage utilisée par le prestataire pour créer des chemins virtuels entre sites. Cela sépare le trafic du trafic d'autres clients. D'autres solutions anciennes incluent les VPN Frame Relay et ATM.

10.2 Types de VPN

10.2.1 VPN à accès distant

Les VPN à accès distant sont en général activés dynamiquement par l'utilisateur lorsque nécessaire. Ils peuvent être créés en utilisant IPSec ou SSL.

L'utilisateur peut créer la connexion de 2 façons :

1. **Sans client** — La connexion est sécurisée en utilisant une connexion SSL sur le navigateur web. SSL est surtout utilisé pour protéger le trafic HTTP (HTTPS) et des protocoles de messagerie comme Internet Message Access Protocol (IMAP) et POP3.
2. **Basé sur un client** — Un logiciel VPN client doit être installé sur le périphérique de l'utilisateur. L'utilisateur doit initialiser la connexion VPN grâce au client, puis s'authentifier à la passerelle VPN.

Quand un client négocie une connexion VPN SSL avec la passerelle VPN, il se connecte en fait avec TLS. TLS est la nouvelle version de SSL et se fait parfois appeler SSL/TLS. Ceci dit, les deux termes sont souvent utilisés de manière interchangeable.

SSL et IPSec offrent tous deux des solutions chiffrés et l'accès à toute application réseau. Mais si la sécurité est une priorité, IPSec sera le meilleur choix. Si la facilité de déploiement est la priorité, alors SSL sera le meilleur choix.

Les VPN IPSec et SSL ne sont pas mutuellement exclusifs mais complémentaires. On peut très bien, en fonction des besoins, implémenter SSL, IPSec ou les deux.

10.2.2 VPN site à site

Les VPN site à site sont utilisés pour joindre des réseaux à travers un autre réseau non fiable, comme internet. Les hôtes s'échanges du trafic TCP/IP normal non chiffré à travers un périphérique VPN. Ce périphérique s'appelle une passerelle VPN, et pourrait être un

routeur ou un pare-feu. Cette passerelle VPN encapsule et chiffre le trafic sortant. Puis elle envoie ce trafic à travers un tunnel VPN sur internet vers une passerelle VPN sur le site distant. La passerelle distante désencapsule et déchiffre le trafic avant de l'envoyer à l'hôte destination dans le réseau privé.

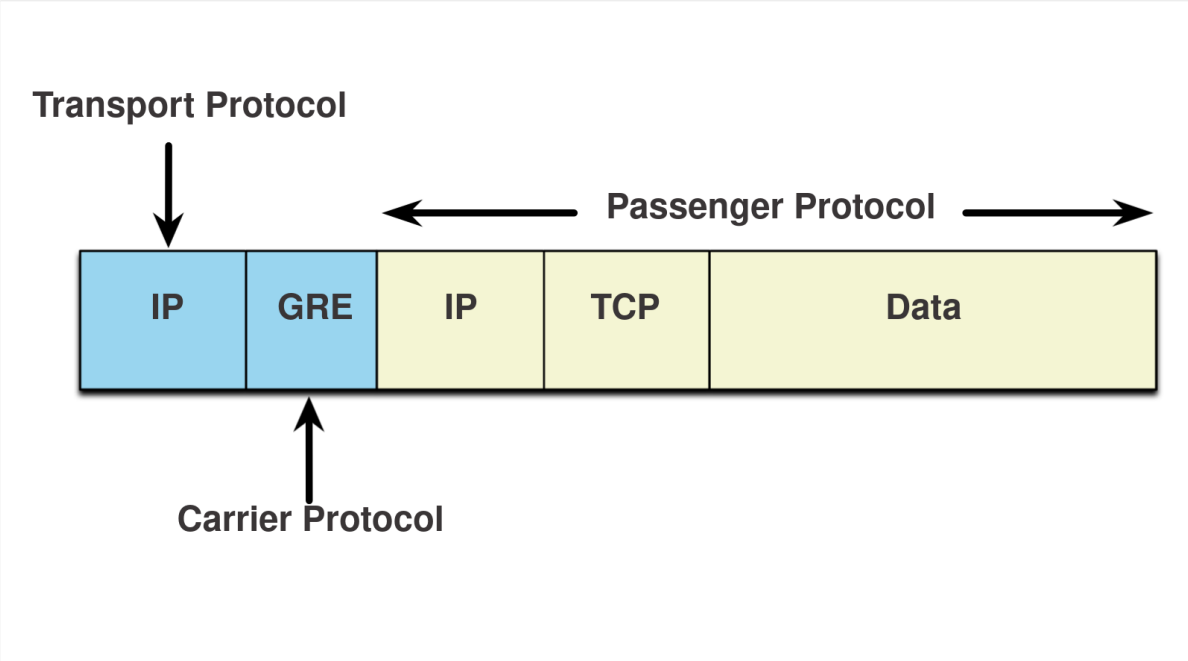
Les VPN site à site sont en général créés et sécurisés avec IPSec.

GRE sur IPSec

GRE est un protocole VPN site à site non sécurisé. Il peut encapsuler différents protocoles réseau, et supporte le multicast et le broadcast, chose que ne fait pas IPSec.

Si on a besoin de faire passer du trafic ayant besoin de GRE dans un VPN (par exemple des protocoles de routage), on peut encapsuler ce trafic dans un paquet GRE, puis encapsuler le paquet GRE dans un paquet IPSec.

Pour décrire ce type d'encapsulation, on utilise les termes de *protocole passager* (passenger protocol), *protocole porteur* (carrier protocol) et *protocole de transport* (transport protocol).



DMVPN

Les VPN site à site IPSec et GRE sur IPSec sont pratiques quand il n'y a qu'un nombre limité de sites à interconnecter. Mais dès qu'on ajoute d'autres sites ils deviennent insuffisants : chaque site aurait alors besoin de configurations statiques vers tous les autres sites, ou vers un site central.

DMVPN est une solution Cisco qui crée des VPN de manière dynamique et extensible. Il repose également sur IPSec et crée des connexions en étoile avec un site central (hub and spoke).

Chaque site est configuré avec Multipoint Generic Routing Encapsulation (mGRE). L'interface de tunnel mGRE permet à une seule interface GRE de fournir dynamiquement de multiples tunnels IPSec. De cette façon, quand on veut ajouter une connexion sécurisée

sur un nouveau site, la même configuration sur le site central soutient le tunnel. Pas de configuration supplémentaire n'est alors nécessaire.

Les site branches peuvent également obetnir les informations sur d'autres branches par le site central et créer des tunnels virtuels de branche à branche.

Virtual Tunnel Interface IPSec

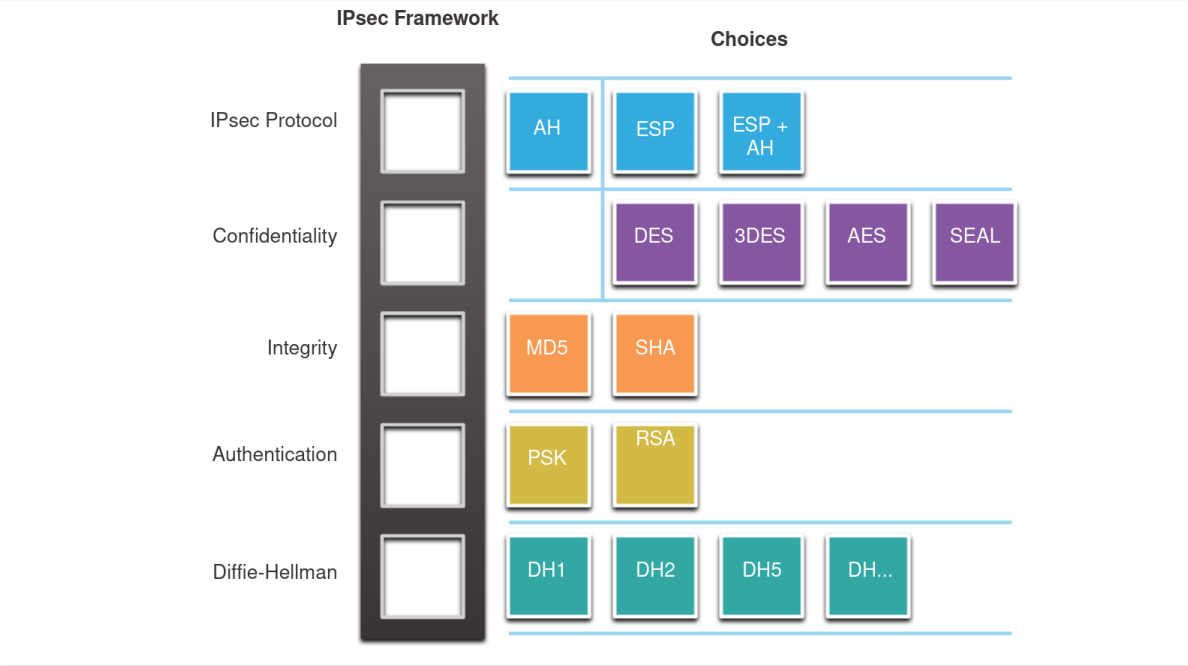
Comme les DMVPN, les Virtual Tunnel Interface (VTI) IPSec simplifient le processus de configuration requis pour de multiples sites. Les configurations VTI IPSec sont appliqués sur une interface virtuelle au lieu d'un mappage statique sur une interface physique.

Il est capable de traiter le trafic chiffrer unicast et multicast. Ainsi, les protocoles de routage n'ont besoin de tunnels GRE.

10.3 IPSec

IPSec est un standard de l'IETF (RFC 2401–2412) qui définit comment sécuriser un VPN à travers des réseaux IP. Il protège et authentifie les paquets IP entre la source et la destination. Il peut donc protéger les couches 4 à 7.

IPSec n'est pas lié à certaines règles spécifiques pour les communications sécurisées. Cette flexibilité lui permet d'intégrer de nouvelles technologies sans devoir mettre à jour les standards IPSec existants.



— **Protocole IPSec** — Les choix incluent Authentication Header (AH) et Encapsulation Security Payload (ESP). AH authentifie le paquet de couche 3. ESP chiffre le paquet de couche 3. Cependant cette combination est rarement utilisée car elle ne traverse pas la NAT.
— **Confidentialité** — Le chiffrement assure la confidentialité du paquet de couche 3. Les choix incluent DES, 3DES, Advanced Encryption Standard (AES) ou Software-optimized Encryption Algorithm (SEAL). On peut aussi choisir de ne pas chiffrer.

— **Intégrité** — Assure que les données arrivent inchangées à la destination en utilisant un algorithme de hachage, comme MD5 ou SHA.
— **Authentification** — IPSec utilise le protocole Internet Key Exchange (IKE) pour autentifier la source et la destination. Les méthodes d'authentification incluent Pre-Shared Key (PSK), certificats, RSA.
— **Diffie-Hellman** — Sécurise l'échange de clés avec l'alhorithme de Diffie-Hellman.

Quand on réunit plusieurs de ces fonctions pour une implémentation, on parle de Security Association (SA). Une SA est le bloc de base d'IPSec. Quand ils établissent un lien VPN, les pairs doivent utiliser la même SA.

10.3.1 Encapsulation de protocole IPSec

La première étape d'une implémentation IPSec est de choisir le protocole d'encapsulation. IPSec encapsule les paquets en utilisant AH ou ESP. Ce choix détermine quels autres options seront disponibles.

— AH est approprié lorsque la confidentialité n'est pas requise ou permise. Il fournit l'authentification et l'intégrité, mais pas le chiffrement (confidentialité).
— ESP fournit à la fois l'authentification et la confidentialité. Bien que le chiffrement et l'authentification soient optionnels avec ESP, il faut au minimum en choisir un des deux.

Chapitre 11

SSH

Historiquement, SSH permet de mettre en place une connexion sécurisée d'une machine vers une autre machine. La sécurité implique l'authentification et la confidentialité de données transportées.

Le protocole s'appuie sur la notion de tunnel sécurisé (comme le VPN).

Deux versions de SSH coexistent mais ne sont pas compatibles (SSH 1.3 ou 1.5 et SSH 2.0).

11.1 SSH 1.X

L'authentification se peut se faire selon 3 méthodes :

- identification des noms de machine et du client
- identification du nom de la machine associé à une empreinte calculée avec RSA
- identification du client basée sur la cryptographie asymétrique : RSA (voir 3.2)

La confidentialité est assurée par RSA, avec une clée de session unique dans les deux sens. Cette clé reste valable pour toute le durée de la session.

11.2 SSH 2.0

La confidentialité est assurée par des algorithmes symétriques (voir 3.1) : AES, 3DES, Blowfish, Arcfour, Carlisle Adams et Stafford Tavares (CAST)-128. Une clé de session est nécessaire par sens de communication. Ces clés sont renégociées périodiquement pendant toute la durée de la session.

Glossaire

LDAP Data Interchange Format
Format défini par la RFC 2849 pour standardiser la configuration de serveur LDAP, ainsi qu'exporter/importer les données. 7, 153, 233, 245

LDAP over SSL
Protocole LDAP sécurisé par SSL. LDAPS fonctionne sur le port 636. 154, 233, 245

VLAN ID
Numéro d'identification du VLAN sur 12 bits, qui permet donc jusqu'à 4096 VLAN différents. 33, 233, 250, 251, 259

VLAN Trunking Protocol
Protocole propriétaire Cisco pour diffuser des configurations VLAN sur plusieurs switchs après en avoir configuré un. 2, 233, 260

ABR
Area Border Router 78, 233

Abstract Syntax Notation One
Langage de description d'interface utilisé pour définir des structures de données, notammant dans les MIB. 126, 233, 235

ACE
Access Control Entry 204–206, 208–211, 214, 233

ACL
Access Control List 10, 128, 154, 155, 204–211, 213, 214, 221, 223, 224, 233

AD
Administrative Distance 69, 73, 75, 78, 87, 135, 233

AD
Active Directory 157, 158, 161, 163–167, 170, 184, 233

AD-DS
Active Directory Domain Service 158, 160, 163, 233

ADC
Analog to Digital Converter 111, 233

Address Resolution Protocol
Protocole de résolution d'adresses IPv4 en adresses MAC. Ce protocole maintient également une table d'adressage. 2, 233, 235

ADSL
Asymmetric DSL 14, 15, 119, 233, 234, 256, *Glossary :* Asymmetric DSL

AEP
Administrative Exchange Protocol 123, 233

AES
Advanced Encryption Standard 231, 233

AFNOR
Association Française de Normalisation 21, 233

AH
Authentication Header 231–233

ANSI
American National Standards Institute 20, 233

Anycast
Adresse unicast pouvant être attribuée à plusieurs périphériques. Un paquet envoyé un une adresse anycast est acheminé vers le périphérique le plus proche ayant cette adresse. 58, 59, 233

AP
Access Point 23, 233, 260

APIPA
Automatic Private IP Addressing 55, 233

Application Specific Integrated Circuits
Permet à un switch de prendre des décisions de commutation rapidement. 28, 233, 235

ARP
Address Resolution Protocol 2, 3, 26–28, 30, 42, 50, 51, 57, 63, 65, 71, 105, 106, 195, 233, 235, 238, 240, *Glossary :* Address Resolution Protocol

ARPANET
Advanced Research Projects Agency Network 233

AS
Autonomous System 75, 233, 241, 244

ASBR
Autonomous System Boundary Router 87, 233, 235, *Glossary :* Autonomous System Boundary Router

ASIC
Application Specific Integrated Circuits 28, 233, 235, *Glossary :* Application Specific Integrated Circuits

ASN.1
Abstract Syntax Notation One 126, 233, 235, *Glossary :* Abstract Syntax Notation One

Asymmetric DSL
Les débits ascendant et descendant ne sont pas les mêmes. Le débit ascendant (upload) est plus bas que le débit descendant (download). C'est la méthode DSL la plus commune pour le grand public, qui « consomme » internet, télécharge des pages et des données mais n'a pas besoin de téléverser grand chose. 14, 233, 234

ATM
Asynchronous Transfer Mode 25, 38, 229, 233

Auto-MDIX
Automatic Medium-Dependent Interface Crossover 6, 138, 139, 233

Autonomous System Boundary Router
En terminologie OSPF, désigne le routeur situé entre un domaine de routage OSPF et un réseau non OSPF. 87, 233, 235

Baby giant
Trame dont la longueur dépasse légèrement les 1500 octets autorisés. 140, 233

Backup Designated Router
Routeur désigné de secours. Élu par le LSP Hello. 79, 233, 236

Bande passante
Capacité d'un support à transporter des données. En anglais, *bandwidth*. Mesurée en fonction du nombre de bits pouvant être transmis en une seconde (bit/s). Parfois à tort considérée comme la vitesse à laquelles circulent les bits, ce qui est faux puisqu'ils circulent à la vitesse de l'électricité. 22, 25, 29, 39, 42, 46, 47, 72, 76, 84, 85, 91, 95, 207, 233, 240

BD
Base de Données 9, 121, 122, 124, 128, 188, 189, 192, 193, 227, 233, 254

BDR
Backup Designated Router 79–83, 85, 87, 233, 236, *Glossary :* Backup Designated Router

BGP
Border Gateway Protocol 75, 233, 236

BGP-MP
Border Gateway Protocol — Multi-Protocol 233

BID
Bridge ID 42–44, 233, 236, 237, *Glossary :* Bridge ID

BIOS
Basic Input/Ouput System 107, 109, 161, 174, 233, 249

Bit
Binary digit. Chiffre binaire : soit 0, soit 1. 19, 24, 27, 31, 33, 42, 43, 53, 56–62, 67, 70, 71, 73, 79, 83, 97, 98, 103, 104, 112, 113, 141, 167, 205, 206, 233, 234, 236, 237, 240, 241, 243, 250

BNC
Bayonet Neill-Concelman 21, 233

BOOTP
BOOTstrap Protocol 109, 233

BPDU
Bridge Protocol Data Unit 2, 42–46, 52, 233, 236, 237, *Glossary :* Bridge Protocol Data Unit

Bridge
Switch n'ayant que deux ports. Dans le cadre du STP, on parle aussi de bridge pour dire switch, parce que c'était la terminologie dans les années 80, quand le STP a été inventé. 2, 42–44, 233, 253

Bridge Protocol Data Unit
Trame utilisée par des switchs pendant les calculs STP pour partager leurs informations. 2, 233, 236

Bridge ID
Numéro qui identifie un switch ayant envoyé un BPDU. Chaque BPDU contient un BID pour identifier le switch qui l'a émis. 42, 233, 236

Broadcast
Un message de broadcast est un message qui s'adresse à tous les hôtes d'un même réseau. Une adresse de broadcast est une adresse destinée à tous les hôtes du même réseau. Il y a deux types de broadcast IP :

1. broadcast limité : ne passe pas les routeurs. Il vaut `255.255.255.255`.
2. broadcast dirigé : passe les routeurs. Il vaut la dernière adresse d'un sous réseau (par exemple `192.168.1.255`).

En MAC il s'agit d'une adresse où les 48 bits sont à 1 : FF:FF:FF:FF:FF:FF. 2, 3, 5, 28–30, 39, 41–43, 51, 52, 54, 55, 58, 59, 64, 71, 86, 105, 106, 108, 109, 122, 130, 195, 230, 233, 255

CAN
Convertisseur Analogique Numérique 111, 233

Canonical Format Identifier
Identificateur de 1 bit qui permet aux trames Token Ring d'être transportées sur les liaisons ethernet. 33, 233, 237

CAST
Carlisle Adams et Stafford Tavares 233

CDP
Cisco Discovery Protocol 3, 52, 233, 237, *Glossary :* Cisco Discovery Protocol

CEF
Cisco Express Forwarding 72, 233

CFI
Canonical Format Identifier 33, 233, 237, *Glossary :* Canonical Format Identifier

CGN
Carrier Grade NAT 219, 233

CIDR
Classless Inter-Domain Routing 54, 233

CIFS
Common Internet FileSystem 184, 233

Cisco Discovery Protocol
Protocole activé par défaut sur les équipements Cisco. Utilise la couche 2 pour découvrir des équipements adjacents et auto-configurer un équipement en fonction. 52, 233, 237

Cisco Internetwork Operating System
OS présent dans les switchs et routeurs Cisco. 45, 233, 244

CLI
Command Line Interface 201, 233

CMIP
Common Management Information Protocol 123, 125, 126, 233, 238

CMIS
Common Management Information Service 125, 233

CMOT
CMIP over TCP/IP 126, 233

CNG
Confort Noise Generation 113, 233

codec
COder/DECoder 5, 110–114, 117, 120, 233

Common Spanning Tree
Autre nom du STP `IEEE 802.1D`. 45, 233, 238

CPU
Central Processing Unit 42, 71, 72, 78, 113, 131, 233

CRC
Cyclic Redundancy Check 27, 28, 140, 141, 233

CSMA/CA
Carrier-Sense Multiple Access / Collision Avoidance 25, 233

CSMA/CD
Carrier-Sense Multiple Access / Collision Detection 25, 233

CST
Common Spanning Tree 45, 233, 238, *Glossary :* Common Spanning Tree

CSV
Comma-Separated Values 170, 233

DAD
Duplicate Address Detection 63, 65, 233

DAI
Dynamic ARP Inspection 51, 233, 238, *Glossary :* Dynamic ARP Inspection

DAP
Directory Access Protocol 144, 233, 238, *Glossary :* Directory Access Protocol

DataBase Description
Description de la base de données. Type de paquet LSP, envoyé par les routeurs utilisant OSPF. 80, 233, 239

Datagramme
PDU de la couche Transport (couche 4 du modèle OSI) pour le protocole UDP. 5, 95, 102, 103, 109, 233

DB
DataBase 188, 233

DBD
DataBase Description 79–81, 233, 239, 246, *Glossary :* DataBase Description

DC
Domain Component 145, 233

DDOS
Distributed Denial of Service 195, 233

Denial of Service
Déni de service. Attaque visant à rendre indisponible un service, en général en le bombardant de requêtes pour qu'il ne puisse plus les gérer. 51, 195, 233, 239

DES
Data Encryption Standard 199, 231, 233

Designated Router
Routeur désigné. Élu par le LSP Hello. 4, 233, 239

DHCP
Dynamic Host Configuration Protocol 3, 5, 46, 51, 55, 58, 61, 62, 66, 102, 105–109, 233, 239, *Glossary :* Dynamic Host Configuration Protocol

Diaphonie
En anglais, *crosstalk*. Il s'agit de la perturbation causée par les champs électriques ou magnétiques sur le câble adjacent. 20, 233

Directory Access Protocol
Protocole utilisé originellement pour les annuaires X.500 de l'ITU. Complexe, il a été remplacé par LDAP. 144, 233, 238

DIT
Directory Information Tree 7, 150, 233

DMVPN
Dynamic Multipoint VPN 11, 230, 231, 233

DN
Distinguished Name 145–149, 151, 154, 156, 233, 252

DNAT
Destination Network Address Translation 233

DNS
Domain Name System 51, 61, 62, 96, 102, 104, 107–109, 144, 160, 161, 164, 233

DoS
Denial of Service 51, 195, 233, 239, *Glossary :* Denial of Service

DR
Designated Router 4, 79–83, 85, 87, 233, 239, *Glossary :* Designated Router

DRAM
Dynamic Random-Access Memory 131, 233

DS
Differentiated Services 56, 60, 233

DSA
Digital Signature Algorithm 199, 233

DSCP
Differentiated Services Code Point 56, 233

DSE
Directory Service Entry 150, 233

DSL
Digital Subscriber Line 14, 15, 233–235, 254, 256

DTP
Dynamic Trunking Protocol 2, 36, 37, 233, 240, *Glossary :* Dynamic Trunking Protocol

DTX
Discontinuous Transmission 113, 233

Dynamic ARP Inspection
Protocole de mitigation des attaques ARP poisoning et ARP spoofing. 51, 233, 238

Dynamic Host Configuration Protocol
Protocole utilisé sur un réseau IP où un serveur DHCP assigne des adresses IP aux hôtes du réseau de manière dynamique. 5, 233, 239

Dynamic Trunking Protocol
Protocole propriétaire Cisco pour configurer automatiquement les VLAN sur les switchs de la topologie. 36, 233, 240

Débit
Mesure du transfert de bits sur le support pendant une période donnée. Ne correspond en général pas à la bande passante à cause de plusieurs facteur :
— la quantité de trafic
— le type de trafic
— la latence créée par le nombre de périphériques rencontrés entre la source et la destination
14, 47, 49, 76, 110, 112–114, 139, 233

ECN
Explicit Congestion Notification 56, 233

EGP
Exterior Gateway Protocol 75, 126, 233, 240, *Glossary :* Exterior Gateway Protocol

Egress
Port de sortie d'un switch ou d'un routeur, par lequel une trame sort de l'appareil. 29, 70, 71, 233

EIA
Electronic Industries Association 20, 233

EIGRP
Enhanced Interior Gateway Routing Protocol 67, 69, 73–76, 233

EMI
Electro-Magnetic Interference 20, 22, 233

EP
End Point 120, 233

ESP
Encapsulation Security Payload 231–233

EtherChannel
Agrégation de liens pour regrouper de multiples liens physiques ethernet. 3, 46–49, 233, 251

Ethernet
Protocole sur la couche Liaison (couche 2 du modèle OSI). Utilise des connexions câblées, qu'elles soient coaxiales, en fibre optique, ou de paires torsadées en cuivre. Défini dans les standards 802.2 et 802.3 de l'IEEE. 1, 6, 25, 26, 28–33, 41, 47, 64, 71, 73, 74, 76, 79, 81, 89, 101, 112, 122, 130, 141, 233, 237, 241, 253, 257

EUI
Extended Unique Identifiers 3, 62, 63, 233

Exterior Gateway Protocol
Famille de protocoles de routage qui détermine la disponibilité d'un réseau entre deux systèmes autonomes (AS). Les protocoles de cette famille utilisent les protocoles IGP pour résoudre les routes dans un AS. 75, 233, 240

Extranet
Fournit un accès à des personnes extérieures à l'organisation mais qui ont besoin d'accéder aux données internes. 14, 233

FAI
Fournisseur d'Accès Internet 14, 15, 31, 54, 59, 72, 75, 109, 217, 219, 222, 233

FCS
Frame Check Sequence 24, 27–29, 141, 233

FDDI
Fiber Distributed Data Interface 31, 38, 233

FFTP cable
Foiled Foiled Twisted Pair 21, 233, 241, *Glossary :* Foiled Foiled Twisted Pair

FIB
Forwarding Information Base 72, 233

File Transfer Protocol
Protocole de transfert de fichiers qui utilise TCP sur les ports 20 et 21. 5, 233, 242

Firewall
Pare-feu. Ensemble de règles pour bloquer ou autoriser des connexions, dans un sens (sortie) comme dans l'autre (entrée). 109, 196, 197, 233

FK
Foreign Key 233

Flag
Bit qui est défini pour être soit actif soit inactif. 97, 233

Foiled Foiled Twisted Pair
Câble à paires doublement écrantées : chaque paire est écrantée et le tout est écranté sous la gaine. 21, 233, 241

Foiled Twisted Pair
Câble à paires non blindées mais écrantées (avec une feuille d'aluminium). 21, 233, 242

FTP
File Transfer Protocol 5, 18, 96, 102, 103, 108, 109, 197, 204, 220, 225, 233, 242, 257, 258, *Glossary :* File Transfer Protocol

FTP cable
Foiled Twisted Pair 21, 233, 242, *Glossary :* Foiled Twisted Pair

FTTH
Fiber To The Home 22, 233

Full-duplex
Sur un même câble, les données peuvent être échangées dans les deux directions en même temps. Pas de domaine de collision. 24, 26, 29, 132, 139, 141, 233

GDMO
Guidelines for the Definition of Managed Objects 124, 125, 233

Gestion Libre de Parc Informatique
Logiciel de gestion du parc informatique d'une entreprise. 233, 242

GID
Group IDentifier 183, 233

GLPI
Gestion Libre de Parc Informatique 233, 242, *Glossary :* Gestion Libre de Parc Informatique

GPO
Group Policy Object 159, 160, 162–165, 170, 233

GRE
Generic Routing Encapsulation 11, 228, 230, 231, 233, 248

GSM
Global System for Mobile communications 114, 233

GUA
Global Unicast Address 58, 59, 233

GUI
Graphical User Interface 201, 233

GUID
Globally Unique IDentifier 233

Half-duplex
Sur un même câble, les données sont échangées dans un sens à la fois. Chaque segment est dans son propre domaine de collision. 24, 29, 132, 139, 141, 233

HDLC
High-level Data Link Control 25, 233

Hextet
Deux octets : donc 16 bits. 57, 233

HIDS
Host-based Intrusion Detection System 233

HTTP
HyperText Transfer Protocol 18, 31, 96, 104, 116, 117, 196, 204, 213, 229, 233, 243

HTTPS
HTTP Secure 31, 104, 213, 229, 233

IANA
Internet Assigned Numbers Authority 53, 104, 233

IAX
Inter Asterisk eXchange 115, 233

IBM
International Business Machines Corporation 233, 257

ICMP
Internet Control Message Protocol 4, 56, 57, 60, 61, 64, 65, 106, 126, 197, 218, 233, 257

IDE
Integrated Development Environment 166, 233

IDEA
International Data Encryption Algorithm 199, 233

IDS
Intrusion Detection System 233, 243

IEC
International Electrotechnical Commission 21, 233

IEEE
Institute of Electrical and Electronics Engineers 20, 27, 31, 32, 37, 45, 48, 62, 233, 238, 241, 245, 249, 251, 252, 257, 260

IETF
Internet Engineering Task Force 20, 115, 116, 119, 126, 224, 231, 233

IGP
Interior Gateway Protocol 75, 233, 241, 243, *Glossary :* Interior Gateway Protocol

IGRP
Interior Gateway Routing Protocol 75, 233

IKE
Internet Key Exchange 232, 233

IMAP
Internet Message Access Protocol 229, 233, 243, 251, *Glossary :* Internet Message Access Protocol

Ingress
Port d'entrée d'un switch ou d'un routeur, par lequel une trame entre dans l'appareil. 29, 233

Interior Gateway Protocol
Famille de protocoles de routage qui peuvent échanger des informations de routage avec des AS. 75, 233, 243

Internet
Ensemble des réseaux sur la toile. Les LAN sont reliés entre eux par des WAN et les WAN sont connectés les uns aux autres.
Des organismes permettant de maintenir la structure et les protocoles incluent l'IETF, l'ICANN, l'IAB. 14, 21, 54, 58, 75, 87, 109, 111, 159, 164, 166, 182, 197, 211, 215, 217–220, 223, 225, 229, 230, 233, 235, 260

Internet Message Access Protocol
Protocole de messagerie (e-mail) qui utilise le port 143. Par rapport à POP3, IMAP laisse les mails reçus sur le serveur. Il les copie localement dans un cache. Il prend compte des actions effectuées par l'utilisateur (mails lus, supprimés, etc.). 229, 233, 243

Intranet
Connexion privée de LAN ou de WAN qui appartient à une organisation et offre un accès sous réserve d'une autorisation. 14, 219, 233

IOS
Cisco Internetwork Operating System 45, 47, 52, 69, 72, 84, 131, 141, 210, 233, 244, *Glossary :* Cisco Internetwork Operating System

IP
Internet Protocol 1, 3, 5, 9–11, 18–20, 26–28, 30, 33, 34, 41, 46, 50–74, 76, 79, 80, 87, 89–94, 101, 104–106, 108, 110–112, 114–120, 124–127, 130, 134–137, 140, 160, 164, 195, 196, 198, 199, 204–207, 210, 211, 215, 217–219, 222, 224, 225, 228, 229, 231, 233, 234, 237, 238, 240, 244, 246, 248, 249, 251, 255, 257, 259

IPBX
IP Private Branch eXchange 110, 116, 233

IPS
Intrusion Prevention System 233

IPSec
IP Security 11, 219, 228–233

IPX
Internetwork Packet eXchange 126, 233

IS-IS
Intermediate System to Intermediate System 73, 75, 233

ISDN
Integrated Services Digital Network 112, 233

ISL
Inter-Switch Link 37, 38, 233

ISN
Initial Sequence Number 99, 233

ISO
International Organization for Standardization 20, 21, 124, 125, 233

ISP
Internet Service Provider 14, 233, 260

ITU
International Telecommunication Union 20, 115, 119, 233, 239

Jumbo
Trame dont la longueur dépasse les 1500 octets autorisés. 233

L2TP
Layer 2 Tunneling Protocol 228, 233

LACP
Link Aggregation Control Protocol 3, 47–49, 233, 245, *Glossary :* Link Aggregation Control Protocol

LAN
Local Area Network 3, 14, 15, 23, 25, 28, 38, 46, 50, 76, 84, 107, 221, 233, 244, 245, 249, 257, 259, 260, *Glossary :* Local Area Network

LANE
LAN Emulation 38, 233

LC
Lucent Connector 23, 233

LDAP
Lightweight Directory Access Protocol 7, 144–147, 149, 150, 153–155, 157, 233, 234, 239, 245, *Glossary :* Lightweight Directory Access Protocol

LDAPS
LDAP over SSL 154, 233, 234, 245, *Glossary :* LDAP over SSL

LDIF
LDAP Data Interchange Format 7, 153, 155, 233, 245, *Glossary :* LDAP Data Interchange Format

LED
Light-Emitting Diode 6, 23, 131, 132, 141, 233

Lightweight Directory Access Protocol
Protocole standardisé pour maintenir un annuaire, qui centralise les informations d'une entreprise. LDAP fonctionne sur le port TCP 389. 7, 233, 245

Link Aggregation Control Protocol
Protocole pour créer des agrégations de ports physiques pour n'avoir qu'un port logique, de la même façon que le PAgP de Cisco. La négociation de ports se fait de la même façon. Contrairement à PAgP, LACP est un standard de la spécification IEEE 802.1AX, et donc compatible sur des switchs de plusieurs fabriquants. 3, 233, 245

Link-State Acknowledgement
Acquittement de la base de données. Type de paquet LSP, envoyé par les routeurs utilisant OSPF. 80, 233, 247

Link-State Advertisement
Annonce d'état de lien. Utilisé par les routeurs utilisant OSPF pour s'échanger des informations. 78, 233, 246

Link-State DataBase
Base de données d'état de lien. Utilisé par les routeurs utilisant OSPF pour construire la table de topologie. 77, 233, 247

Link-State Packet
Paquet d'état de lien. Paquet de LSA, pouvant être l'un de 5 types : Hello, DBD, LSR, LSU ou LSAck. 79, 233, 247

Link-State Request
Requête d'état de lien. Type de paquet LSP, envoyé par les routeurs utilisant OSPF. 80, 233, 247

Link-State Update
Mise à jour d'état de lien. Type de paquet LSP, envoyé par les routeurs utilisant OSPF. 80, 233, 247

LIR
Local Internet Registry 233

LLA
Link Local Address 58, 233

LLC
Logical Link Control 24, 26, 233, 246, *Glossary :* Logical Link Control

Local Area Network
Infrastructure de réseau qui couvre une zone géographique restreinte.
— Administré par une seule entreprise ou une seule personne.
— Bande passante à haut débit.
14, 233, 245

Logical Link Control
Sous-couche de la couche Liaison (couche OSI 2) qui communique entre le logiciel de réseau dans les couches supérieures et le matériel du périphérique dans les couches inférieures. Elle place dans la trame des informations indentifiant quel protocole de réseau est utilisé pour la trame. Cette information permet à de multiples protocoles de couche 3 comme IPv4 et IPv6 d'utiliser la même interface et le même support réseau. 24, 233, 246

Logical Volume Manager
Gestionnaire de volumes logiques. Permet de regrouper des volumes indépendamment des disques et partitions physiques. 8, 233, 247

LSA
Link-State Advertisement 78, 80–82, 85, 233, 246, *Glossary :* Link-State Advertisement

LSAck
Link-State Acknowledgement 79, 80, 85, 233, 246, 247, *Glossary :* Link-State Acknowledgement

LSDB
Link-State DataBase 77, 78, 80, 233, 247, *Glossary :* Link-State DataBase

LSP
Link-State Packet 79, 233, 236, 238, 239, 246, 247, *Glossary :* Link-State Packet

LSR
Link-State Request 79–81, 85, 233, 246, 247, *Glossary :* Link-State Request

LSU
Link-State Update 79–81, 85, 233, 246, 247, *Glossary :* Link-State Update

LUKS
Linux Unified Key Setup 203, 233

LV
Logical Volume 8, 176–178, 182, 233

LVM
Logical Volume Manager 8, 172, 176–178, 180, 182, 233, 247, *Glossary :* Logical Volume Manager

MAC
Media Access Control 1–3, 18, 24, 26–28, 42, 43, 45, 50, 51, 61–65, 71, 105, 108, 122, 130, 140, 233, 234, 237, 247, 249, 250, *Glossary :* Media Access Control

MAN
Metropolitan Area Network 233, 247, *Glossary :* Metropolitan Area Network

Maximum Segment Size
Taille maximale qu'un hôte peut recevoir pour un segment, en octets. N'inclut pas l'en-tête TCP. 101, 233, 248

MCU
Multipoint Control Unit 120, 233

MD5
Message-Digest Algorithm 128, 199, 232, 233

Media Access Control
Sous-couche de la couche Liaison (couche OSI 2) responsable de l'encapsulation et du contrôle d'accès au support. Il procure l'adressage de la couche Liaison et est intégré dans diveres technologies de couche Physique. 24, 233, 247

MEGACO
MEdia GAteway COntrol 119, 233

Metropolitan Area Network
Infrastructure de réseau qui couvre une zone géographique équivalente à un campus ou une petite ville. 233, 247

MG
Media Gateway 119, 233

MGC
Media Gateway Controller 119, 233

MGCP
Media Gateway Control Protocol 5, 115, 119, 233

mGRE
Multipoint Generic Routing Encapsulation 230, 233

MIB
Management Information Base 6, 124–130, 233, 234, 255

MISTP
Multiple Instance Spanning Tree Protocol 45, 233, 248, *Glossary :* Multiple Instance Spanning Tree Protocol

MitM
Man-in-the-Middle 51, 99, 195, 233

MMF
Multi-Mode optical Fiber 23, 233

MMUSIC
Multiparty MUtimedia SessIon Control 116, 233

MOC
Managed Object Class 6, 124, 233

MOI
Managed Object Instance 124, 233

MOS
Mean Opinion Score 5, 114, 233

MOTD
Message Of The Day 134, 233

MPD
Modèle Physique de Données 9, 189, 190, 193, 233

MPLS
MultiProtocol Label Switching 229, 233

MSS
Maximum Segment Size 101, 233, 248, *Glossary :* Maximum Segment Size

MST
Multiple Spanning Tree 45, 233, 248, *Glossary :* Multiple Spanning Tree

MSTP
Multiple Spanning Tree Protocol 45, 233, 248, 249, *Glossary :* Multiple Spanning Tree Protocol

MTU
Maximum Transmission Unit 18, 38, 101, 233

Multicast
Une adresse destinée aux hôtes faisant partie d'un groupe. Les destinataires faisant partie du groupe se voient attribuer une adresse IP de groupe multicast. En IPv4

celles-ci correspondent à la classe D (voir 5.2.2). En IPv6 elles commencent par FF00::/8. Pour une adresse MAC associée à une adresse IPv4, c'est une valeur spéciale commançant par 01:00:5E et dont les trois derniers octets sont directement convertis en hexadécimal. Pour une adresse MAC associée à une adresse IPv6, elle commence par 33:33. 4, 30, 38, 39, 42, 52, 54, 55, 58, 64, 80, 82, 86, 130, 228, 230, 231, 233

Multiple Instance Spanning Tree Protocol
Implémentation Cisco de STP ayant inspiré le standard IEEE MSTP (voir 4.10.5). 45, 233, 248

Multiple Spanning Tree
Implémentation Cisco du MSTP (voir 4.10.5). 45, 233, 248

Multiple Spanning Tree Protocol
Pointe plusieurs VLAN dans une seule instance STP (voir 4.10.5). 45, 233, 248

NA
Neighbour Advertisement 65, 233

NAT
Network Address Translation 10, 54, 57, 58, 215–224, 231, 233, 237, 239, 249, 255

NAT-PT
Network Address Translation — Protocol Translation 224, 233

NAT64
Network Address Translation IPv6 to IPv4 11, 57, 224, 233

NDP, ND
Neighbour Discovery Protocol 65, 233

NetBIOS
Permet à des applications de différents ordinateurs de communiquer sur un LAN. 107, 109, 161, 233

Network File System
Protocole historique de partage de fichiers entre machines UNIX. 8, 233, 249

Network Policy Server
Service Windows qui se base sur le protocole RADIUS pour autoriser ou non l'accès réseau à des postes clients ou à des utilisateurs. 164, 233, 249

NFS
Network File System 8, 182–184, 186, 233, 249, *Glossary :* Network File System

NIC
Network Interface Card 140, 233

NIS
Network Information Services 144, 233

NMS
Network Management Station 6, 122–126, 129, 233

NPS
Network Policy Server 164, 233, 249, *Glossary :* Network Policy Server

NS
Neighbour Solicitation 65, 71, 233

NTP
Network Time Protocol 158, 233

NVRAM
Non Volatile RAM 38, 39, 134, 233

Octet
Un groupe de huit bits. 26, 27, 32, 53, 55, 57, 59, 60, 62, 97, 99–102, 140, 233, 236, 243, 245, 247, 250, 253

OID
Object IDentifier 6, 124, 125, 152, 233

Organizationally Unique Identifier
Trois premiers octets de l'adresse MAC correspondant à l'identifiant du revendeur du matériel. 27, 233, 250

OS
Operating System 54, 104, 122, 160, 174, 196, 233, 238, 251, 258

OSI
Open Systems Interconnection 1, 18–20, 26, 95, 115, 130, 204, 205, 233, 238, 241, 246, 247, 250, 254, 257, 260

OSPF
Open Shortest Path First 4, 67, 69, 73–84, 87, 93, 205, 228, 233, 236, 238, 246

OU
Organisational Unit 145, 158–163, 169, 233

OUI
Organizationally Unique Identifier 27, 62, 233, 250, *Glossary :* Organizationally Unique Identifier

P2P
Peer To Peer 233, 250, *Glossary :* Peer To Peer

PVID
Port VLAN ID 33, 233, 250, *Glossary :* Port VLAN ID

PABX
Private Branch eXchange 119, 233

PAgP
Port Aggregation Protocol 3, 47, 48, 233, 245, 250, *Glossary :* Port Aggregation Protocol

Paquet
PDU de la couche Réseau (couche 3 du modèle OSI). 4, 10, 18, 24, 27, 39, 41, 47, 48, 54, 56, 57, 60, 61, 64–73, 75–77, 79–81, 84–89, 91, 100, 102, 110–112, 115, 122, 199, 204, 205, 207, 208, 211, 218–220, 223, 224, 230–233, 235, 238, 246, 255, 257

PAT
Port Address Translation 10, 217–219, 222–224, 233

PC
Personal Computer 34, 116, 122, 141, 233

PCM
Pulse Code Modulation 111, 233

PDU
Protocol Data Unit 2, 6, 26, 27, 95, 126, 127, 129, 233, 236–238, 250, 254, 257

Peer To Peer
Les terminaux peuvent avoir à la fois le rôle du client et du serveur. Avantage : plus résilient, pour les attaques, pas de point central à attaquer. Inconvénient : moins de vie privée (l'adresse IP est partagée), d'avantage d'accès aux données pour quiconque serait intéressé. 233, 250

Per-VLAN Spanning Tree
Version de STP dans un environnement contenant plusieurs VLAN (voir 4.10.5). 43, 233, 252

PK
Primary Key 193, 233

POP3
Post Office Protocol version 3 104, 229, 233, 244, 251, *Glossary :* Post Office Protocol version 3

POP3S
Post Office Protocol version 3 Secure 104, 233, 251, *Glossary :* Post Office Protocol version 3 Secure

Port VLAN ID
Numéro d'identification correspondant au numéro du VLAN natif. Si le VLAN natif est défini comme VLAN 99, alors le PVID sera de 99. Si le VLAN natif n'a pas été configuré, le PVID sera de 1. 33, 233, 250

Port Aggregation Protocol
Protocole propriétaire Cisco pour créer des EtherChannels automatiquement par négociation de ports. 3, 233, 250

Portable Operating System Interface
Standard de l'IEEE pour maintenir la compatibilité entre OS. 233, 251

POSIX
Portable Operating System Interface 214, 233, 251, *Glossary :* Portable Operating System Interface

POST
Power-On Self-Test 131, 233

Post Office Protocol version 3
Protocole de messagerie (e-mail) qui utilise le port 110. Par rapport à IMAP, POP3 ne laisse pas par défaut les mails reçus sur le serveur. Il ne prend pas non plus compte des actions effectuées par l'utilisateur (mails lus, supprimés, etc.). 104, 233, 251

Post Office Protocol version 3 Secure
POP3 sur TLS ou SSL, utilisant le port TCP 995. 104, 233, 251

PPP
Point to Point Protocol 25, 71, 233

Prise RJ11
Type de prise pour un câble de cuivre utilisé pour le téléphone. Cette prise a la même forme que la prise RJ45 mais en plus petit. 233

PSK
Pre-Shared Key 232, 233

PV
Physical Volume 176–178, 233

PVST
Per-VLAN Spanning Tree 43, 45, 233, 252, *Glossary :* Per-VLAN Spanning Tree

QoS
Quality of Service 13, 14, 33, 112, 204, 233

RA
Router Advertisement 46, 64, 65, 136, 233

RADIUS
Remote Authentication Dial-In User Service 11, 227, 233, 249

RAID
Redundant Array of Independant Disks 8, 172–178, 180, 182, 233, 252, *Glossary :* Redundant Array of Independant Disks

RAM
Random-Access Memory 38, 74, 77, 134, 233, 250

Rapid Spanning Tree Protocol
Amélioration de STP, standard `IEEE 802.1w` (voir 4.10.5). 2, 233, 253

RC
Rivest Cipher 199, 233

RDN
Relative Distinguished Name 145, 148, 149, 151, 233

Redundant Array of Independant Disks
En français regroupement redondant de disques indépendants. Ensemble de techniques qui permet de répartir le stockage de données sur plusieurs disques. Cette répartition est transparente pour l'utilisateur qui ne voit qu'un volume de stockage unique. 176, 233, 252

RFC
Request For Comments 54, 65, 106, 115, 116, 119, 124, 125, 129, 130, 145, 156, 215, 219, 224, 231, 233, 234, 258

RFI
Radio-Frequency Interference 20, 22, 233

RID
Registered IDentifier 158, 233

RIP
Routing Information Protocol 75, 76, 233

RIR
Regional Internet Registry 53, 59, 233

RJ45
Type de prise pour un câble de cuivre utilisé pour ethernet. 21, 25, 233, 252

RMon
Remote Monitoring 6, 125, 126, 129, 130, 233

RNIS
Réseau Numérique à Intégration de Service 112, 114, 119, 233

ROM
Read-Only Memory 27, 131, 233

Root bridge
En français, pont racine, switch sélectionné par le protocole STP. 2, 42–44, 233

RS
Router Solicitation 64, 65, 233

RSA
Rivest-Shamir-Adleman 137, 199, 232, 233

RSTP
Rapid Spanning Tree Protocol 2, 45, 46, 233, 253, *Glossary :* Rapid Spanning Tree Protocol

RTC
Real-Time Communication 112, 233

RTCP
Real-Time Control Protocol 5, 115, 233

RTP
Real-Time Protocol 5, 115, 233

Runt
Trame dont la longueur est inférieure aux 64 octets minimum autorisés. 140, 233

Réseau d'extrémité
En anglais, *stub network*. Réseau qui n'est accessible que par une seule route. Le routeur n'est connecté qu'à un seul autre routeur. 72, 233

SA
Security Association 232, 233

SACK
Selective Acknowledgement 100, 233

SASL
Simple Authentication and Security Layer 144, 147, 155, 233

SC
Subscriber Connector 23, 233

SCCP
Skinny Client Control Protocol 115, 233

SDP
Session Description Protocol 117, 233

SDSL
Symmetric DSL 15, 233, 254, *Glossary :* Symmetric DSL

SEAL
Software-optimized Encryption Algorithm 231, 233

Secure Shell
Ouvre un shell à distance de manière sécurisée en chiffrant toutes les données échangées, contrairement à Telnet. 11, 233, 256

Segment
PDU de la couche Transport (couche 4 du modèle OSI) pour le protocole TCP. 18, 95–97, 99–102, 233, 247, 258, 259

SFD
Start Frame Delimiter 26, 233

SFTP cable
Shielded and Foiled Twisted Pair 21, 233, 254, *Glossary :* Shielded and Foiled Twisted Pair

SGBD
Système de Gestion de Base de Données 188, 233, 254

SGBD-R
SGBD Relationnel 188, 233

SGMP
Simple Gateway Monitoring Protocol 233, 255

SHA
Secure Hash Algorithm 199, 232, 233

Shielded and Foiled Twisted Pair
Câble à paires blindées et écrantées. 21, 233, 254

Shielded Twisted Pair
Câble à paires blindées mais non écrantées. 21, 233, 256

SI
Système d'information 144, 233

SID
Security IDentifier 158, 160, 233

Simple Mail Transfer Protocol
Protocole d'envoi de mail qui utilise le port 25. 96, 233, 255

Simple Mail Transfer Protocol Secure
SMTP sur TLS ou SSL, qui utilise les ports TCP 465 ou 587. 103, 233, 255

Simple Network Management Protocol
Protocole de gestion des périphériques réseaux.

— issu de Simple Gateway Monitoring Protocol (SGMP)
— standard du monde TCP/IP
— protocole le plus répandu
Fonctionne avec un Manager (qui a une base de données MIB) et un Agent. Utilise les ports UDP. 6, 233, 255

Simplex
Sur un même câble, les données sont échangées dans un sens seulement. 233

SIP
Session Initiation Protocol 5, 115–119, 233

SLAAC
StateLess Address Autoconfiguration 3, 61, 62, 233

SMB
Server Message Block 184, 233

SMF
Single-Mode optical Fiber 22, 233

SMI
Structure of Management Information 124, 125, 233

SMTP
Simple Mail Transfer Protocol 18, 96, 103, 116, 117, 164, 233, 254, 255, *Glossary :* Simple Mail Transfer Protocol

SMTPS
Simple Mail Transfer Protocol Secure 103, 233, 255, *Glossary :* Simple Mail Transfer Protocol Secure

SNAT
Source Network Address Translation 233

SNMP
Simple Network Management Protocol 5, 6, 31, 37, 39, 96, 102, 104, 120, 122, 123, 125–130, 233, 255, 258, *Glossary :* Simple Network Management Protocol

Socket
Combination d'une adresse IP et d'un port (`192.168.0.15:80`). 104, 233

SOCKS
Protocole IP qui échange les paquets entre un client et un serveur par l'intermédiaire d'un proxy. 11, 225, 233

Spanning tree
Protocole pour éviter les tempêtes de broadcast. 233

Spanning Tree Algorithm
Algorithme inventé par Radia Perlman en 1985 et utilisé par les calculs du protocole STP. 2, 233, 256

Spanning Tree Protocol
Protocole de couche 2 pour prévenir les boucles de commutation. 2, 233, 256

SPF
Shortest Path First 77–79, 81, 87, 233

SPX
Sequenced Packet eXchange 126, 233

SQL
Structured Query Language 188, 233

SSH
Secure Shell 6, 11, 31, 96, 103, 137, 217, 233, 256, 257, *Glossary :* Secure Shell

SSL
Secure Sockets Layer 154, 155, 229, 233, 234, 245, 251, 254

ST
Straight-Tip 23, 233

STA
Spanning Tree Algorithm 2, 42–44, 233, 256, *Glossary :* Spanning Tree Algorithm

StartTLS
Établit une connexion sécurisée entre un client et un serveur. À la connexion, le client et le serveur établissent une connexion sécurisée avant toute autre commande. 147, 154, 233

Stateful
Avec état, c'est-à-dire qui suit les informations. C'est le contraire de stateless. Un protocole *stateful* traite chaque requête en se rappelant des évènements et de requêtes précédentes. L'information dont se souvient un protocole *stateful* s'appelle l'*état* du système. 3, 61, 62, 96, 109, 197, 213, 219, 233

Stateless
Sans état, c'est-à-dire qui ne suit pas les informations. Un protocole *stateless* traite chaque requête indépendamment sans tenir compte des requêtes précédentes. 3, 61, 62, 102, 196, 197, 220, 233, 256

STP
Spanning Tree Protocol 2, 3, 41–47, 51, 52, 233, 237, 238, 249, 251–253, 255, 256, *Glossary :* Spanning Tree Protocol

STP cable
Shielded Twisted Pair 21, 233, 256, *Glossary :* Shielded Twisted Pair

SVI
Switch Virtual Interface 4, 6, 88, 89, 91, 92, 136, 137, 233, 256, *Glossary :* Switch Virtual Interface

Switch Virtual Interface
Port (ou interface) virtuel sur un switch. Par défaut il s'agit du VLAN1. 4, 233, 256

Symmetric DSL
Au contraire de l'ADSL, les débits ascendant et descendant sont égaux, permettant un téléversement aussi rapide qu'un téléchargement. Utilisé surtout en entreprise, où les besoins spécifiques justifient de devoir disposer d'un débit ascendant élevé. 15, 233, 254

TACACS
Terminal Access Controller Access-Control System 109, 233

TC
Topology Change 233

TCA
Topology Change Acknowledgement 233

TCP
Transmission Control Protocol 1, 4, 10, 18–20, 56, 95–103, 109, 114, 124, 126, 127, 149, 182, 205, 211, 213, 218, 219, 225, 229, 233, 238, 241, 245, 247, 251, 254, 255, 257, 259, *Glossary :* Transmission Control Protocol

TDM
Time Division Multiplexing 112, 233

Telnet
Ouvre un shell à distance de manière non sécurisée. Contrairement à SSH, ne chiffre pas les données échangées. 31, 103, 137, 197, 204, 233, 254

TFTP
Trivial FTP 18, 102, 109, 233, 257, *Glossary :* Trivial FTP

Three-way handshake
Procédé d'établissement d'une connexion TCP qui ressemble à une poignée de main en 3 étapes. Voir 6.2.1 98, 100, 101, 233

TIA
Telecommunications Industry Association 20, 233

TLS
Transport Layer Security 144, 147, 154, 155, 229, 233, 251, 254, 256

TLV
Type-Length Value 38, 233

ToIP
Telephony over IP 5, 110, 115, 116, 233

Token Ring
Réseau de LAN introduit par International Business Machines Corporation (IBM) en 1984 et standardisé en 1989 par l'IEEE `802.5`. Il a été supplanté par Ethernet. 31, 33, 38, 130, 233, 237

Traceroute
La commande `traceroute` utilise le TTL pour connaître tous les tronçons sur le chemin d'une destination. Elle commence par envoyer un paquet avec un TTL à 1. Le premier routeur rencontré va donc lui renvoyer une réponse ICMP de dépassement de délai. Puis la commande incrémente le TTL et renvoie le paquet, ce qui l'amène au tronçon suivant. Il fait de même jusqu'à arriver à la destination. 233

Trame
PDU de la couche Liaison (couche 2 du modèle OSI). 1, 2, 19, 24, 26–28, 31–33, 36, 38, 41, 42, 44, 45, 50–52, 71, 89, 112, 113, 122, 130, 140, 141, 233, 236, 237, 240, 244–246, 253

Transmission Control Protocol
Protocole en mode connecté, commence par un Three Way Handshake. Plus lent que UDP mais garantit la connexion en renvoyant les segments perdus. Défini dans la RFC 793. 4, 233, 257

Trap
Interruption. Notification non sollicitée envoyée par un équipement. Il s'agit de l'envoi de requêtes par des clients au serveur SNMP. 104, 123, 125–127, 129, 233

Trivial FTP
Comme FTP mais utilise UDP sur le port 69. 102, 233, 257

Trunk
Lien de point à point entre deux périphériques d'un réseau possédant plus d'un VLAN. Étend des VLAN à travers un réseau entier. Cela permet à des appareils connectés à différents switchs mais sur le même VLAN de communiquer sans passer par un routeur. 2, 31–33, 35–39, 46, 47, 49–51, 89–91, 94, 113, 233

TTL
Time To Live 28, 41, 56, 60, 65, 197, 233, 257

U/L
Universal/Local 62, 233

UA
Universal Alcatel 115, 117, 233

UDP
User Datagram Protocol 4, 5, 18, 19, 56, 95, 96, 102, 103, 109, 114, 115, 126, 127, 205, 218–220, 227, 233, 238, 255, 258, 259, *Glossary :* User Datagram Protocol

UID
User IDentifier 150, 183, 184, 233

ULA
Unicast Local Address 58, 224, 233

UML
Unified Modeling Language 8, 188, 190, 191, 233

Unicast
Une adresse pour un seul destinataire. 30, 39, 42, 54, 56, 58, 106, 109, 228, 231, 233, 235

UNIX
Famille d'OS dérivés du Unix d'AT&T. 9, 182, 184, 201, 233, 249

Unshielded Twisted Pair
Câble à paires non blindées. 21, 233, 259

URI
Uniform Resource Identifier 117, 233

URL
Uniform Resource Locator 7, 149, 233

User Datagram Protocol
Au contraire du TCP, UDP n'est pas connecté, donc il ne renvoie pas de segment perdu. En streaming vidéo par exemple TCP ne sert à rien, ce qui est perdu est perdu. Plus rapide que TCP. 4, 233, 258

UTP cable
Unshielded Twisted Pair 21, 233, 259, *Glossary :* Unshielded Twisted Pair

VAD
Voice Activity Detection 113, 233

VDI
Virtual Disk Image 171, 233, 259, *Glossary :* Virtual Disk Image

VG
Volume Group 176–178, 233

VID
VLAN ID 33, 233, 250, 251, 259, *Glossary :* VLAN ID

Virtual Disk Image
(extension de fichier `.vdi`) Format de disque virtuel créé par VirtualBox. Bien qu'il ne soit pas compatible avec d'autres logiciels de virtualisation, on le choisit quand on travaille dans VirtualBox parce qu'il y fonctionne mieux.
171, 233, 259

Virtual LAN
LAN virtuel. Assure la segmentation et favorise la flexibilité dans un réseau commuté. Un groupe d'appareils dans un VLAN communiquent comme s'ils étaient reliés au même câble.
Les VLAN reposent sur des connexions logiques et non physiques. Chaque VLAN est considéré comme un réseau logique distinct. 2, 233, 259

Virtual Machine Disk File
(extension de fichier `.vmdk`) Format de disque virtuel créé par VMware, et compatible par le plus grand nombre de logiciels de virtualisation. 171, 233, 259

VLAN
Virtual LAN 2–4, 30–40, 43, 45, 49–52, 88–94, 112, 136, 233, 234, 240, 249–252, 258–260, *Glossary :* Virtual LAN

VM
Virtual Machine 160, 233

VMDK
Virtual Machine Disk File 171, 233, 259, *Glossary :* Virtual Machine Disk File

VoIP
Voice over IP 5, 14, 31, 33, 96, 102, 110–112, 114, 204, 219, 233

VPN
Virtual Private Network 11, 196, 228–233, 239

VTI
Virtual Tunnel Interface 11, 231, 233

VTP
VLAN Trunking Protocol 2, 31, 37–41, 233, 260, *Glossary :* VLAN Trunking Protocol

VTY
Virtual Teletype 10, 210, 233

WAN
Wide Area Network 14, 15, 23, 25, 233, 244, 260, *Glossary :* Wide Area Network

WAP
Wireless AP 233

Wide Area Network
Infrastructure de réseau qui couvre une vaste zone géographique.
— Généralement géré par des fournisseurs de services (SP) ou des fournisseurs de services internet (ISP).
— Relient des LAN.
— Liaisons à plus bas débit entre les réseaux locaux.
14, 233, 260

WiFi
Wireless Fidelity 1, 23, 25, 233, 260, *Glossary :* Wireless Fidelity

Windows Server Update Services
Service Windows qui permet de déployer les mises à jour Windows sur un réseau. 164, 233, 260

WINS
Windows Internet Name Service 107, 233

Wireless Fidelity
Protocole sur la couche Liaison (couche 2 du modèle OSI). Utilise des connexions WLAN. Défini dans les standards 802.11 de l'IEEE. 1, 233, 260

WLAN
Wireless LAN 23, 233, 260

World Wide Web
L'ensemble des sites web forme le World Wide Web. C'est en fait une sous-partie d'internet. 217, 229, 233

WPAN
Wireless Personal Area Network 23, 233

WSUS
Windows Server Update Services 164, 233, 260, *Glossary :* Windows Server Update Services

WWAN
Wireless WAN 23, 233

XAMPP
X Apache MySQL PHP Perl 233