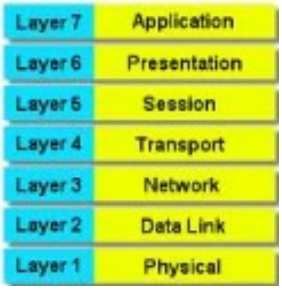
Lycée Gaston CRAMPE - 40800 AIRE/ADOUR

Section de Techniciens Supérieurs en Cybersécurité Informatique réseaux et ELectronique :
Option A : Informatique et Réseaux

STS CIEL-IR

Les Réseaux Locaux (LAN : Local Area Network) Couches Hautes

Étudiant :



Enseignant : Jean-Claude CABIANCA

LES RÉSEAUX LOCAUX
LAN : Local Area Network
Couches Hautes

Fiche 10 : Couche 4 – Transport

Fiche 11 : Couche 7 – Application

Fiche 12 : Introduction à la Cybersécurité

Fiche 13 : Cryptographie pour la Cybersécurité

Fiche 14 : Firewall (Pare-Feu ou Garde-Barrière)

Compléments Couches Basses :

Fiche 15 : VLAN (Virtual LAN)

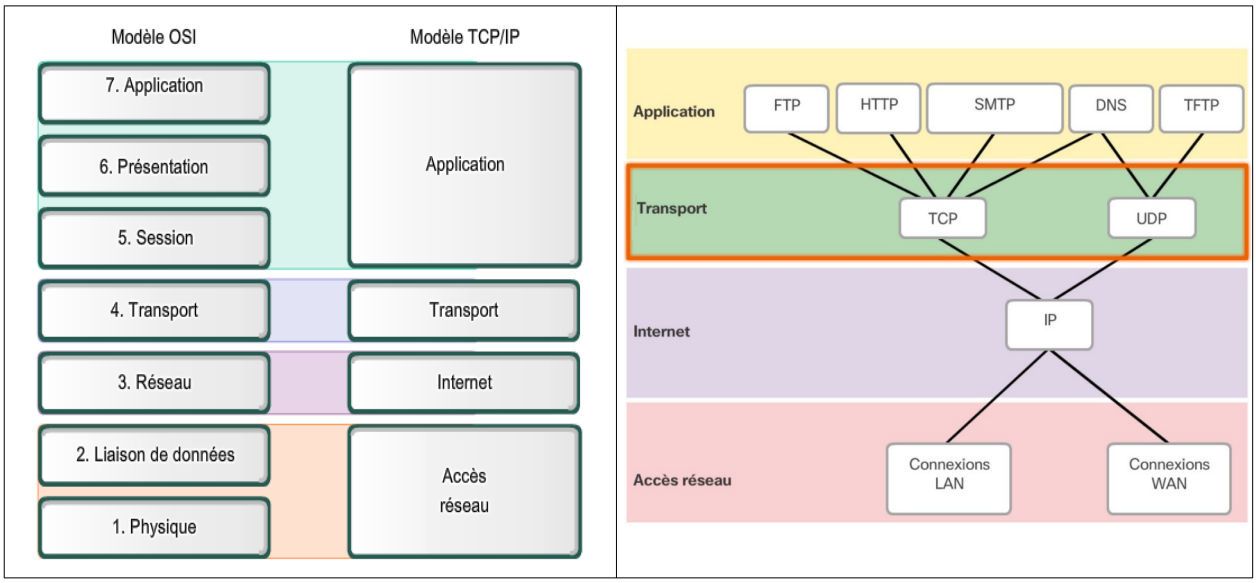
Fiche 16 : NAT (Network Adress Translation)

Fiche 17 : Adressage IPv6

10 : Couche 4 - Transport

I - Introduction

Le **rôle** de la couche « **Transport** » est de **segmenter** les données provenant des couches supérieures en ajoutant un **en-tête** pour suivre le flux des données et faciliter leur réorganisation. Elle se situe sur la **couche 4** du modèle **OSI** :



Les deux protocoles utilisés par cette couche sont le protocole **TCP** (**T**ransmission **C**ontrol **P**rotocol) et le protocole **UDP** (**U**ser **D**atagram **P**rotocol).

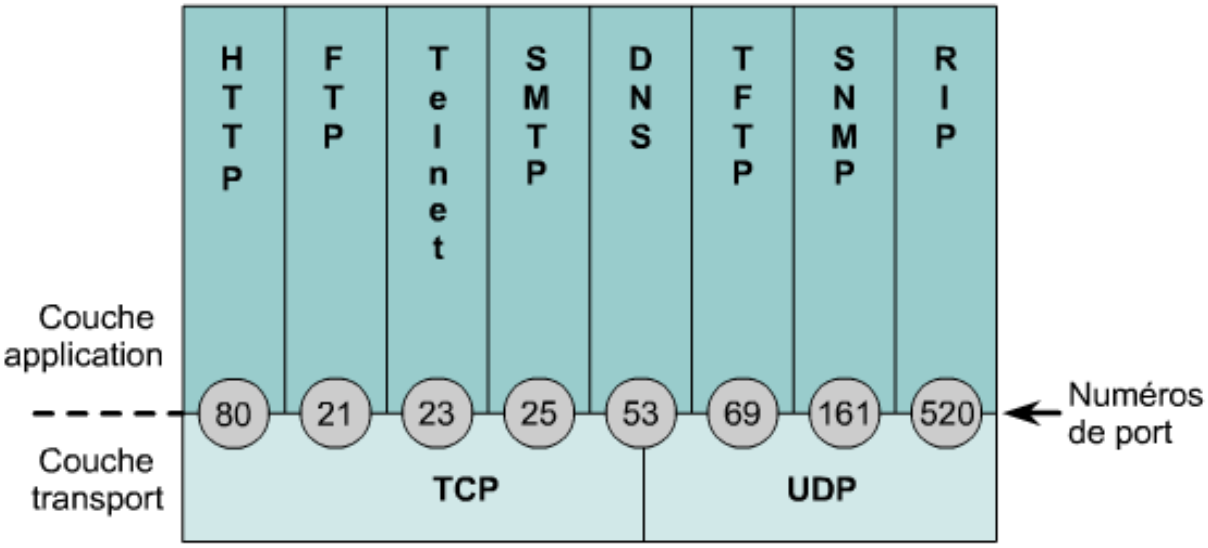
II - Comparaison entre UDP et TCP

UDP	TCP
- Protocole sans connexion ; - Aucune vérification logicielle de la livraison des messages ; - Pas de ré-assemblage des messages entrants ; - Pas d'accusé de réception ; - Aucun contrôle de flux.	- Protocole orienté connexion ; - Fiable ; - Division des messages sortants en segments ; - Ré-assemblage des messages au niveau du destinataire car chaque segment est numéroté à l'aide d'un **numéro d'ordre** ou **de séquence** ; - Ré-envoi de toute donnée non reçue.

III - Les ports utilisés

Pour **différencier les segments de chaque application**, les protocoles **TCP** (**T**ransmission **C**ontrol **P**rotocol) et **UDP** (**U**ser **D**atagram **P**rotocol) utilisent des identificateurs uniques appelés **ports**. La valeur du port est comprise ente **1** et **65535**.

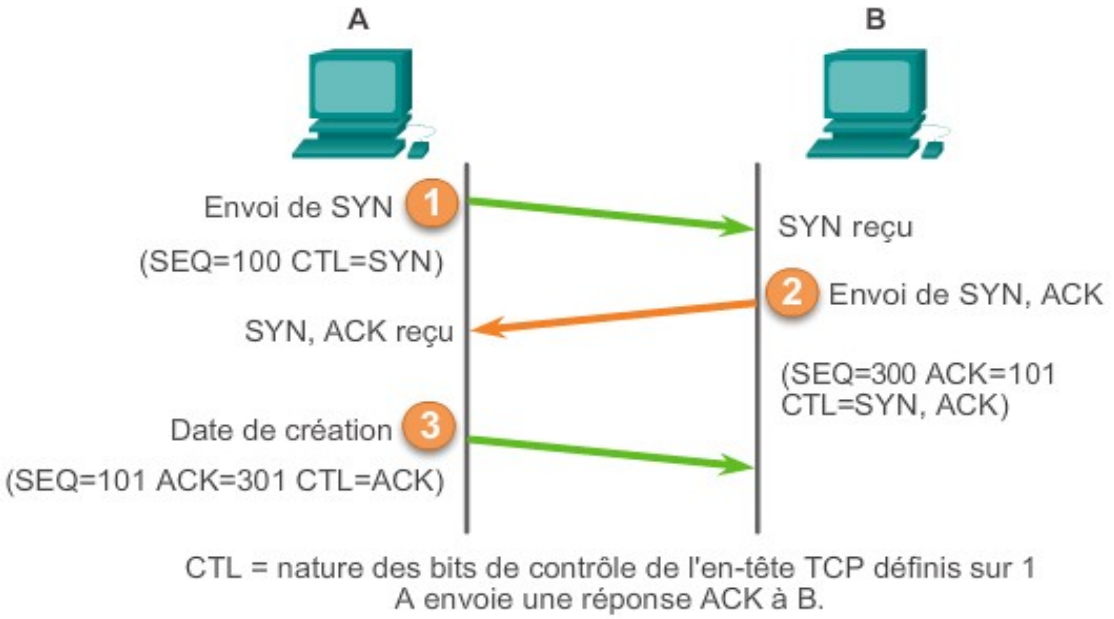
- Les numéros **inférieurs à 1024** sont considérés comme des numéros de **port reconnus** ;
- Les numéros **supérieurs à 1024** sont des numéros attribués de manière dynamique ;
- Les numéros de port enregistrés sont destinés à des applications spécifiques d'un fournisseur.



IV - Connexion TCP

Les hôtes **TCP** établissent une **connexion** en **3 étapes**, « **SYN – SYN/ACK - ACK** » :

- **SYN** : Requête du client envoyant l'indicateur de contrôle **SYN** (SYnchronize sequence Number) ;
- **SYN/ACK** : Réponse du serveur qui accuse la réception du segment **SYN** ;
- **ACK** : Le client répond par un accusé de réception.

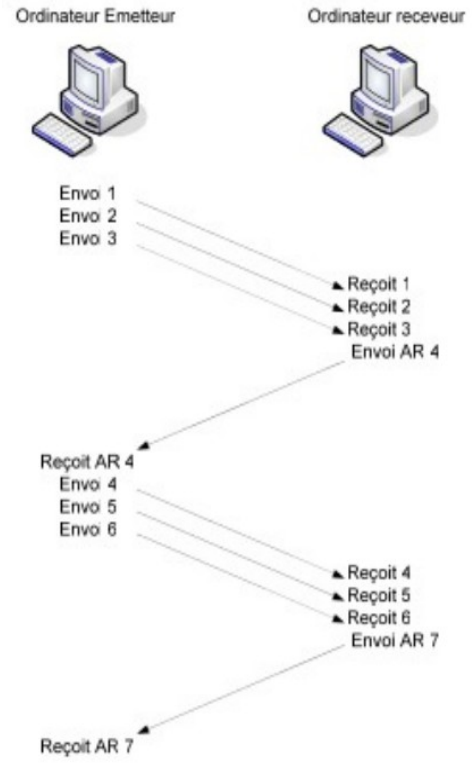


V - Fenêtrage TCP

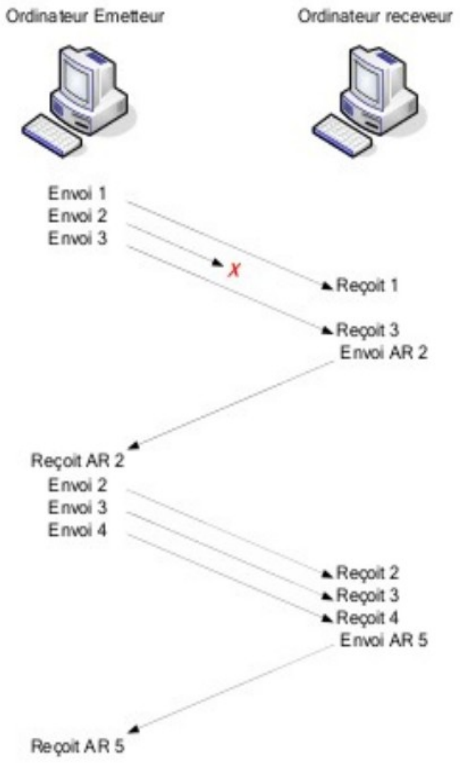
Le **Fenêtrage** est un mécanisme dans lequel le récepteur envoi un **accusé de réception** (**ACK**) après avoir reçu un certain nombre de données. Si le destinataire n'envoie pas d'accusé de réception, cela signifie pour l'émetteur que les informations ne sont pas parvenues correctement et dans ce cas elles sont retransmises.

La **taille de la fenêtre** détermine la quantité de données que l'on peut transmettre avant de recevoir un accusé de réception. **TCP** utilise un système d'accusé de réception prévisionnel, ce qui signifie que le **numéro** d'accusé renvoyé indique la prochaine séquence attendue.

Exemple d'échange pour une **taille de fenêtre = 3** :



Transmission sans perte de paquets

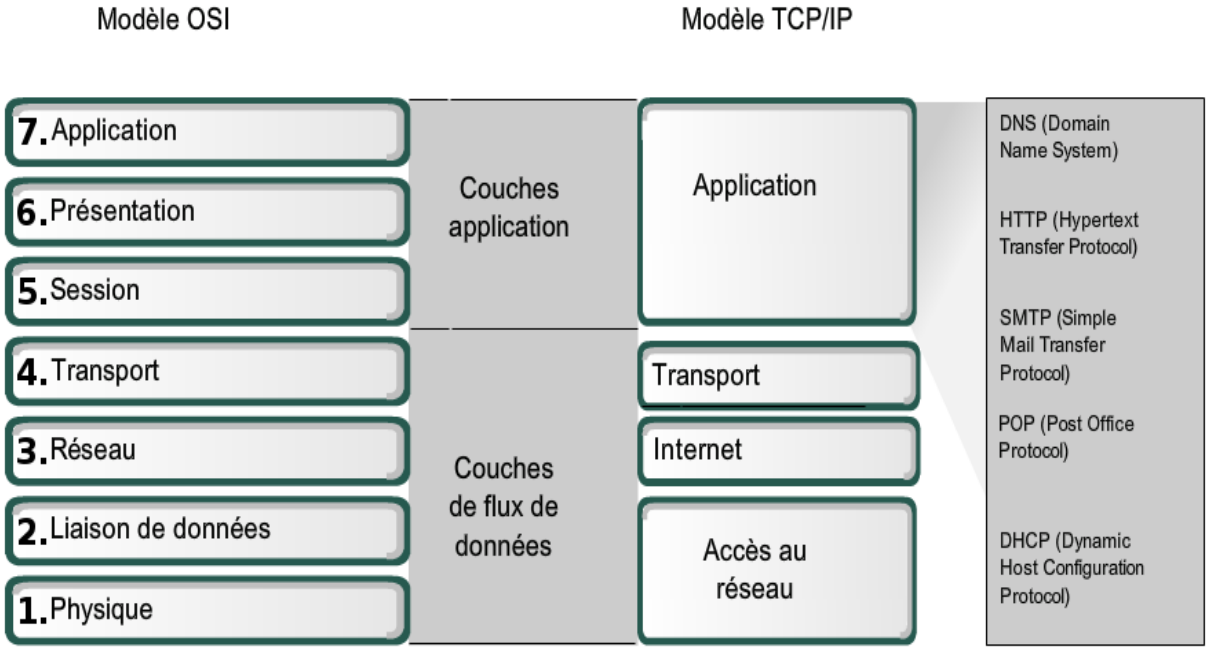


Transmission avec perte de paquets : ici le paquet 2 est renvoyé (le 3 aussi même s'il a été reçu).

11 : Couche 7 - Application

I - Introduction

Les couches **session**, **présentation** et **application** du **modèle OSI** sont regroupées dans la couche **application du modèle TCP/IP**. Cela signifie que la représentation, le code et le contrôle du dialogue sont traités au niveau de la couche **application TCP/IP**. Cette structure permet de garantir un maximum de flexibilité dans la couche application du modèle **TCP/IP** pour les développeurs d'applications.



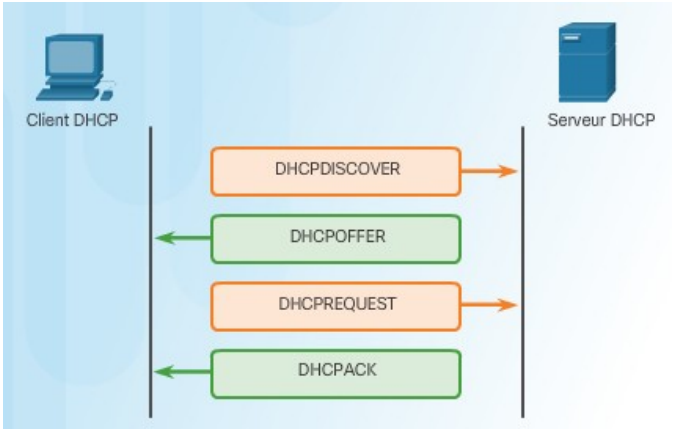
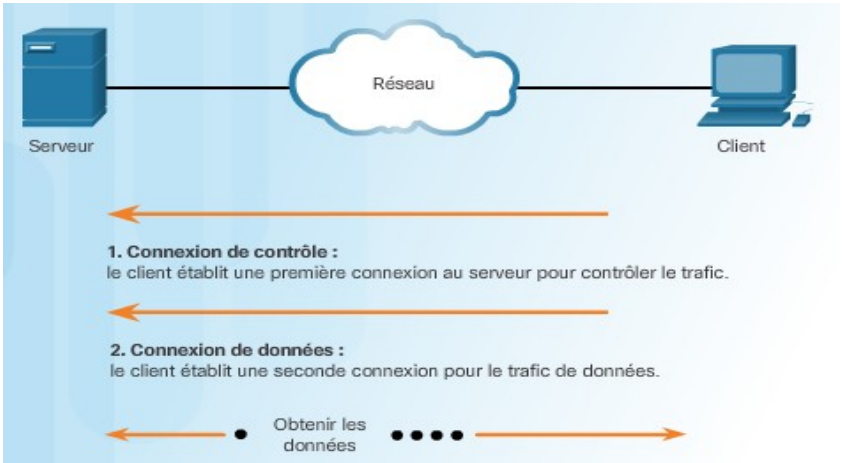
II - Modèle Client-Serveur

L'environnement **client-serveur** désigne un mode de **communication** à travers un réseau entre plusieurs programmes ou logiciels : l'un, qualifié de **client**, envoie des requêtes et l'autre qualifié de **serveur**, attend les requêtes des clients et y répond.

Toutes les **données sont centralisées sur le serveur**, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.

Téléchargement descendant	**download** en anglais : Les fichiers sont téléchargés du **serveur vers le client**.
Téléchargement montant	**upload** en anglais : Les fichiers sont téléchargés du **client vers le serveur**.
Architecture “peer-to-peer”	Le réseau est dit **pair à pair** (***peer-to-peer*** en anglais, ou ***P2P***), lorsque chaque ordinateur connecté au réseau est susceptible de jouer tour à tour le rôle de **client** et celui de **serveur**.

III - Les principaux protocoles de la couche « Application »

Protocole	Rôle
DHCP (**D**ynamic **H**ost **C**onfiguration **P**rotocol)	Le protocole **DHCP** permet à un hôte d'obtenir une configuration IP de manière dynamique sans que l'administrateur réseau ait à définir un profil pour chaque équipement. Le service **DHCP** utilise le protocole de transport **UDP** (**U**ser **D**atagram **P**rotocol). Le **serveur DHCP** utilise le **port 67** et le **client** le **port 68**. 
DNS (**D**omain **N**ame **S**ystem)	Afin de pouvoir créer un **lien** entre le contenu d'un **site** et son **adresse**, un système de **noms de domaine** a été établi. Le système de noms de domaine (**DNS**) est utilisé sur Internet pour convertir en **adresses IP** les **noms de domaine** et **leurs nœuds de réseau**. Le service **DNS** utilise le protocole de transport **TCP** (**T**ransmission **C**ontrol **P**rotocol) et **UDP** (**U**ser **D**atagram **P**rotocol). Le **serveur DNS** utilise le **port 53**.
FTP (**F**ile **T**ransfer **P**rotocol)	**FTP** est un service **orienté connexion** fiable qui utilise le protocole **TCP** pour **transférer des fichiers** entre des systèmes qui le prennent en charge. Le service **FTP** utilise le protocole de transport **TCP** (**T**ransmission **C**ontrol **P**rotocol). Le **serveur FTP** utilise le **port 20** pour la **transmission des données** et le **port 21** pour le canal de **connexion**. 

TFTP (**T**rivial **F**ile **T**ransfer **P**rotocol)	**TFTP** est un service **non orienté connexion** qui se sert du protocole **UDP** pour transférer des fichiers. Le service **TFTP** utilise le protocole de transport **UDP** (**U**ser **D**atagram **P**rotocol) et et le **serveur TFTP** utilise le **port 69**.
HTTP (**H**ypertext **T**ransfer **P**rotocol)	Le protocole **HTTP** est le support du Web. Dans l'URL **http://www.cisco.com/edu/**, la partie «**http://**» indique au navigateur le **protocole** à utiliser. La seconde partie, «**www**», indique le **nom de l'hôte** ou le nom d'un ordinateur précis doté d'une adresse IP spécifique. Enfin, le suffixe « **/edu/** » précise l'emplacement exact du **dossier** sur le serveur qui contient la page Web. Le service **HTTP** utilise le protocole de transport **TCP** et le **serveur HTTP** utilise le **port 80**.
SMTP (**S**imple **M**ail **T**ransfer **P**rotocol)	Les **serveurs de messagerie** communiquent entre eux à l'aide du protocole **SMTP** pour envoyer et recevoir des messages électroniques. Les protocoles de client de messagerie les plus répandus sont **POP3** et **IMAP4**, qui utilisent tous deux TCP pour transporter les données. Le service **SMTP** utilise le protocole de transport **TCP** et le **serveur SMTP** utilise le **port 25**.
SNMP (**S**imple **N**etwork **M**anagement **P**rotocol)	Le protocole **SNMP** est un protocole qui facilite l'échange d'informations de gestion entre les équipements du réseau. Il permet aux administrateurs réseau de gérer les performances du réseau, de diagnostiquer et de résoudre les problèmes, ainsi que d'anticiper la croissance du réseau. **SNMP** utilise le protocole de transport **UDP**. Le port **161** est utilisé par l'agent pour recevoir les requêtes du serveur. Le port **162** est réservé au serveur pour recevoir les alertes des agents.
SSH (**S**ecure **Sh**ell)	Le protocole **SSH** a été conçu avec l'objectif de remplacer les différents programmes **rlogin**, **telnet**, **rcp**, **ftp** et **rsh**. **SSH** est un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Le service **ssh** utilise le protocole de transport **TCP** (**T**ransmission **C**ontrol **P**rotocol) et le **serveur ssh** utilise le **port 22**.

12 : Introduction à la Cybersécurité

I - Introduction

La **cybersécurité** est la **pratique** consistant à **protéger** les systèmes, les réseaux et les programmes **contre les attaques numériques**.

II - La nécessité de la Cybersécurité

La **Cybersécurité** s'appuie sur le modèle **CID** (**Confidentialité**, **Intégrité** et **Disponibilité**) :

Confidentialité : Protéger l'accès aux données sensibles contre les personnes non autorisées. Elle garantit l'anonymat des données en limitant l'accès par le **chiffrement** (**cryptographie**) ;

Intégrité : Assurer que les données ne sont ni altérées ni corrompues. Elle garantit que l'information est exacte et fiable ;

Disponibilité : Garantir que les personnes autorisées peuvent accéder à l'information.

Remarque : Au-delà des 3 critères **DIC** (**disponibilité**, **intégrité** et **confidentialité**), un quatrième critère est parfois ajouté : la **traçabilité** (aussi appelés critères **DICT**). Si des données ont été modifiées ou supprimées, il peut être important de pouvoir identifier d'où provient cette modification.

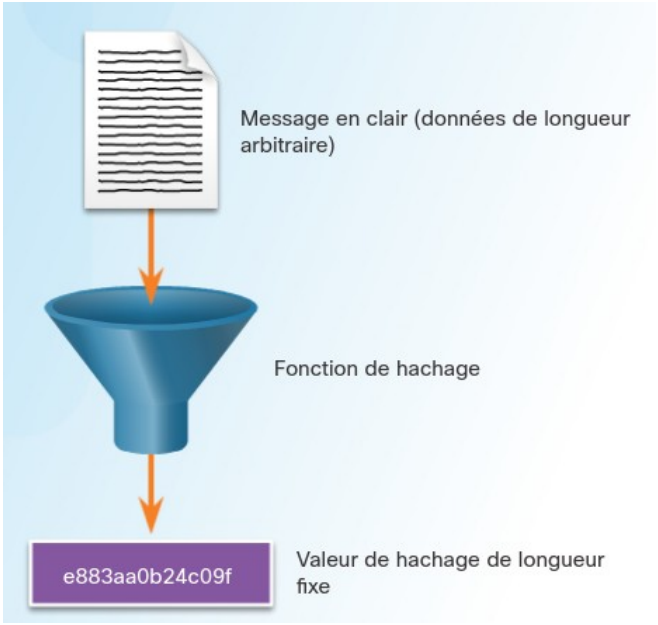
III - La confidentialité des données

La **confidentialité des données** est réalisée à l'aide de la **cryptographie** qui consiste à chiffrer des données pour les rendre **confidentielles**. Voir **Fiche 13**.

IV - L'intégrité des données

Une **somme de contrôle** est utilisée pour vérifier l'intégrité des fichiers ou des chaînes de caractères après leur transfert d'un périphérique à un autre dans votre réseau local ou sur Internet.
Les **sommes de contrôle** sont calculées grâce à des fonctions de **hachage**. Parmi les sommes de contrôle les plus courantes, il y a **MD5**, **SHA-1**, **SHA-256** et **SHA-512**.

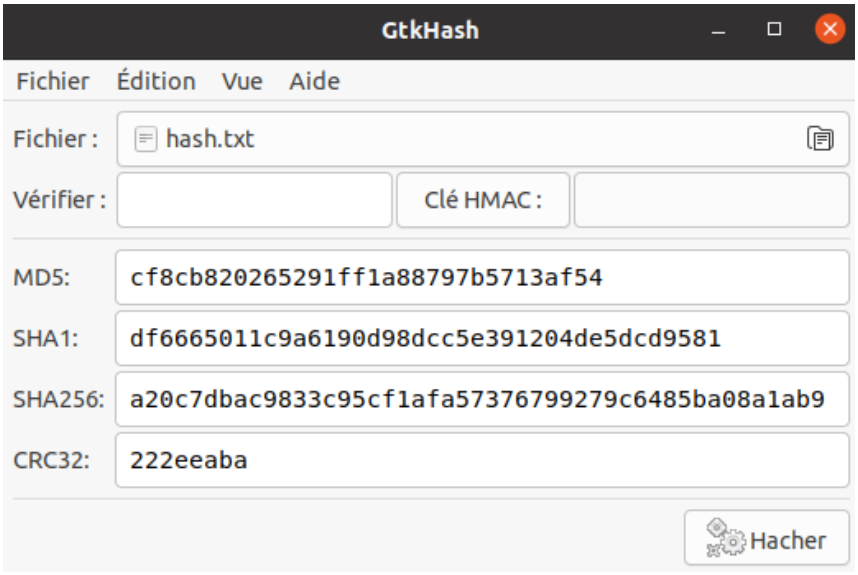
Comme le montre la figure ci-dessous, une fonction de **hash** utilise un algorithme mathématique pour transformer les données en une **valeur de longueur fixe** qui représente les données. La valeur hachée est simplement présente pour comparaison.



Il est impossible d'extraire directement les données d'origine à partir de la valeur hachée. Par exemple, si vous avez oublié votre mot de passe, vous ne pourrez pas le récupérer à partir de la valeur hachée. Il vous faut réinitialiser le mot de passe.

Remarque : Afin d'améliorer l'efficacité du hachage, on peut ajouter un **sel** (**salt**). c'est une donnée informatique qui n'est pas secrète, souvent aléatoire, qui peut être ajoutée au message avant hachage.

On peut générer des **sommes de contrôle** avec le logiciel **HashCalc** sous Windows ou **GtkHash** sous **Ubuntu** :



V - La disponibilité des données

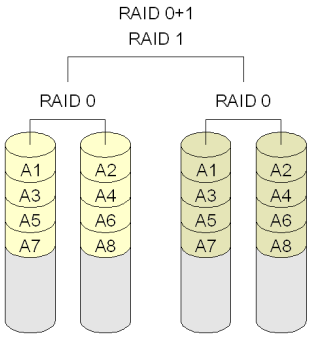
La **disponibilité** garantit que les utilisateurs autorisés d'un système ont un accès rapide et ininterrompu aux informations contenues dans ce système, ainsi qu'au réseau. Voici les **méthodes** qui permettent cette disponibilité :

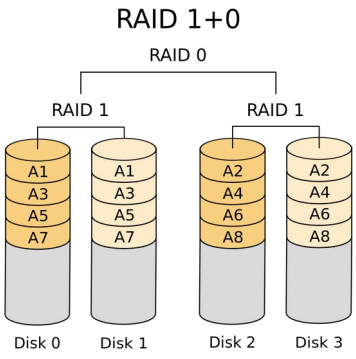
- **Répartition équitable** : Communément appelée **équilibrage des charges**, la répartition équitable permet de distribuer la charge (demandes de fichiers, acheminement des données, etc.) de manière à ce qu'aucun appareil ne soit trop sollicité.
- **Haute disponibilité** : La haute disponibilité se rapporte aux mesures utilisées pour maintenir opérationnels les services et les systèmes d'information pendant une panne. L'objectif de la haute disponibilité est souvent que les services clés soient disponibles 99,999 % du temps (disponibilité « cinq neufs »). Les stratégies de haute disponibilité comprennent la **redondance** et le **basculement**.
- **Redondance** : La redondance fait référence aux systèmes qui sont soit dupliqués, soit basculés vers d'autres systèmes en cas de dysfonctionnement. On appelle « **basculement** » le processus de reconstruction d'un système ou de passage à d'autres systèmes lorsqu'une défaillance est détectée.

Dans le cas d'un **serveur**, lorsqu'une défaillance est détectée, le serveur bascule vers un serveur redondant. Cette stratégie permet de maintenir la continuité du service jusqu'à ce que le serveur principal soit restauré. Si, l'environnement exige un niveau de disponibilité élevé, les serveurs doivent être regroupés en clusters.

La **redondance** et le **basculement** s'appuient sur :

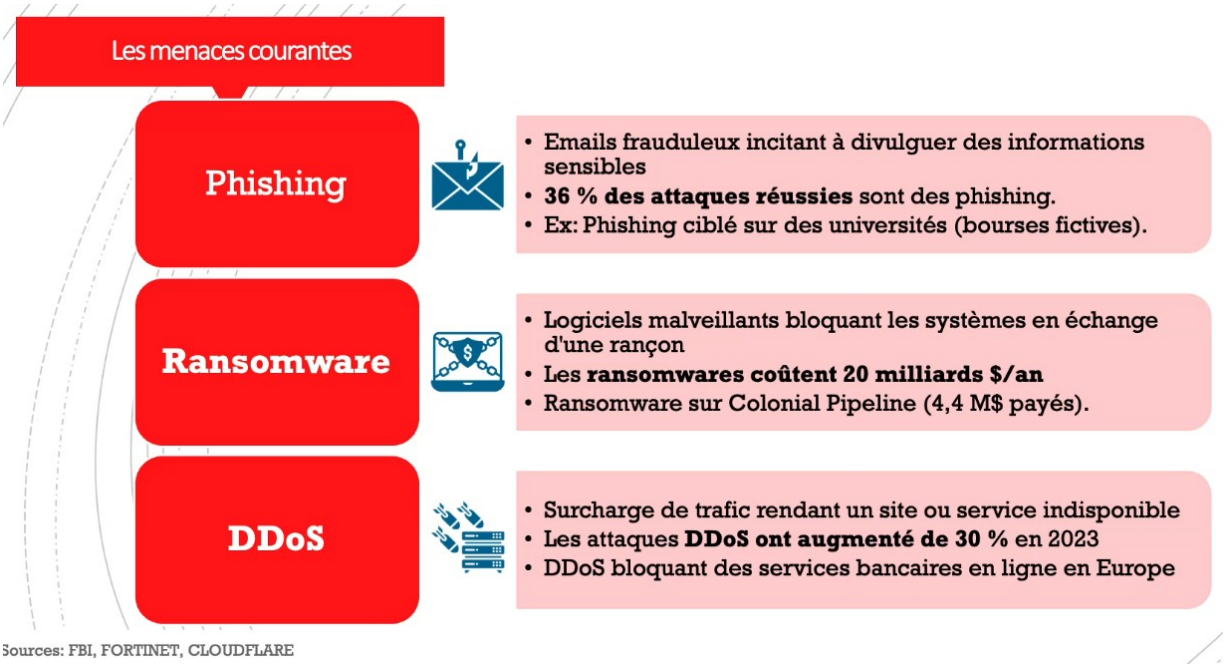
- **La Tolérance aux pannes**. La tolérance aux pannes est la capacité d'un système à rester opérationnel en cas de défaillance d'un composant. Les systèmes tolérants aux pannes peuvent continuer à fonctionner même si un composant critique, par exemple un lecteur de disque, tombe en panne.
- **RAID (Redundant Array of Independent Disks)**. La technologie **RAID** utilise plusieurs disques pour assurer la tolérance aux pannes :
  - **RAID 0** (entrelacement de disques – striping) : Répartition des données sur plusieurs disques sans tolérance aux pannes,
  - **RAID 1** (disques en miroir) : Les données sont dupliquées,
  - **RAID 3 ou 4** (entrelacement de disques avec parité dédiée) : Un disque supplémentaire est nécessaire pour stocker les contrôles de parité,
  - **RAID 5** (entrelacement de disques avec parité distribuée) : Reconstruction des fichiers en cas de panne d'un des disques. Technologie qui nécessite l'emploi d'au moins cinq disques,
  - **RAID 6** (entrelacement de disques avec double parité) : Evolution du **RAID 5**.





VI - Hygiène numérique essentielle

L'**hygiène numérique** consiste à avoir un **comportement** permettant de **maintenir les données sensibles en sécurité** et de les **protéger contre les cyberattaques et le vol**.



Le **comportement à respecter** à minima est :

- d'avoir des **mots de passe robustes** : Utiliser des mots de passe complexes et uniques pour chaque compte ;
- de faire des **mises à jour régulières** : Maintenir logiciels et systèmes à jour pour corriger les failles de sécurité ;
- d'être **vigilant en ligne** : Être attentif aux courriels suspects et liens douteux.

La **CNIL** (**C**ommission **N**ationale de l'**I**nformatique et des **L**ibertés) recommande d'utiliser des mots de passe comportant au moins **12 caractères** dont **1 minuscule**, **1 majuscule**, **1 chiffre** et **1 caractère spécial** (par exemple **A4c3b1d4f6e5#**).

Il existe différentes méthodes pour générer des mots de passe, une méthode simple est la suivante :

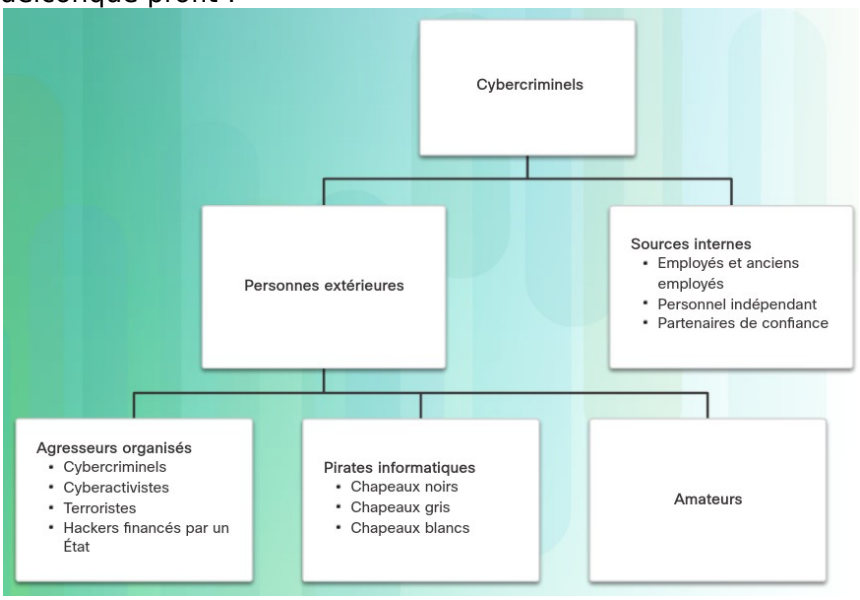
1. Penser à une phrase (paroles de chanson, réplique de film ou comptine de votre enfance) ;
2. Prendre la première lettre des 5 premiers mots ;
3. Ajouter un symbole entre chaque lettre ;
4. Rajouter une couleur à la fin ;
5. Exemple :
6. « Au clair de la lune mon ami Pierrot » => **A#C#D#L#L#Bleu**

Remarques :

- Il est possible **tester la rapidité de découverte d'un mot de passe** avec des outils en ligne comme **Bitwarden Password Strength Tester** (https://bitwarden.com/password-strength/) et **Kaspersky Password Checker** (https://password.kaspersky.com/).
- Pour éviter de devoir mémoriser un grand nombre de mots de passe, il est possible d'utiliser un **gestionnaire de mots de passe** recommandé par la **CNIL**. Par exemple, **Keepass**.

VII - Cybercriminels et programmes malveillants (malware)

Les **Cybercriminels** sont des hommes qui commettent un délit à l'aide d'outils informatiques, notamment en piratant des données existantes sur Internet afin d'obtenir illégalement de l'argent ou un quelconque profit :



Les malware ou **programmes malveillants**, représentent tout code pouvant être utilisé pour voler des données, contourner les contrôles d'accès ou pour nuire à un système ou le compromettre. Voici quelques types communs de malware :

Malwares	**Description**
Bot	Du mot robot, un bot est un programme malveillant conçu pour exécuter automatiquement une action, généralement en ligne.
Rançongiciel ou ransomware	Programme malveillant conçu pour tenir en otage un système informatique ou les données qu'il contient jusqu'à ce qu'un paiement soit effectué
Rootkit	Logiciel conçu pour modifier le système d'exploitation afin de créer une porte dérobée.
Logiciel Espion	Souvent fourni avec des programmes légitimes, ce programme malveillant est conçu pour suivre l'activité d'un utilisateur.
Virus	Code exécutable malveillant joint à d'autres fichiers exécutables, qui sont souvent des programmes légitimes.
Cheval de troie	Programme malveillant qui effectue des opérations nuisibles sous couvert d'une opération souhaitée.
Publiciel	Parfois fourni avec d'autre logiciels, il est conçu pour diffuser automatiquement des publicités.
MitM (Man in the Middle – l'homme au milieu)	Un attaquant intercepte les données afin de prendre le contrôle d'un appareil (MitMo : cas d'un appareil mobile).
Scareware	Logiciel conçu pour convaincre l'utilisateur d'effectuer une action spécifique en lui faisant peur. Le scareware crée des fenêtres contextuelles factices avec la même apparence que les fenêtres de dialogue du système d'exploitation.
Vers	Code malveillant qui se reproduit en exploitant indépendamment des vulnérabilités dans les réseaux.

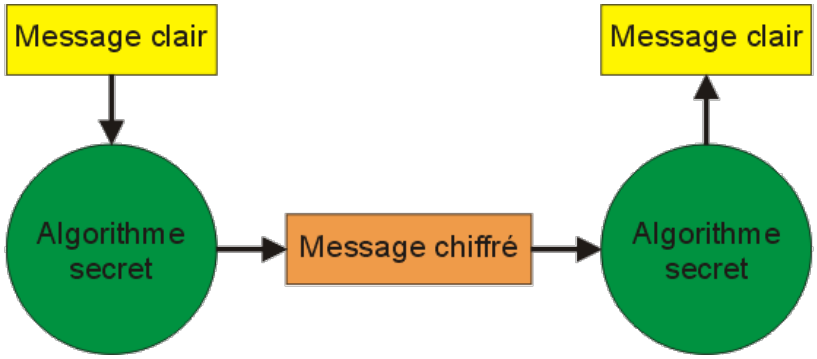
13 : Cryptographie pour la Cybersécurité

I - Introduction

La **cryptographie** consiste à chiffrer des données pour les rendre **confidentielles**. C'est la base de tout échange d'informations **sécurisé**.

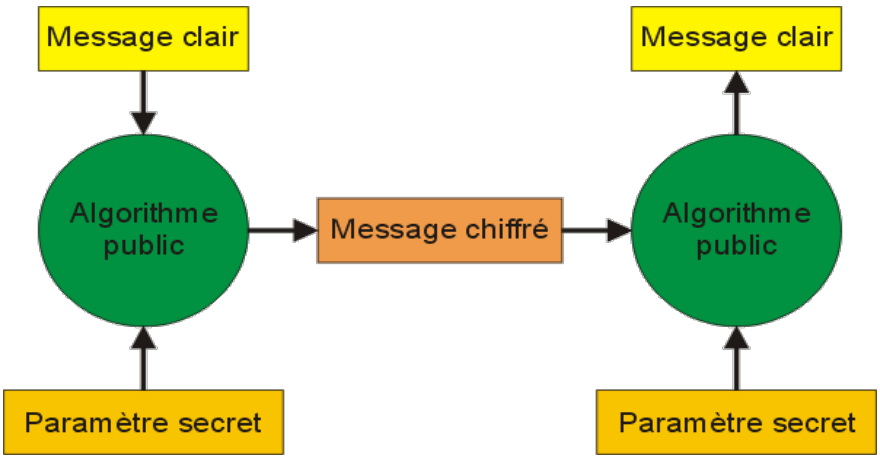
II - Sécurisation d'un échange d'informations avec un algorithme secret

Ici on dispose d'un **algorithme de chiffrement secret** qui assure à lui seul la confidentialité du message. Un tel procédé, cependant, n'est pas considéré comme sûr, si quelqu'un réussis à reconstituer l'algorithme alors il n'y aura plus de secret.



III - Sécurisation d'un échange d'informations avec un algorithme et une clé

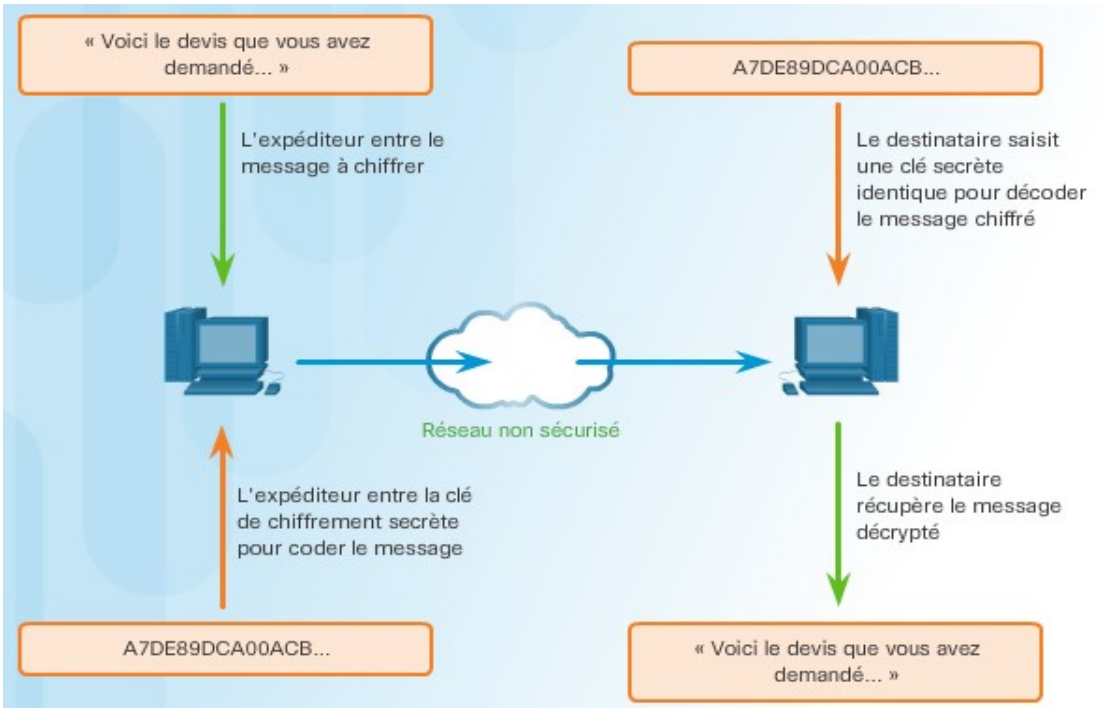
Ici on utilise un **algorithme de chiffrement public**, que tout le monde peut analyser et utiliser, mais qui exploitera un paramètre de chiffrement (**clé de chiffrement**) qui, lui ne sera pas partagé.



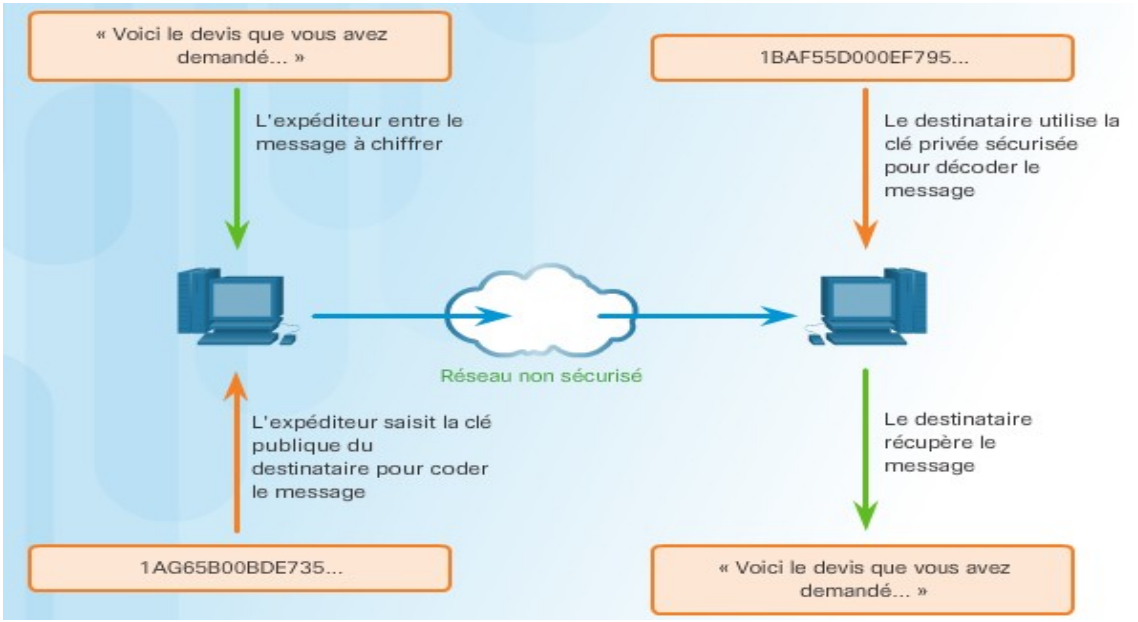
Ce principe, qui peut éventuellement adopter des **combinaisons de clés**, reste à l'heure actuelle le procédé le plus sûr. Ici, pour déchiffrer le message, il faudra la bonne clé, l'algorithme étant public.

IV - Chiffrement Symétrique et Asymétrique

Chiffrement Symétrique : On utilise **une clé** de chiffrement et de déchiffrement identiques. Inconvénient : Comment transmettre cette clé de chiffrement en toute sécurité ?



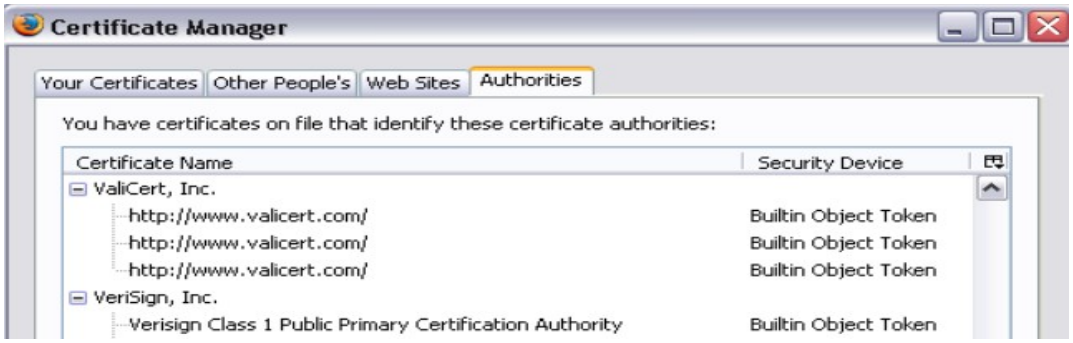
Chiffrement Asymétrique : On utilise 2 clés, une **clé publique** qui peut être partagée et une **clé privée**. Ce qui est chiffré avec une clé publique ne peut être déchiffré qu'avec la clé privée correspondante. Ce qui est chiffré avec la clé privée ne peut être déchiffré qu'avec la clé publique correspondante.



Remarque : La **clé publique** est unique, il suffit de la récupérer via un **certificat numérique signé** par un dépositaire, dit « **tiers de confiance** » ou **CA** : **C**ertificate **A**uthority.

V - Les certificats numériques

Lorsqu'on s'adresse à un organisme de confiance (**CA**) pour récupérer une clé publique, il l'envoi avec un **certificat numérique signé**. Les certificats sont à la norme **X.509**. Par exemple un navigateur installe des certificats de divers organismes qui contiennent une clé publique :



VI - Protocole Telnet

Telnet permet d'ouvrir un shell sur un hôte distant. Il utilise le protocole de transport **TCP** et le port **23**. Avec **Telnet** la totalité de la transaction passe en **clair** sur le réseau. Ce protocole n'est plus utilisé en raison de son manque de sécurité.

VII - Protocole SSH

SSH (Secure Shell) permet d'ouvrir un shell sur un hôte distant. Il utilise le protocole de transport **TCP** et le port **22**. Avec **SSH** la totalité de la transaction entre un client et le serveur est cryptée grâce au **chiffrement symétrique** et **asymétrique**.

Le dialogue **SSH** peut se résumer par les étapes suivantes :

- Le serveur envoie sa **clé publique** au client ;
- Le client génère une **clé secrète** (**privée**) et l'envoie au serveur, en cryptant l'échange avec la **clé publique** du serveur (**chiffrement asymétrique**) ;
- Le client et le serveur peuvent alors établir un **canal sécurisé** grâce à la **clé secrète** commune (**chiffrement symétrique**).

SSH possède deux mécanismes différents d'authentification :

- **Authentification par mot de passe** : Couple Nom de compte / Mot de passe ;
- **Authentification par clés** : Couple de **clés privée/publique**.

Utilisé généralement avec un mécanisme d'**authentification par mot de passe**. Lors de la première connexion du client au serveur, le serveur propose d'envoyer **la clé publique** au client :

```
jcabianca@jcabianca-HP-PC:~$ ssh etudiant@192.168.43.45
The authenticity of host '192.168.43.45 (192.168.43.45)' can't be established.
ECDSA key fingerprint is SHA256:Cf2h/nVfzceNWJnxtFh2iDIMPYmHNHNaac0aTMnBiRk.
Are you sure you want to continue connecting (yes/no)?
```

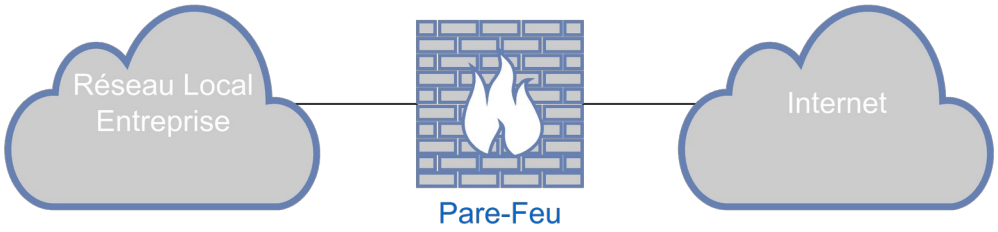
On accepte en saisissant **yes**. La **la clé de chiffrement** est maintenant sauvegardée sur le client.

14 : Firewall (Pare-Feu ou Garde-Barrière)

I - Introduction

Un **FireWall** correspond à un ensemble de moyens **matériels** et **logiciels** mettant en œuvre des fonctionnalités de **sécurité** sur un réseau local afin de gérer les accès avec les autres réseaux interconnectés.

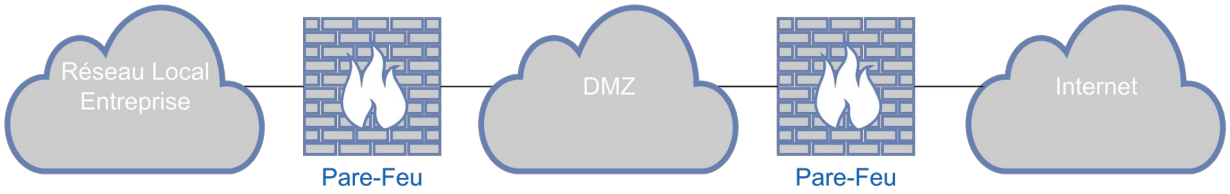
II - Architecture simple



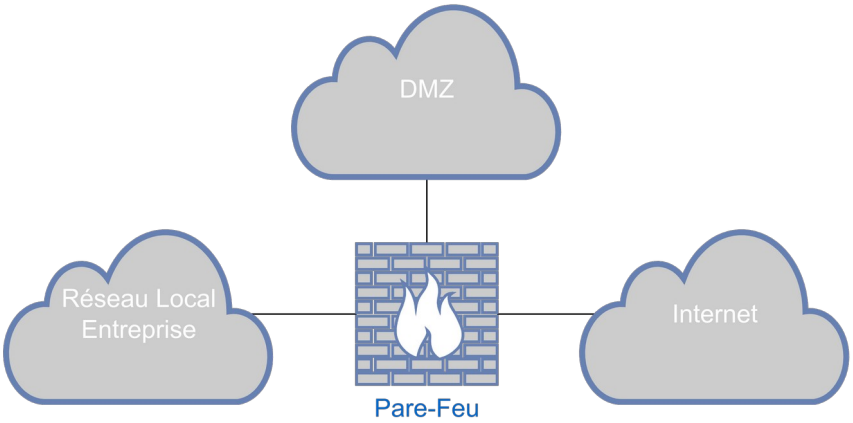
III - Architecture avec DMZ

Une **DMZ** (**De**M**ilitarized** **Z**one ou Zone démilitarisée) est une zone qui n'est ni publique, ni interne. On placera sur la **DMZ** tous les serveurs qui ont besoins d'être accessibles depuis l'extérieur.

Architecture avec 2 routeurs :



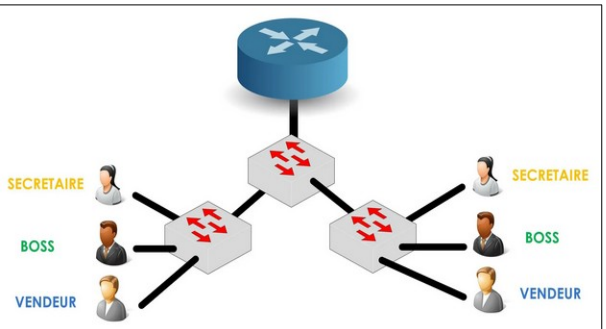
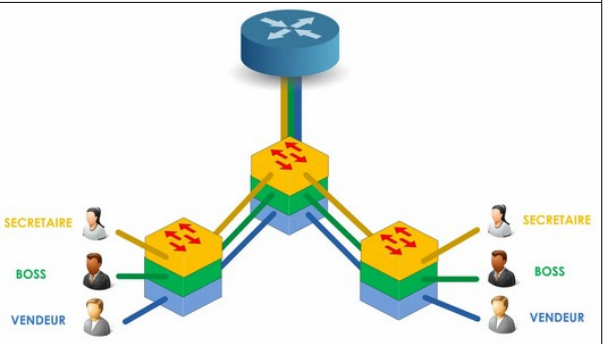
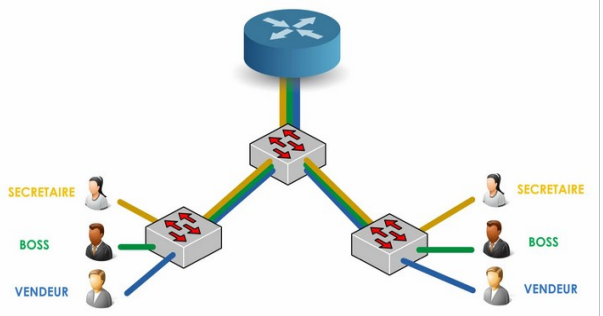
Architecture avec un seul routeur à 3 interfaces :



15 : VLAN (Virtual LAN)

I - Introduction

Grâce à la technologie des **réseaux locaux virtuels** (**VLAN**), les architectes réseau peuvent **segmenter** les périphériques physiques en sous-groupes logiques et bénéficier d'avantages en matière de performance et de sécurité.

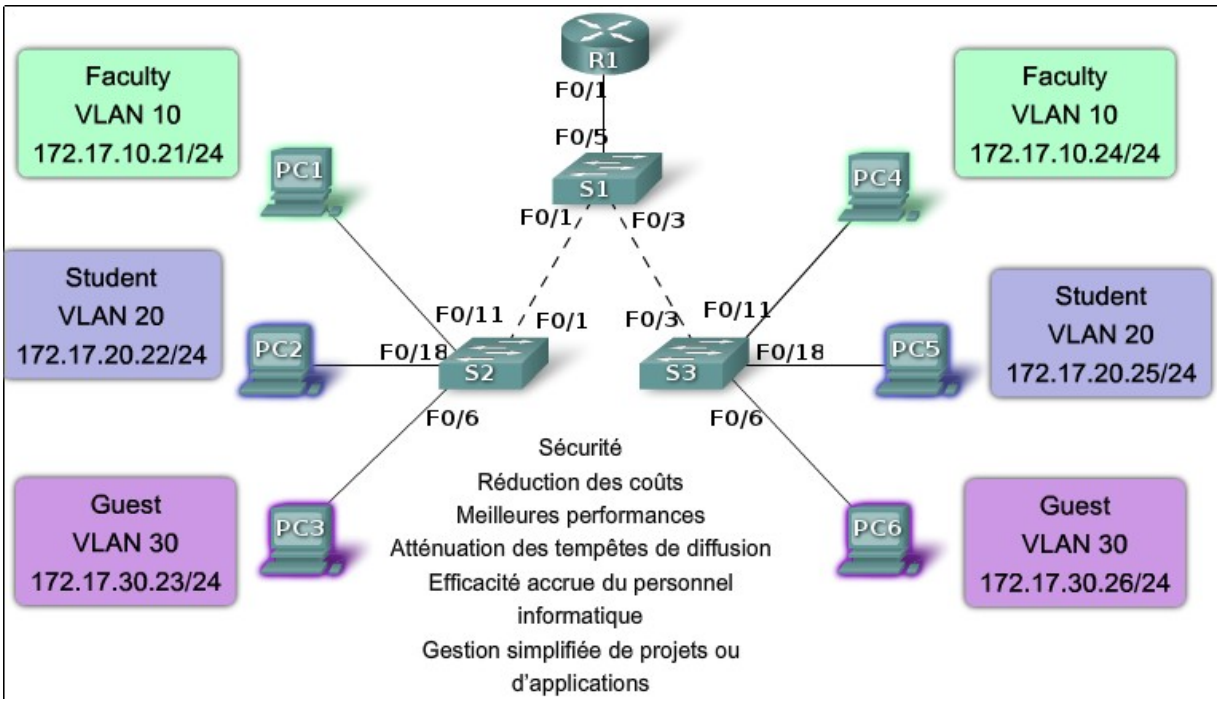
Sans VLAN : Avec une architecture comportant 3 groupes de travail, si un ordinateur d'un groupe émet un broadcast (ARP par exemple), tous les ordinateurs du réseau vont être concernés. **Comment limiter les broadcasts ?**	
Solution 1 : Créer un réseau physique par groupe de travail. Inconvénient : Il faut revoir le câblage et multiplier le matériel actif.	
Solution 2 : Les VLAN (Virtual LAN) Les **VLAN** (***V**irtual **LAN***) ont été mis au point afin de définir des **domaines de diffusion de couche 2 virtuels** en regroupant des stations de travail **indépendantes** du point de vue **géographique** sans intervenir sur le **câblage**. Les réseaux « **logiques** » ainsi constitués auront les mêmes caractéristiques que les **réseaux physiques**.	

Remarque : Il sera nécessaire de faire du **routage** inter-Vlan si l'on veut faire communiquer les différents **VLAN**. Le **lien** physique **véhiculant plusieurs VLAN** est appelé une **agrégation** ou un **trunk**.

On distingue **3 niveaux** de **VLAN** :

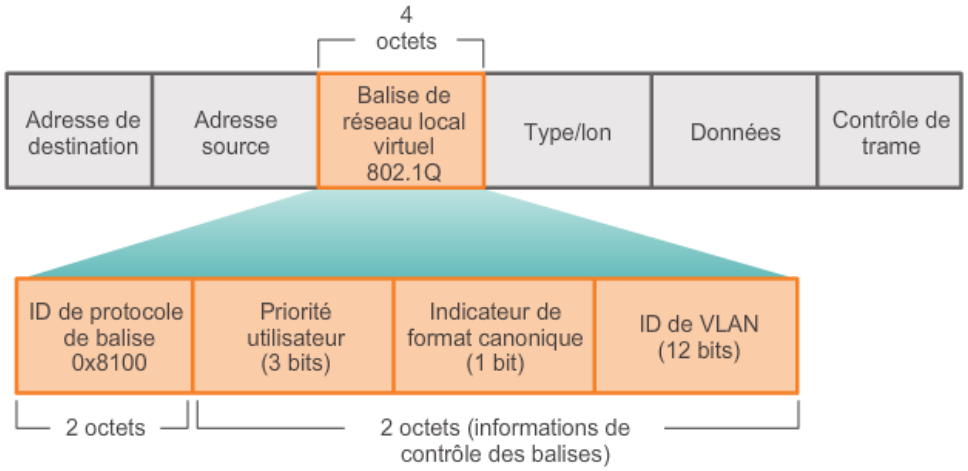
- **VLAN** de **niveau 1** ou de niveau **physique** : **VLAN par ports** qui consiste à associer à chaque port d'un **commutateur** un **VLAN** ;
- **VLAN** de **niveau 2** ou de niveau **liaison** : **groupe d'adresses MAC** ;
- **VLAN** de **niveau 3** ou de niveau **paquet** : **groupe de sous-réseaux**.

II - Exemple de VLAN par ports



III - Reconnaissance des trames

La **norme 802.1Q** (*Virtual Bridge Local Area Network ou VBLAN*) de l'IEEE qui est le standard pour les **LAN commuté** comme **Ethernet**, permet de « **signer** » un **VLAN** par une **étiquette** ou un **tag** inséré dans la **trame Ethernet** (ajout de 4 octets dans la trame) :



L'en-tête de la trame **Ethernet** est modifié, il comporte **4 octets** supplémentaires :

- **Tag Protocol Identifier (TPID) – 2 octets :** Dans le cas de la balise **802.1Q**. La valeur de ce champ est fixée à **0x8100**.
- **Priorité Utilisateur – 3 bits :** Ce champ de **3 bits** permet de coder 8 niveaux de priorités de 0 à 7.
- **Control Format – 1 bit :** Un commutateur Ethernet fixera toujours cette valeur à 0.
- **VLAN Identification (VID) – 12 bits :** Ce champ de **12 bits** sert à identifier le réseau local virtuel auquel appartient la trame. Il est possible de coder **4094** (2^{12}-2) VLANs.

16 : NAT (Network Address Translation)

I - Introduction

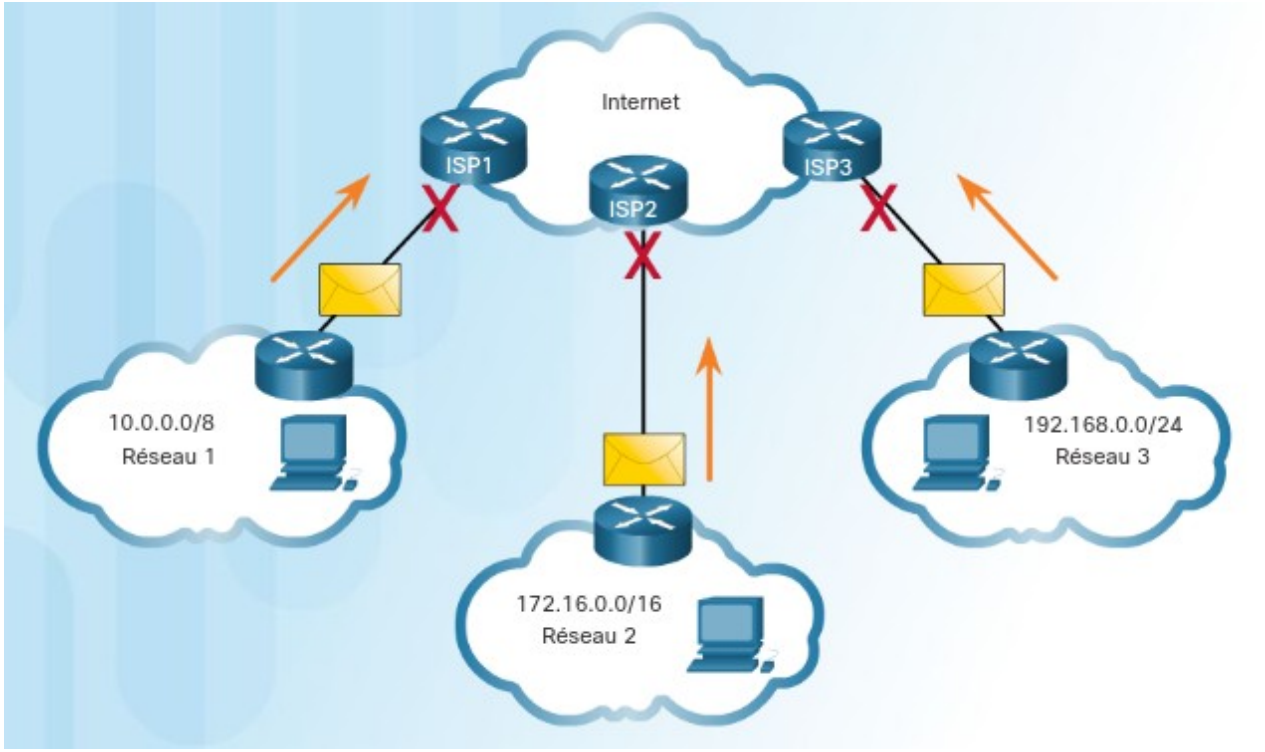
La plupart des entreprises utilisent des **adresses IPv4 privées** pour leurs hôtes internes.

Les **adresses IPv4 privées** ont été créées au milieu des années 1990 en raison de la pénurie d'espace d'adresses IPv4. Les adresses IPv4 privées ne sont pas uniques et peuvent être utilisées par un réseau interne.

Les blocs d'adresses **privées** sont les suivants :

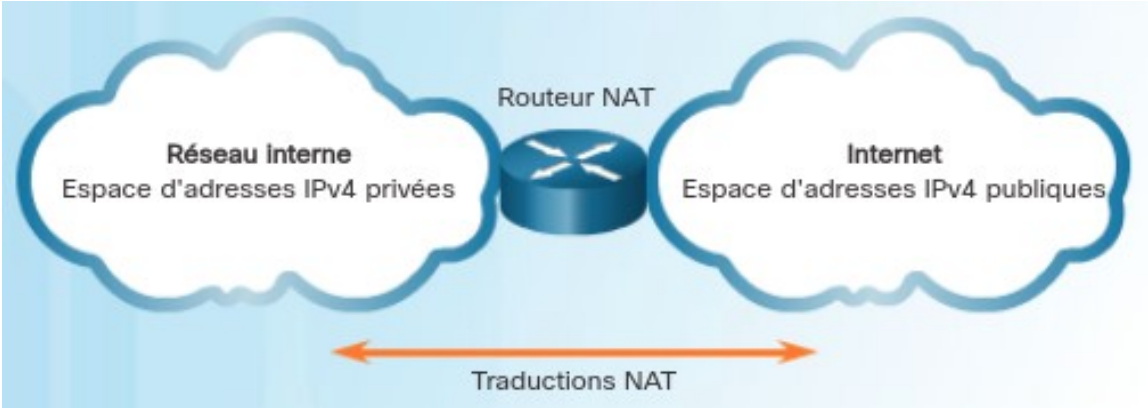
- **10.0.0.0 /8** ou **10.0.0.0** à **10.255.255.255**
- **172.16.0.0 /12** ou **172.16.0.0 à 172.31.255.255**
- **192.168.0.0 /16** ou **192.168.0.0 à 192.168.255.255**

Il est important de savoir que les adresses appartenant à ces blocs ne sont pas **routables**. Cela veut dire qu'elles ne sont pas autorisées sur Internet et doivent être filtrées (rejetées) par les routeurs Internet.



Comme ces adresses ne sont pas **routables** via Internet, elles doivent être traduites en **adresses IPv4 publiques**. Cette opération se nomme la **traduction d'adresses** réseau (**NAT**).

Dans la terminologie **NAT**, le réseau **interne** désigne l'ensemble des réseaux soumis à la traduction et le réseau **externe** désigne tous les autres réseaux :

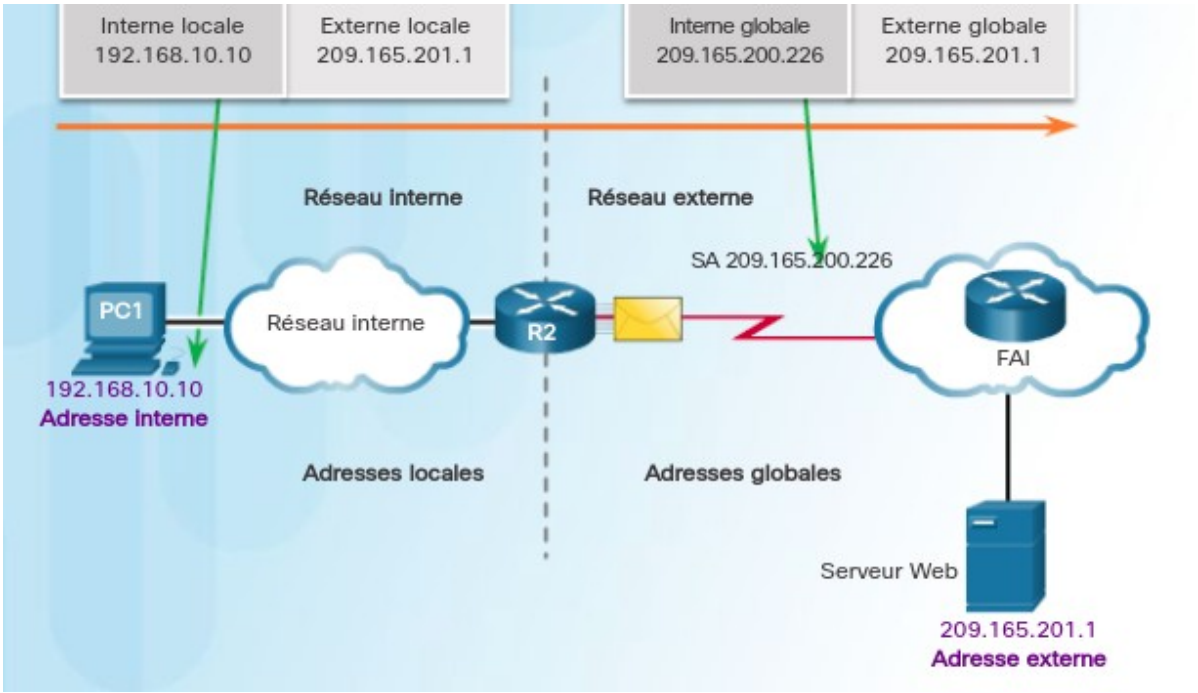


L'**adresse interne** correspond à l'adresse du périphérique traduite via la **NAT** et l'**adresse externe** correspond à l'adresse du **périphérique de destination**.

Généralement, cette opération s'effectue sur le routeur qui connecte le réseau interne à celui du **FAI** et une adresse **IPv4 publique** est attribuée à l'interface du routeur qui se connecte au réseau du **FAI**.

La fonction **NAT** comprend quatre types d'adresses :

- **Adresse locale interne** ;
- **Adresse globale interne** ;
- **Adresse locale externe** ;
- **Adresse globale externe**.



II - NAT Statique

Le **NAT statique** utilise un mappage de type **un à un** des adresses locales et globales. Ces mappages sont configurés par l'administrateur réseau et restent constants.

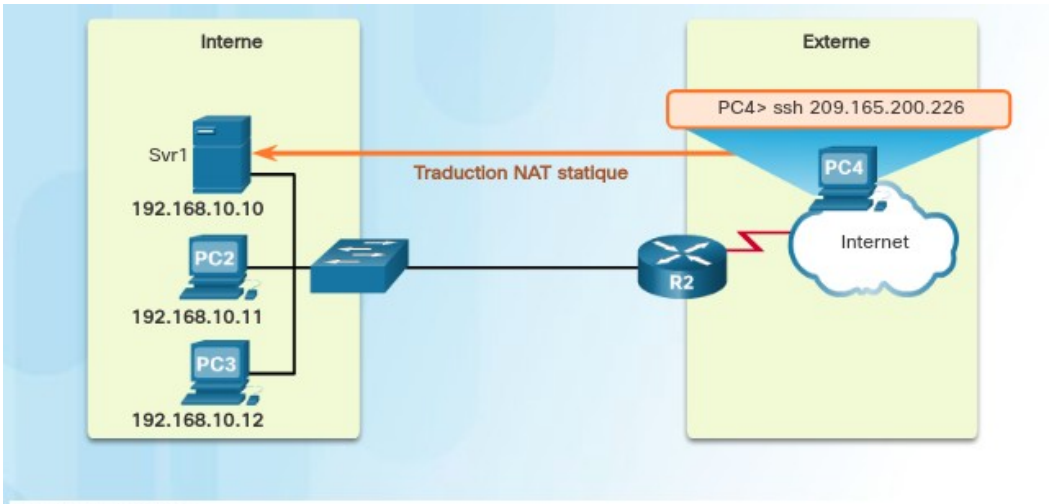
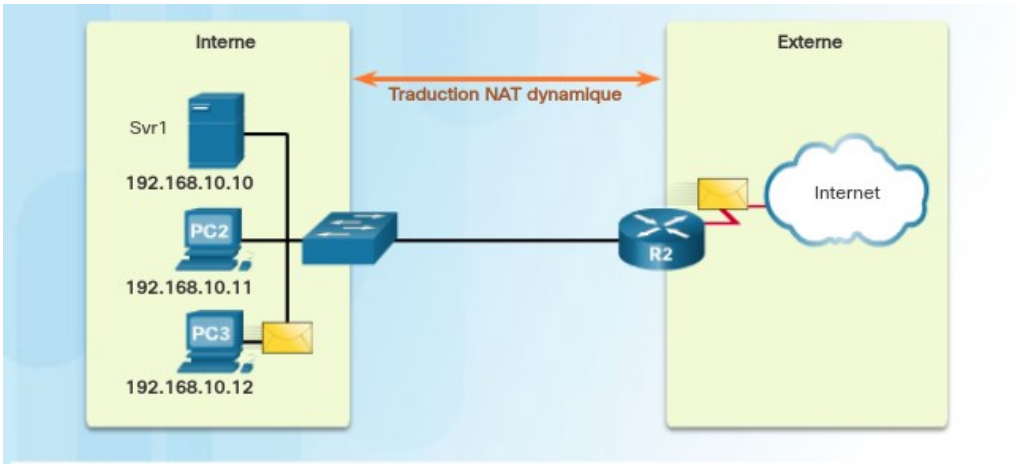


Table NAT statique

Adresse locale interne	Adresse globale interne – adresses accessibles 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

III - NAT Dynamique – Traduction d'adresses dynamique

Le **NAT dynamique** utilise un **pool d'adresses publiques** et les attribue selon la méthode du premier arrivé, premier servi. Lorsqu'un périphérique interne demande l'accès à un réseau externe, le **NAT dynamique** attribue une adresse IPv4 publique disponible du pool.



Pool NAT IPv4

Adresse locale interne	Pool d'adresses globales internes – adresses accessibles via R2
192.168.10.12	209.165.200.226
Disponible	209.165.200.227
Disponible	209.165.200.228
Disponible	209.165.200.229
Disponible	209.165.200.230

IV - PAT (Port Address Translation) – Traduction d'adresses de ports

La traduction d'adresses de port (**PAT**), également appelée **surcharge NAT**, mappe plusieurs adresses IPv4 privées à une seule adresse IPv4 publique unique (ou à quelques adresses) en utilisant les **ports** comme paramètre supplémentaire.

C'est la forme la plus courante de **NAT**.

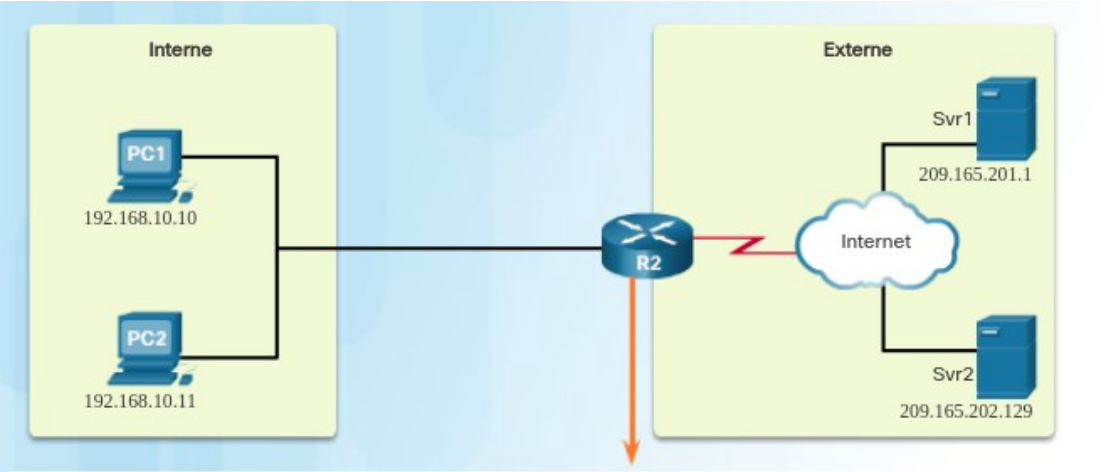


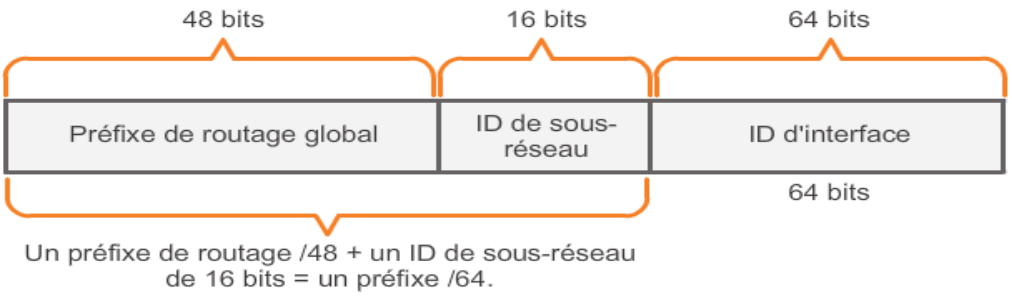
Table NAT avec surcharge

Adresse IP globale interne	Adresse IP locale interne	Adresse IP locale externe	Adresse IP globale externe
209.165.200.226:1555	192.168.10.10:1555	209.165.201.1:80	209.165.201.1:80
209.165.200.226:1331	192.168.10.11:1331	209.165.202.129:80	209.165.202.129:80

17 : Adressage Logique de couche 3 Adresse IPv6

I - Introduction

L'objectif d'**IPv6** est de fournir au moins une adresse publique à chaque équipement. Une adresse **IPv6** est codée sur **16 octets** (**128 bits**) :



le format privilégié pour noter une adresse IPv6 est **x:x:x:x:x:x:x:x**, chaque segment « **x** » de 16 bits comporte quatre valeurs **hexadécimales**.

Par exemple : **2001:0DB8:0000:0000:ABCD:0000:0000:0100**.

II - Format privilégié

Deux règles permettent d'**abréger** la notation des adresses **IPv6** :
- **Omission des zéros** en début de segment de 16 bits ;
- **Remplacement** des segments contenant que **des zéros** par **::** (une fois seulement).

Format Recommandé	Format Sans zéro	Format Compressé
FF02:0000:0000:0000:0000:0000:0000:0001	FF02:0:0:0:0:0:0:1	FF02::1
2001:0DB8:0000:1111:0000:0000:0000:0200	2001:DB8:0:1111:0:0:0:200	2001:DB8:0:1111::200
2001:0DB8:0000:0000:ABCD:0000:0000:0100	2001:DB8:0:0:ABCD:0:0:100	2001:DB8::ABCD:0:0:100 2001:DB8:0:0:ABCD::100
0000:0000:0000:0000:0000:0000:0000:0001 Adresse de **loopback** équivalente à **127.0.0.1**	0:0:0:0:0:0:0:1	::1
0000:0000:0000:0000:0000:0000:0000:0000	0:0:0:0:0:0:0:0	::

III - Préfixe

L'**IPv6** utilise la longueur de **préfixe** pour indiquer la **partie réseau** d'une adresse IPv6 à l'aide de la notation : **adresse IPv6 / longueur de préfixe**.

Par exemple : **2001:0DB8:ACAD:1::10 / 64**.

En **IPv6** les sous-réseaux existent aussi, mais il est conseillé d'utiliser **64 bits** pour le **préfixe** et les divers sous-réseaux et 64 bits pour les stations.

IV - Types d'adresses IPv6

Il existe trois types d'adresses **IPv6** :

- **Unicast** ou **Monodiffusion** : Une adresse de **monodiffusion IPv6** identifie une interface sur un périphérique IPv6 de façon **unique**. Une adresse source IPv6 doit être une adresse de monodiffusion ;
- **Multicast** ou **Multidiffusion** : Elles se trouvent dans la plage **FF00::/8**. Une adresse de multidiffusion IPv6 est utilisée pour envoyer un seul paquet IPv6 vers **plusieurs destinations** . Contrairement à l'IPv4, l'IPv6 n'a pas d'**adresse de diffusion**. Cependant, il existe une adresse de **multidiffusion** à tous les nœuds IPv6 **FF02::1** qui offre globalement les mêmes résultats ;
- **Anycast** : Une adresse **anycast IPv6** est une adresse de **monodiffusion IPv6** qui peut être attribuée à plusieurs périphériques.

Une adresse **Unicast ou Monodiffusion** est :

- Soit une adresse de **lien local (LL : link-local)** : Utilisées pour communiquer avec d'autres périphériques sur la même liaison locale. Les adresses **LL** ne sont pas routables au-delà de la liaison. Les adresses **LL** se trouvent dans la plage **FE80::/10**. **/10** Indique que les 10 premiers bits sont **1111111010**. Le premier hextet dispose d'une plage allant de 1111 1110 10**00 0000** (**FE80**) à 1111 1110 10**11 1111** (**FEBF**). Par exemple, l'adresse : **FE80::3AB1:DBFF:FEF3:FA9 / 64** ;
- Soit une adresse **locale unique (ULA : Unique Local Address)** : Ressemble à une adresse **IPv4 privée**. Les adresses **ULA** sont routables au-delà de la liaison. Les adresses **ULA** sont comprises entre **FC00::/7** et **FDFF::/7**. Par exemple, l'adresse IPv6 : **FDE4:8DBA:82E1::/64** ;
- Soit une adresse de **monodiffusion globale (GA : Global Address)** : Similaire à une adresse **IPv4 publique**. Les adresses **GA** sont routables sur Internet. Seules des adresses **GA** dont les premiers bits sont **001** ou **2000::/3** sont attribuées. Par exemple, l'adresse IPv6 : **2001:0DB8:ACAD:1::10/64**.

V - ID d'interface

Le périphérique client détermine son propre **ID d'interface** de 64 bits, à partir de son adresse **MAC** à l'aide de la méthode **EUI-64** détaillée ci-dessous :