

Version : **2024.01**

Dernière mise-à-jour : 2024/10/10 14:21

# LCF605 - Gestion du Réseau

## Contenu du Module

- **LCF605 - Gestion du Réseau**
  - Contenu du Module
  - Comprendre les Réseaux
    - Présentation des Réseaux
      - Classification des Réseaux
        - Classification par Mode de Transmission
        - Classification par Topologie
        - Classification par Étendue
        - Les Types de LAN
      - Le Modèle Client/Serveur
      - Modèles de Communication
        - Le modèle OSI
        - Spécification NDIS et le Modèle ODI
        - Le modèle TCP/IP
      - Les Raccordements
        - Les Modes de Transmission
        - Les Câbles
        - Les Réseaux sans Fils
        - Le Courant Porteur en Ligne
      - Technologies
        - Ethernet
        - Token-Ring
      - Périphériques Réseaux Spéciaux

- Les Concentrateurs
- Les Répéteurs
- Les Ponts
- Les Commutateurs
- Les Routeurs
- Les Passerelles
- Comprendre le Chiffrement
  - Introduction à la cryptologie
    - Définitions
      - La Cryptographie
      - Le Chiffrement par Substitution
  - Algorithmes à clé secrète
    - Le Chiffrement Symétrique
  - Algorithmes à clef publique
    - Le Chiffrement Asymétrique
    - La Clef de Session
  - Fonctions de Hachage
  - Signature Numérique
  - Utilisation de GnuPG
    - Présentation
    - Installation
    - Configuration
    - Signer un message
    - Chiffrer un message
  - PKI
    - Certificats X509
- Comprendre IPv4
  - En-tête TCP
  - En-tête UDP
  - Fragmentation et Ré-encapsulation
  - Adressage
  - Masques de sous-réseaux
  - VLSM
  - Ports et sockets

- /etc/services
- Résolution d'adresses Ethernet
- Comprendre IPv6
  - Présentation
  - Adresses IPv6
  - Masque de Sous-réseau
  - Adresses IPv6 Réservées
  - L'Adresse Link-local
  - DHCPv6
- Configurer le Réseau
  - La Commande nmcli
- LAB #1 - Configuration du Réseau
  - 1.1 - Connections et Profils
  - 1.2 - Résolution des Noms
  - 1.3 - Ajouter une Deuxième Adresse IP à un Profil
  - 1.4 - La Commande hostname
  - 1.5 - La Commande ip
  - 1.6 - Activer/Désactiver une Interface Manuellement
  - 1.7 - Routage Statique
    - La commande ip
    - Activer/désactiver le routage sur le serveur
- LAB #2 - Diagnostique du Réseau
  - 2.1 - ping
  - 2.2 - netstat -i
  - 2.3 - traceroute
- LAB #3 - Connexions à Distance
  - 3.1 - Telnet
  - 3.2 - wget
  - 3.3 - ftp
  - 3.4 - SSH
    - Présentation
      - SSH-1
      - SSH-2
    - Authentification par mot de passe

- Authentification par clef asymétrique
- Configuration du Serveur
- Configuration du Client
- Tunnels SSH
- 3.5 - SCP
  - Présentation
  - Utilisation
- 3.6 - Mise en Place des Clefs Asymétriques

## Comprendre les Réseaux

### Présentation des Réseaux

La définition d'un réseau peut être résumé ainsi :

- un ensemble d'**Equipements** (systèmes et périphériques) communiquant entre eux,
- une entité destinée au transport de données dans différents environnements.

Pour que la communication soit efficace, elle doit respecter les critères suivants :

- présenter des informations compréhensibles par tous les participants,
- être compatible avec un maximum d'interlocuteurs différents (dans le cas d'un réseau, les interlocuteurs sont des équipements : imprimantes, ordinateurs, clients, serveurs, téléphones...),
- si l'interlocuteur n'est pas disponible, les informations ne doivent pas se perdre,
- permettre une réduction des coûts (par ex. interconnexion à bas coût),
- permettre une productivité accrue (par ex. interconnexion à haut débit),
- être sécurisée si les informations à transmettre sont dites sensibles,
- garantir l'**unicité** et de l'**universalité** de l'**accès à l'information**.

On peut distinguer deux familles d'**Equipements** - les **Eléments Passifs** et les **Eléments Actifs**.

Les **Eléments Passifs** transmettent le signal d'un point à un autre :

- **Les Infrastructures ou Supports** - des câbles, de l'atmosphère ou des fibres optiques permettant de relier **physiquement** des équipements,
- **La Topologie** - l'architecture d'un réseau définissant les connexions entre les **Equipements** et, éventuellement, la hiérarchie entre eux.

Les **Eléments Actifs** sont des équipements qui consomment de l'énergie en traitant ou en interprétant le signal. Les **Equipements** sont classés selon leurs fonctions :

- **Equipement de Distribution Interne au Réseau** - Répartiteur (Hub, Switch, Commutateur etc.), Borne d'accès (Hotspot), Convertisseur de signal (Transceiver), Amplificateur (Répéteur) ...,
- **Equipement d'Interconnexion de Réseaux** - Routeurs, Ponts ...,
- **Nœuds et Interfaces Réseaux** - postes informatiques, équipements en réseau ....

Un **Nœud** est une extrémité de connexion qui peut être une intersection de plusieurs connexions ou de plusieurs **Equipements**.

Une **Interface Réseau** est une prise ou élément d'un **Equipement Actif** faisant la connexion vers d'autres **Equipements** réseaux et qui reçoit et émet des données.



**Important** - Dans le cas d'un mélange d'**Equipements** non-homogènes en termes de performances au sein du même réseau, c'est la loi du plus faible qui emporte.

Tous les **Equipements** connectés au même support doivent respecter un ensemble de règles appelé une **Protocole de Communication**.

Les **Protocoles de Communication** définissent de façon formelle et interopérable la manière dont les informations sont échangées entre les **Equipements**.

Des **Logiciels**, dédiés à la gestion de ces **Protocoles de Communication**, sont installés sur des **Equipements d'Interconnexion** afin de fournir des fonctions de contrôle permettant une communication entre les **Equipements**.

Se basant sur des **Protocoles de Communication**, des **Services** fournissent des fonctionnalités accessibles aux utilisateurs ou d'autres programmes.

L'ensemble des **Equipements**, **Logiciels** et **Protocoles de Communication** constitue l'**Architecture Réseau**.

## Classification des Réseaux

Les réseaux peuvent être classifiés de trois façon différentes :

- par **Mode de Transmission**,
- par **Topologie**,
- par **Étendue**.

### Classification par Mode de Transmission

Il existe deux **Classes** de réseaux dans cette classification :

- les **Réseaux en Mode de Diffusion**,
  - utilise un seul support de transmission,
  - le message est envoyé sur tout le réseau à l'adresse d'**un** destinataire,
- les **Réseaux en Mode Point à Point**,
  - une seule liaison entre deux équipements,
  - les nœuds permettent de choisir la route en fonction de l'adresse du destinataire,
  - quand deux nœuds non directement connectés entre eux veulent communiquer ils le font par l'intermédiaire des autres noeuds du réseau.

### Classification par Topologie



**Important** - La **Topologie Physique** d'un réseau décrit l'organisation de ce dernier en termes de câblage. La **Topologie Logique** d'un réseau décrit comment les données circulent sur le réseau. En effet c'est le choix des concentrateurs ainsi que les connections des câbles qui déterminent la topologie logique.

### La Topologie Physique

Il existe 6 topologies physiques de réseau :

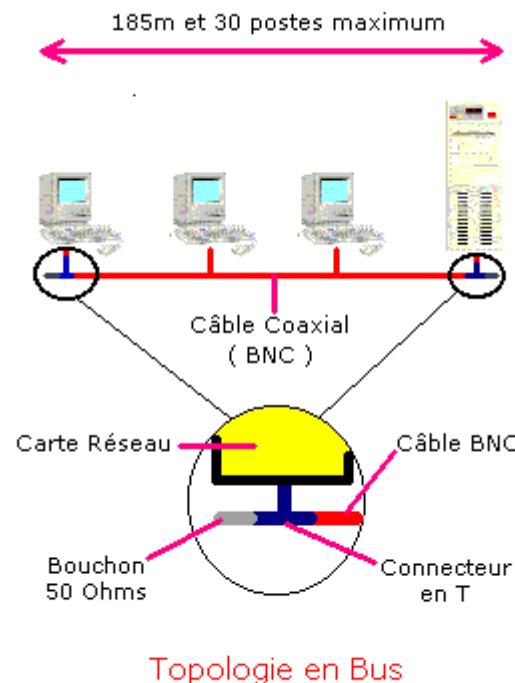
- La Topologie en Ligne,
- La Topologie en Bus,
- La Topologie en Etoile,
- La Topologie en Anneau,
- La Topologie en Arbre,
- La Topologie Maillée.

### **La Topologie en Ligne**

Tous les nœuds sont connectés à un seul support. L'inconvénient de cette topologie est que dans le cas d'une défaillance d'une station, le réseau se trouve coupé en deux sous-réseaux.

### **La Topologie en Bus**

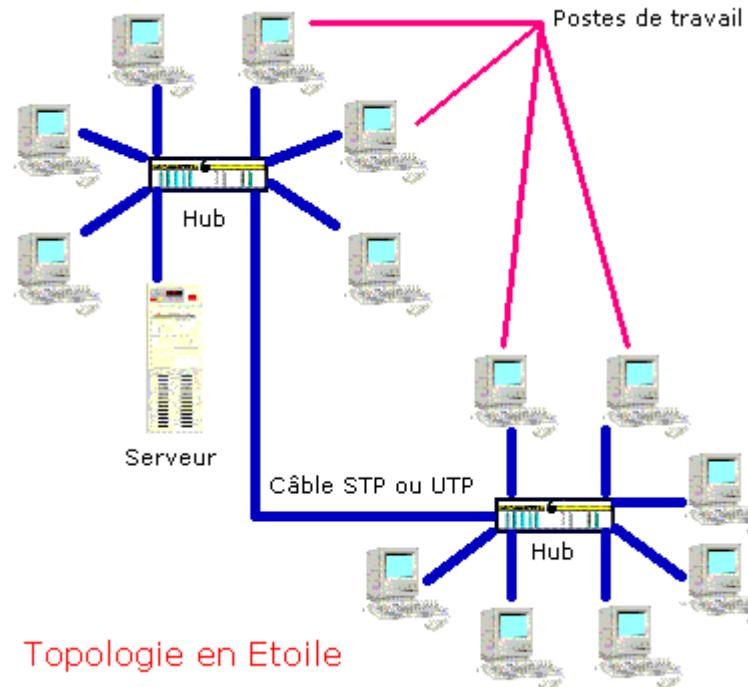
Tous les nœuds sont connectés à un seul support (un câble BNC en T) avec des bouchons à chaque extrémité. La longueur du bus est limitée à **185m**. Le nombre de stations de travail est limité à **30**. Les Stations sont reliées au Bus par des 'T'. Les bouchons sont des terminateurs qui sont des résistances de **50 Ohms**. Quand le support tombe en panne, le réseau ne fonctionne plus. Quand une station tombe en panne, elle ne perturbe pas le fonctionnement de l'ensemble du réseau. Les Stations étant reliés à un seul support, ce type de topologie nécessite un **Protocole d'Accès** pour gérer le tour de parole des Stations afin d'éviter des conflits.



Topologie en Bus

## La Topologie en Étoile

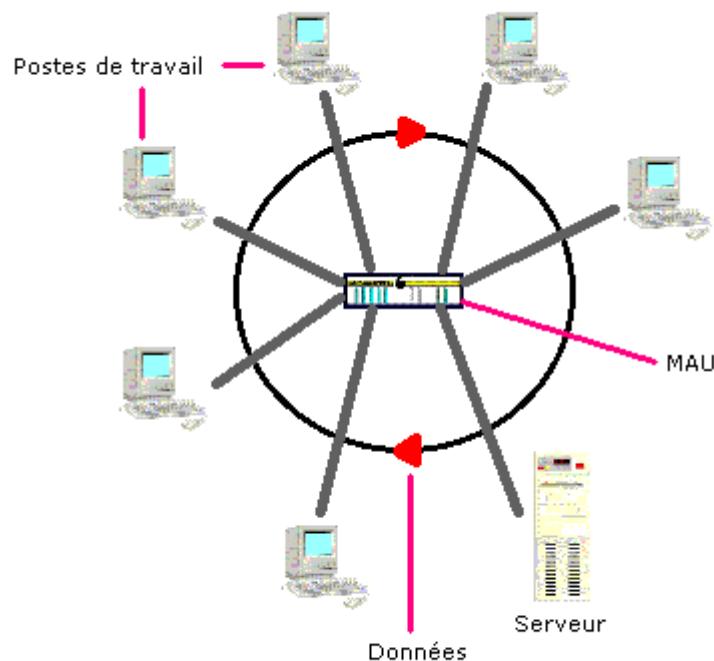
Chaque nœud est connecté à un périphérique central appelé un **Hub (Concentrateur)** ou un **Switch (Commutateur)**. Un Hub ou un Switch est prévu pour 4, 8, 16, 32 ... stations. En cas d'un réseau d'un plus grand nombre de stations, plusieurs Hubs ou Switches sont connectés ensemble. Quand une station tombe en panne, elle ne perturbe pas le fonctionnement de l'ensemble du réseau. Le point faible de cette topologie est l'équipement central.



Topologie en Etoile

### La Topologie en Anneau

Chaque nœud est relié directement à ses deux voisins dans une topologie logique de cercle ininterrompu et une topologie physique en étoile car les stations sont reliées à un type de hub spécial, appelé un **Multistation Access Unit** (MAU).



### Topologie en Anneau

Les stations sont reliées à la MAU par un câble 'IBM' munie d'une prise **AUI** du côté de la carte et une prise **Hermaphrodite** du coté de la MAU. Les données sont échangées dans un sens unidirectionnel. Une trame, appelée un **jeton**, circule en permanence. Si l'anneau est brisé, l'ensemble du réseau s'arrête. Pour cette raison, il est courant de voir deux anneaux contre-rotatifs.

### La Topologie en Arbre

La Topologie en Arbre est utilisée dans un réseau hiérarchique où le sommet, aussi appelé la **racine**, est connecté à plusieurs noeuds de niveau inférieur. Ces noeuds peuvent à leur tour être connectés à d'autres noeuds inférieurs. L'ensemble forme une arborescence. Le point faible de cette topologie est sa racine. En cas de défaillance, le réseau est coupé en deux.

### La Topologie Maillée

Cette Topologie est utilisée pour des grands réseaux de distribution tels Internet ou le WIFI. Chaque noeud à tous les autres via des liaisons point à

point. Le nombre de liaisons devient très rapidement important en cas d'un grand nombre de noeuds. Par exemple dans le cas de 100 Stations (N), le nombre de liaisons est obtenu par la formule suivante :

$$N(N-1)/2 = 100(100-1)/2 = 4\ 950$$



**Important** - La **Topologie Physique** la plus répandue est la **Topologie en Etoile**.

### Classification par Etendue

La classification par étendue nous fournit 4 réseaux principaux :

| Nom | Description                | Traduction                    | Taille Approximative (M) |
|-----|----------------------------|-------------------------------|--------------------------|
| PAN | Personal Area Network      | Réseau Personnel              | 1 -10                    |
| LAN | Local Area Network         | Réseau Local Entreprise (RLE) | 5 - 1 200                |
| MAN | Métropolitain Area Network | Réseau Urbain                 | 900 - 100 000            |
| WAN | Wide Area Network          | Réseau Long Distance (RLD)    | 50 000 et au delà        |

Cependant, d'autres classification existent :

|     |                      |                    |
|-----|----------------------|--------------------|
| CAN | Campus Area Network  | Réseau de Campus   |
| GAN | Global Area Network  | Réseau Global      |
| TAN | Tiny Area Network    | Réseau Minuscule   |
| FAN | Family Area Network  | Réseau Familial    |
| SAN | Storage Area Network | Réseau de Stockage |



**Important** - Etant donné que les WANs sont gérés par des opérateurs de télécommunications qui doivent demander une licence à l'état mais que les LANs ont été historiquement mis en oeuvre dans les entreprises, ces derniers sont en majorité issus du



monde informatique.

## Les Types de LAN

Il existe deux types de LAN :

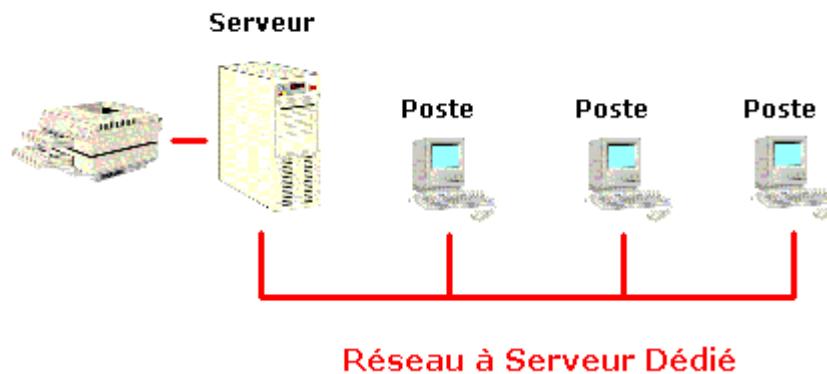
- le réseau à serveur dédié,
- le réseau poste à poste.

### Réseau à Serveur Dédié

Le réseau à serveur dédié est caractérisé par le fait que toutes les ressources ( imprimantes, applications, lecteurs etc. ) sont gérées par le serveur. Les autres micro-ordinateurs ne jouent le rôle de client.

Des exemples des systèmes d'exploitation du réseau à serveur dédié sont :

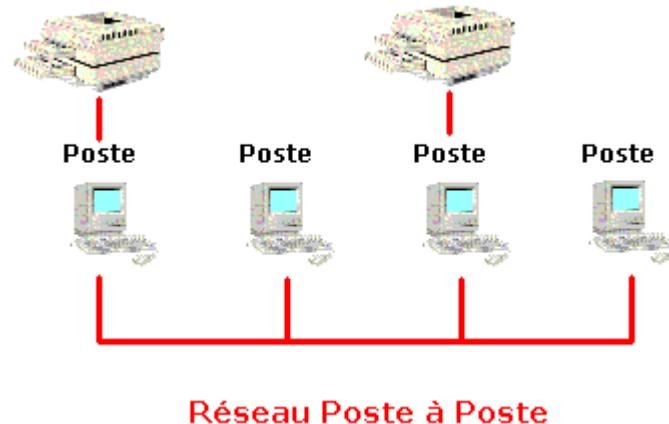
- Windows NT Server,
- Windows 2000 Server,
- Windows 2003 Server,
- Windows 2008 Server,
- Linux,
- Unix.



### Réseau Poste-à-Poste

Le réseau poste à poste est caractérisé par le fait que tous les ordinateurs peuvent jouer le rôle de client et de serveur :

- Windows 95,
- Windows 98,
- Windows NT Workstation.



## Le Modèle Client/Serveur

Le modèle Client/Serveur est une des modalités des architectures informatiques distribuées. Dans ce modèle un serveur est tout **Logiciel** fournissant un **Service**.

Le serveur est aussi :

- passif, c'est-à-dire en attente permanente d'une demande, appelée une requête d'un client,
- capable de traiter plusieurs requêtes simultanément en utilisant le **multi-threading**,
- garant de l'intégrité globale.

Le client est, par contre **actif**, étant à l'origine des requêtes.

Il existe trois types de modèle client/serveur :

- **Plat** - tous les clients communiquent avec un seul serveur,
- **Hiérarchique** - les clients n'ont de contact qu'avec les serveurs de plus haut niveau qu'eux,
- **Peer-to-Peer** - les équipements sont à la fois client **et** serveur en même temps.

## Modèles de Communication

Les réseaux sont bâtis sur des technologies et des modèles. Le modèle **théorique** le plus important est le modèle **Open System Interconnection** créé par l'**International Organization for Standardization** tandis que le modèle pratique le plus important est le modèle **TCP/IP**.

### Le modèle OSI

Le modèle OSI qui a été proposé par l'ISO est devenu le standard en termes de modèle pour décrire l'échange de données entre ordinateurs. Cette norme se repose sur sept couches, de la une - la Couche Physique, à la sept - la Couche d'Application, appelés des services. La communication entre les différentes couches est synchronisée entre le poste émetteur et le poste récepteur grâce à ce que l'on appelle un protocole.

Ce modèle repose sur trois termes :

- Les **Couches**,
- Les **Protocoles**,
- Les **Interfaces**.

## Les Couches

Des sept couches :

- Les couches 1 à 3 sont les **Couches Basses** orientées **Transmission**,
- La couche 4 est la **Couche Charnière** entre les **Couches Basses** et les **Couches Hautes**,
- Les couches 5 à 7 sont les **Couches Hautes** orientées **Traitement**.

La couche du même niveau du système **A** parle avec son homologue du système **B**.

- **La Couche Physique** ( Couche 1 ) est responsable :
  - du transfert de données binaires sur le câble physique ou virtuel
  - de la définition de tout aspect physique allant du connecteur jusqu'au câble en passant par la carte réseau, y compris l'organisation même du réseau
  - de la définition des tensions électriques sur le câble pour obtenir le 0 et le 1 binaires
- **La Couche de Liaison** ( Couche 2 ) est responsable :
  - de la réception des données de la couche physique
  - de l'organisation des données en fragments, appelés des trames qui ont un format différent selon s'il s'agit d'un réseau basé sur la technologie Ethernet ou la technologie Token-Ring
  - de la préparation, émission et réception des trames
  - de la gestion de l'accès au réseau
  - de la communication nœud à nœud
  - de la gestion des erreurs
    - avant la transmission, le nœud émetteur calcule un code appelé un CRC et l'incorpore dans les données envoyées
    - le nœud récepteur recalcule un CRC en fonction du contenu de la trame reçue et le compare à celui incorporé avec l'envoi
    - en cas de deux CRC identiques, le nœud récepteur envoie un accusé de réception au nœud émetteur
  - de la réception de l'accusé de réception
  - éventuellement de la ré-émission des données
  - En prenant ce modèle, l'IEEE ( Institute of Electrical and Electronics Engineers ) l'a étendu avec le Modèle IEEE ( 802 ).
    - Dans ce modèle la Couche de Liaison est divisée en deux sous-couches importantes :

- La **Sous-Couche LLC** ( Logical Link Control ) qui :
  - gère les accusés de réception
  - gère le flux de trames
- La **Sous-Couche MAC** ( Media Access Control ) qui :
  - gère la méthode d'accès au réseau
  - le CSMA/CD dans un réseau basé sur la technologie Ethernet
  - l'accès au jeton dans un réseau basé sur la technologie Token-Ring
  - gère les erreurs
- **La Couche de Réseau** ( Couche 3 ) est responsable de la gestion de la bonne distribution des différentes informations aux bonnes adresses en :
  - identifiant le chemin à emprunter d'un nœud donné à un autre
  - appliquant une conversion des adresses logiques ( des noms ) en adresses physiques
  - ajoutant des informations d'adressage aux envois
  - détectant des paquets trop volumineux avant l'envoi et en les divisant en trames de données de tailles autorisées
- **La Couche de Transport** ( Couche 4 ) est responsable de veiller à ce que les données soient envoyées correctement en :
  - constituant des paquets de données corrects
  - les envoyant dans le bon ordre
  - vérifiant que les données sont traitées dans le même ordre que l'ordre d'émission
  - permettant à un processus sur un nœud de communiquer avec un autre nœud et d'échanger des messages avec lui
- **La Couche de Session** ( Couche 5 ) est responsable :
  - de l'établissement, du maintien, et de la mise à fin de la communication entre deux noeuds distants, c'est-à-dire, de la session
  - de la conversation entre deux processus de vérification de la réception des messages envoyés en séquences, c'est-à-dire, le point de contrôle
  - de la sécurité lors de l'ouverture de la session, c'est-à-dire, les droits d'utilisateurs etc.
- **La Couche de Présentation** ( Couche 6 ) est responsable :
  - du formatage et de la mise en forme des données
  - des conversions de données telles le cryptage/décryptage
- **La Couche d'Application** ( Couche 7 ) est responsable :
  - du dialogue homme/machine via des messages affichés
  - du partage des ressources
  - de la messagerie

## Les Protocoles

Un **protocole** est un langage commun utilisé par deux entités en communication pour pouvoir se comprendre. La nature du Protocole dépend directement de la nature de la communication. Cette bature dépend du **paradigme** de communication que l'application nécessite. Le paradigme est un modèle abstrait d'un problème ou d'une situation. Dans le paradigme de la diffusion, l'émetteur envoie dans informations au récepteur sans se soucier de ce que le récepteur va en faire. C'est la responsabilité du récepteur de comprendre et d'utiliser les informations.

## Les Interfaces

Chaque couche rend des **services** à la couche immédiatement supérieure et utilise les services de la couche immédiatement inférieure. L'ensemble des services s'appelle une **Interface**. Les services sont composés de **Service Data Units** et sont disponibles par un **Service Access Point**.

### Protocol Data Units

L'**Unité de Données** ou *Protocol Data Unit* pour chaque couche comporte un nom spécifique :

- **Application Protocol Data Units** pour la couche **Application**,
- **Présentation Protocol Data Units** pour la couche **Présentation**,
- **Session Protocol Data Units** pour la couche **Session**,
- **Transport Protocol Data Units** pour la couche **Transport**.

Or, pour les **Couches Basses** on parle de :

- **Paquets** pour la couche **Réseau**,
- **Trames** pour la couche **Liaison**,
- **Bits** pour la couche **Physique**.

### Encapsulation et Désencapsulation

Lorsque les données sont communiquées par le système A au système B, celles-ci commencent au niveau de la couche d'Application. Le couche d'Application ajoute une en-tête à l'unité de données qui contient des **informations de contrôle du protocole**. Au passage de chaque couche, celle-ci ajoute sa propre en-tête. De cette façon, lors de sa descente vers la couche physique, les données et l'en-tête de la couche supérieure sont encapsulées :

| Couche Système A | Encapsulation                                   |
|------------------|---|
| Application      | Application Header (AH) + Unité de Données (UD) |

| <b>Couche Système A Encapsulation</b> |   |
|---------------------------------------|---|
| Présentation                          | Présentation Header (PH) + AH + UD                |
| Session                               | Session Header (SH) + PH + AH + UD                |
| Transport                             | Transport Header (TH) + SH + PH + AH + UD         |
| Réseau                                | Network Header (NH) + TH + SH + PH + AH + UD      |
| Liaison                               | Liaison Header (DH) + NH + TH + SH + PH + AH + UD |

Lors de son voyage de la couche Physique vers la couche Application dans le système B, les en-têtes sont supprimées par chaque couche correspondante. On parle alors de **désencapsulation** :

| <b>Couche Système B Encapsulation</b> |   |
|---------------------------------------|---|
| Liaison                               | Liaison Header (DH) + NH + TH + SH + PH + AH + UD |
| Réseau                                | Network Header (NH) + TH + SH + PH + AH + UD      |
| Transport                             | Transport Header (TH) + SH + PH + AH + UD         |
| Session                               | Session Header (SH) + PH + AH + UD                |
| Présentation                          | Présentation Header (PH) + AH + UD                |
| Application                           | Application Header (AH) + Unité de Données (UD)   |

### Spécification NDIS et le Modèle ODI

La spécification NDIS ( Network Driver Interface Specification ) a été introduite conjointement par les sociétés Microsoft et 3Com. Cette spécification ainsi que son homologue, le modèle ODI ( Open Datalink Interface ) introduit conjointement par les sociétés Novell et Apple à la même époque, définit des standards pour les pilotes de cartes réseau afin qu'ils puissent être indépendants des protocoles utilisées et les systèmes d'exploitation sur les machines. Des deux 'standards', la spécification NDIS est le plus répandu, intervenant à niveau de la sous-couche MAC et à la couche de liaison. Elle spécifie :

- l'interface pilote-matériel
- l'interface pilote-protocole
- l'interface pilote - système d'exploitation

## Le modèle TCP/IP

La suite des protocoles TCP/IP ( Transmission Control Protocol / Internet Protocol ) est issu de la DOD ( Dept. Américain de la Défense ) et le travail de l'ARPA ( Advanced Research Project Agency ).

- La suite des protocoles TCP/IP
  - a été introduite en 1974
  - a été utilisée dans l'ARPAnet en 1975
  - permet la communication entre des réseaux à base de systèmes d'exploitation, architectures et technologies différents
  - est très proche du modèle OSI en termes d'architecture et se place au niveau de la couche d'Application jusqu'à la couche Réseau.
  - est, en réalité, une suite de protocoles et de services :
    - **IP** ( Internet Protocol )
      - le protocole IP s'intègre dans la couche Réseau du modèle OSI en assurant la communication entre les systèmes. Bien qu'il puisse découper des messages en fragments ou datagrammes et les reconstituer dans le bon ordre à l'arrivée, il ne garantit pas la réception.
    - **ICMP** ( Internet Control Message Protocol )
      - le protocole ICMP produit des messages de contrôle aidant à synchroniser le réseau. Un exemple de ceci est la commande ping.
    - **TCP** ( Transmission Control Protocol )
      - le protocole TCP se trouve au niveau de la couche de Transport du modèle OSI et s'occupe de la transmission des données entre noeuds.
    - **UDP** ( User Datagram Protocol )
      - le protocole UDP n'est pas orienté connexion. Il est utilisé pour la transmission rapide de messages entre nœuds sans garantir leur acheminement.
    - **Telnet**
      - le protocole Telnet est utilisé pour établir une connexion de terminal à distance. Il se trouve dans la couche d'Application du modèle OSI.
    - **Ftp** ( File Transfer Protocol )
      - le protocole ftp est utilisé pour le transfert de fichiers. Il se trouve dans la couche d'Application du modèle OSI.
    - **SMTP** ( Simple Message Transfer Protocol )
      - le service SMTP est utilisé pour le transfert de courrier électronique. Il se trouve dans la couche d'Application du modèle OSI.
    - **DNS** ( Domain Name Service )
      - le service DNS est utilisé pour la résolution de noms en adresses IP. Il se trouve dans la couche d'Application du modèle OSI.
    - **SNMP** ( Simple Network Management Protocol )

- le protocole SNMP est composé d'un agent et un gestionnaire. L'agent SNMP collecte des informations sur les périphériques, les configurations et les performances tandis que le gestionnaire SNMP reçoit ses informations et réagit en conséquence.
- **NFS** ( Network File System )
  - le NFS a été mis au point par Sun Microsystems
  - le NFS génère un lien virtuel entre les lecteurs et les disques durs permettant de monter dans un disque virtuel local un disque distant
  - et aussi POP3, NNTP, IMAP etc ...

Le modèle TCP/IP est composé de 4 couches :

- La couche d'Accès Réseau
  - Cette couche spécifie la forme sous laquelle les données doivent être acheminées, quelque soit le type de réseau utilisé.
- La couche Internet
  - Cette couche est chargée de fournir le paquet de données.
- La couche de Transport
  - Cette couche assure l'acheminement des données et se charge des mécanismes permettant de connaître l'état de la transmission.
- La couche d'Application
  - Cette couche englobe les applications standards de réseau telles ftp, telnet, ssh, etc..

Les noms des Unités de Données sont différents selon le protocole utilisé et la couche du modèle TCP/IP :

| Couche      | TCP      | UDP      |
|-------------|----------|----------|
| Application | Stream   | Message  |
| Transport   | Segment  | Packet   |
| Internet    | Datagram | Datagram |
| Réseau      | Frame    | Frame    |

## Les Raccordements

### Les Modes de Transmission

On peut distinguer 3 modes de transmission :

- La **Liaison Simplex**,
  - Les données ne circulent que dans un **seul** sens de l'émetteur vers le récepteur,
  - La liaison nécessite deux canaux de transmissions,
- La **Liaison Half-Duplex** aussi appelée la **Liaison à l'Alternat** ou encore la **Liaison Semi-Duplex**,
  - Les données circulent dans un sens ou l'autre mais jamais dans les deux sens en même temps. Chaque extrémité émet donc à son tour,
  - La liaison permet d'avoir une liaison bi-directionnelle qui utilise la totalité de la bande passante,
- La **Liaison Full-Duplex** dans les deux sens en **même** temps. Chaque extrémité peut émettre et recevoir simultanément,
  - La liaison est caractérisée par une bande passante divisée par deux pour chaque sens des émissions.

## Les Câbles

### Le Câble Coaxial

En partant de l'extérieur, le câble coaxial est composé :

- d'une **Gaine** en caoutchouc, PVC ou Téflon pour protéger le câble,
- d'un **Blindage** en métal pour diminuer le bruit des parasites,
- d'un **Isolant** (diélectrique) pour éviter le contact entre le blindage et l'âme et ainsi éviter des courts-circuits,
- d'un **Âme** en cuivre ou torsadés pour transporter les données.

Avantages :

- **Peux coûteux**,
- Facilement **manipulable**,
- Peut être utilisé pour de **longues distances**,
- A un débit de 10 Mbit/s dans un LAN et 100 Mbit/s dans un WAN.

Inconvénients :

- Fragile,
- Instable,
- Vulnérable aux interférences,
- Half-Duplex.

## Le Câble Paire Torsadée

Ce câble existe sous deux formes selon son utilisation :

- **Monobrin** pour du câblage **horizontal (Capillaire)**,
  - chaque fil est composé d'un seul conducteur en cuivre,
  - la distance ne doit pas dépasser 90m.
- **Multibrin** pour des **cordons de brassage** :
  - chaque fil est composé de plusieurs brins en cuivre,
  - câble souple.

Avantages :

- Un débit de 10 Mbit/s à 10 GBit/s,
- A une bande passante plus large,
- Pas d'interruption par coupure du câble,
- Permet le **câblage universel** (téléphonie, fax, données ...),
- Full-Duplex.

Inconvénients :

- Nombre de câbles > câble coaxial,
- Plus cher,
- Plus encombrant dans les gaines techniques.

## Catagories de Blindage

Il existe trois catagories de blindage :

- **Twisted** ou Torsadé,
- **Foiled** ou Entouré,
- **Shielded** ou Avec Ecran.

De ce fait, il existe 5 catagories de câbles Paire Torsadée :

Nom anglais ^ Appelation Ancienne ^ Nouvelle Appelation ^

|                            |       |         |
|----------------------------|-------|---------|
| Unshielded Twisted Pair    | UTP   | U/UTP   |
| Foiled Twisted Pair        | FTP   | F/UTP   |
| Shield Twisted Pair        | STP   | S/UTP   |
| Shield Foiled Twisted Pair | SFTP  | SF/UTP  |
| Shield Shield Twisted Pair | S/STP | SS/STP3 |

Ces catégories donnent lieu à des **Classes** :

| Classe | Débit  | Nombre de Paires Torsadées | Connecteur     | Commentaires  |
|--------|--|----------------------------|----------------|---|
| 3      | 10 Mbit/s                                      | 4                          | RJ11           |   |
| 4      | 16 Mbit/s                                      | 4                          | S/O            | Non-utilisée de nos jours   |
| 5      | 100 Mbit/s                                     | 4                          | RJ45           | Obsolète  |
| 5e/D   | 1 Gbit/s sur 100m                              | 4                          | RJ45           | S/O   |
| 6/E    | 2.5 Gbit/s sur 100m ou 10 Gbit/s sur 25m à 55m | 4                          | Idéal pour PoE |   |
| 7/F    | 10 Gbit/s sur 100m                             | 4                          | GG45 ou Tera   | Paires individuellement et collectivement blindées. Problème de compatibilité avec les classes précédentes due au connecteur. |

## La Prise RJ45

Une prise RJ45 comporte 8 broches. Un câble peut être **droit** quand la broche 1 d'une extrémité est connectée à la broche 1 de la prise RJ45 à l'autre extrémité, la broche 2 d'une extrémité est connectée à la broche 2 de la prise RJ45 à l'autre extrémité et ainsi de suite ou bien **croisé** quand le brochage est inversé.

Les câbles croisés sont utilisés lors du branchement de deux équipements identiques (PC à PC, Hub à Hub, Routeur à Routeur).

## Channel Link et Basic Link

Le **Channel Link** ou **Canal** est l'ensemble du **Basic Link** ou **Lien** de base et les cordons de brassage et de raccordement des équipements qui sont limités en distance à 10m.

Le **Basic Link** est le lien entre la prise RJ45 murale et la baie de brassage. Il est limité à 90m en classe 5D.

### La Fibre Optique

La **Fibre Optique** est un fil de **Silice** permettant le transfert de la lumière. De ce fait elle est caractérisée par :

- des meilleures performances que le cuivre,
- de plus de communications simultanément,
- de la capacité de relier de plus grandes distances,
- une insensibilité aux perturbations,
- une résistance à la corrosion.

Qui plus est, elle ne produit aucune perturbation.

Elle est composée :

- d'un coeur de 10, de 50/125 ou de 62.50 micron,
- d'une gaine de 125 micron,
- d'une protection de 230 micron.

Il existe deux types de fibres, la **Fibre Monomode** et la **Fibre Multimodes**.

La Fibre Monomode :

- a un coeur de 8 à 10 Microns,
- est divisée en sous-catégories de distance,
  - 10 Km,
  - 15 Km,
  - 20 Km,
  - 50 Km,
  - 80 Km,
  - 100 Km.

La Fibre Multimode :

- a un cœur de 62,50 micron ou de 50/125 micron avec une gaine orange,
- permet plusieurs trajets lumineux appelés **modes** en même temps en Full Duplex,
- est utilisée pour de bas débits ou de courtes distances,
  - 2 Km pour 100 Mbit/s,
  - 500 m pour 1 Gbit/s.

## Les Réseaux sans Fils

Les réseaux sans fils sont basés sur une liaison qui utilise des ondes radio-électriques (radio et infra-rouges).

Il existe des technologies différentes en fonction de la fréquence utilisée et de la portée des transmissions :

- Réseaux Personnels sans Fils - Bluetooth, HomeRF,
- Réseaux Locaux sans Fils - LiFi, WiFi,
- Réseaux Métropolitains sans Fil - WiMax,
- Réseaux Etendus sans Fils - GSM, GPRS, UMTS.

Les principales ondes utilisées pour la transmission des données sont :

- Ondes GSM - Ondes Hertziennes reposant sur des micro-ondes à basse fréquence avec une portée d'une dizaine de kilomètres,
- Ondes Wi-Fi - Ondes Hertziennes reposant sur des micro-ondes à haute fréquence avec une portée de 20 à 50 mètres,
- Ondes Satellitaires - Ondes Hertziennes longues portées.

## Le Courant Porteur en Ligne

Le CPL utilise le réseau électrique domestique, le réseau moyenne et basse tension pour transmettre des informations numériques.

Le CPL superpose un signal à plus haute fréquence au signal électrique.

Seuls donc, les fils conducteurs transportent les signaux CPL.

Le coupleur intégré en entrée des boîtiers CPL élimine les composants basses fréquences pour isoler le signal CPL.

Le CPL utilise la phase électrique et le neutre. De ce fait, une installation triphasée fournit 3 réseaux CPL différents.

Le signal CPL ne s'arrête pas nécessairement aux limites de l'installation électrique. En effet en cas de compteurs non-numériques le signal les traversent.

Les normes CPL sont :

| Norme         | Débit Théorique | Débit Pratique | Temps pour copier 1 Go |
|---------------|-----------------|----------------|------------------------|
| Homeplug 1.01 | 14 Mbps         | 5.4 Mbps       | 25m 20s                |
| Homeplug 1.1  | 85 Mbps         | 12 Mbps        | 11m 20s                |
| PréUPA 200    | 200 Mbps        | 30 Mbps        | 4m 30s                 |

## Technologies

Il existe plusieurs technologies de réseau :

- Ethernet,
- Token-Ring,
- ARCnet,
- etc..

Nous détaillerons ici les deux technologies les plus répandues, à savoir Ethernet et Token-Ring.

### Ethernet

La technologie Ethernet se repose sur :

- une topologie logique de bus,
- une topologie physique de bus ou étoile.

L'accès au bus utilise le **CSMA/CD**, Carrier Sense Multiple Access / Collision Detection (Accès Multiple à Détection de Porteuse / Détection de Collisions).

Il faut noter que :

- les données sont transmises à chaque nœud - c'est la méthode d'**accès multiple**,
- chaque nœud qui veut émettre écoute le réseau - c'est la **détection de porteuse**,
- quand le réseau est silencieux une trame est émise dans laquelle se trouvent les données ainsi que l'adresse du destinataire,
- le système est dit donc **aléatoire** ou **non-déterministe**,
- quand deux nœuds émettent en même temps, il y a **collision de données**,
- les deux nœuds vont donc cesser d'émettre, se mettant en attente jusqu'à ce qu'ils commencent à émettre de nouveau.

## Token-Ring

La technologie Token-Ring se repose sur :

- une topologie logique en anneau,
- une topologie physique en étoile.

Token-Ring se traduit par **Anneau à Jeton**. Il n'est pas aussi répandu que l'Ethernet pour des raisons de coûts. En effet le rajout d'un nœud en Token-Ring peut coûter jusqu'à **4 fois plus cher qu'en Ethernet**.

Il faut noter que :

- les données sont transmises dans le réseau par un système appelé **méthode de passage de jeton**,
- le jeton est une **trame numérique vide** de données qui tourne en permanence dans l'anneau,
- quand un nœud souhaite émettre, il saisit le jeton, y dépose des données avec l'adresse du destinataire et ensuite laisse poursuivre son chemin jusqu'à sa destination,
- pendant son voyage, aucun autre nœud ne peut émettre,
- une fois arrivé à sa destination, le jeton dépose ses données et retourne à l'émetteur pour confirmer la livraison,
- ce système est appelé **déterministe**.

L'intérêt de la technologie Token-Ring se trouve dans le fait :

- qu'il **évite des collisions**,
- qu'il est **possible de déterminer avec exactitude le temps que prenne l'acheminement des données**.

La technologie Token-Ring est donc idéale, voire obligatoire, dans des installations où chaque nœud doit disposer d'une opportunité à intervalle fixe d'émettre des données.

## Périphériques Réseaux Spéciaux

En plus du câblage, les périphériques de réseau spéciaux sont des éléments primordiaux tant au niveau de la topologie physique que la topologie logique.

Les périphériques de réseau spéciaux sont :

- les Concentrateurs ou *Hubs*,
- les Répéteurs ou *Repeaters*,
- les Ponts ou *Bridges*,
- les Commutateurs ou *Switches*,
- les Routeurs ou *Routers*,
- les Passerelles ou *Gateways*.

L'objectif ici est de vous permettre de comprendre le rôle de chaque périphérique.

### Les Concentrateurs

Les Concentrateurs permettent une connectivité entre les nœuds en topologie en étoile. Selon leur configuration, la topologie logique peut être en étoile, en bus ou en anneau. Il existe de multiples types de Concentrateurs allant du plus simple au Concentrateur intelligent.

- **Le Concentrateur Simple**

- est une boîte de raccordement centrale,
- joue le rôle de récepteur et du réémetteur des signaux sans accélération ni gestion de ceux-ci,
- est un périphérique utilisé pour des groupes de travail.

- **Le Concentrateur Évolué**

- est un Concentrateur simple qui offre en plus l'amplification des signaux, la gestion du type de topologie logique grâce à des capacités d'être configurés à l'aide d'un logiciel ainsi que l'homogénéisation du réseau en offrant des ports pour un câblage différent. Par exemple, 8 ports en paire torsadée non-blindée et un port BNC.

- **Le Concentrateur Intelligent**

- est un Concentrateur évolutif qui offre en plus la détection automatique des pannes, la connectique avec un Pont ou un Routeur ainsi que le

diagnostic et la génération de rapports.

## Les Répéteurs

Un Répéteur est un périphérique réseau simple. Il est utilisé pour amplifier le signal quand :

- la longueur du câble dépasse la limite autorisée,
- le câble passe par une zone où les interférences sont importantes.

Éventuellement, et uniquement dans le cas où le Répéteur serait muni d'une telle fonction, celui-ci peut être utilisé pour connecter deux réseaux ayant un câblage différent.

## Les Ponts

Un Pont est **Répéteur intelligent**. Outre sa capacité d'amplifier les signaux, le Pont analyse le trafic qui passe par lui et met à jour une liste d'adresses des cartes réseau, appelée **une table de routage**, n'autorisant que les transmissions destinées à d'autres segments du réseau.

Les **diffusions** sont néanmoins autorisées.

Comme un Pont doit être intelligent, on utilise souvent un micro-ordinateur comme Pont. Forcément équipé de 2 cartes réseau, le Pont peut également jouer le rôle de serveur de fichiers.

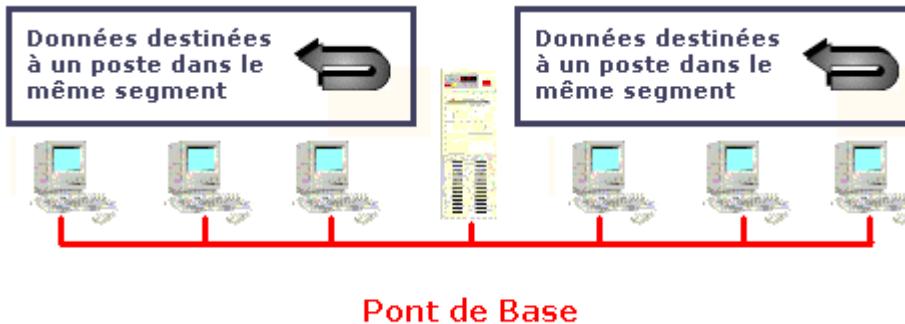
Le Pont sert donc à isoler des segments du réseau pour des raisons de :

- **sécurité** afin d'éviter à ce que des données sensibles soient propagées sur tout le réseau,
- **performance** afin qu'une partie du réseau trop chargée ralentisse le réseau entier,
- **fiabilité** afin par exemple qu'une carte en panne ne gêne pas le reste du réseau avec une diffusion.

Il existe trois types de configuration de Ponts

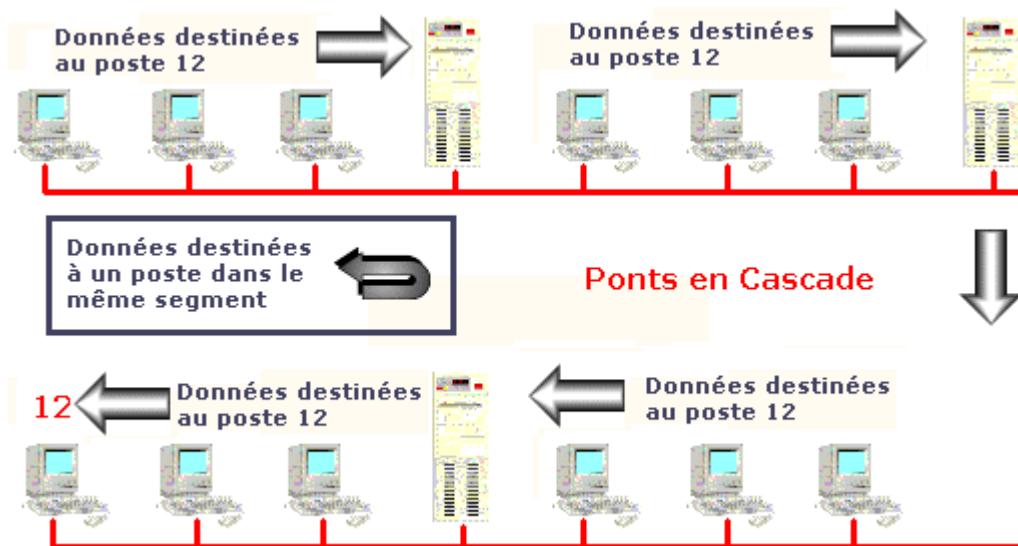
### Le Pont de Base

Le Pont de Base est utilisé très rarement pour isoler deux segments.



### Le Pont en Cascade

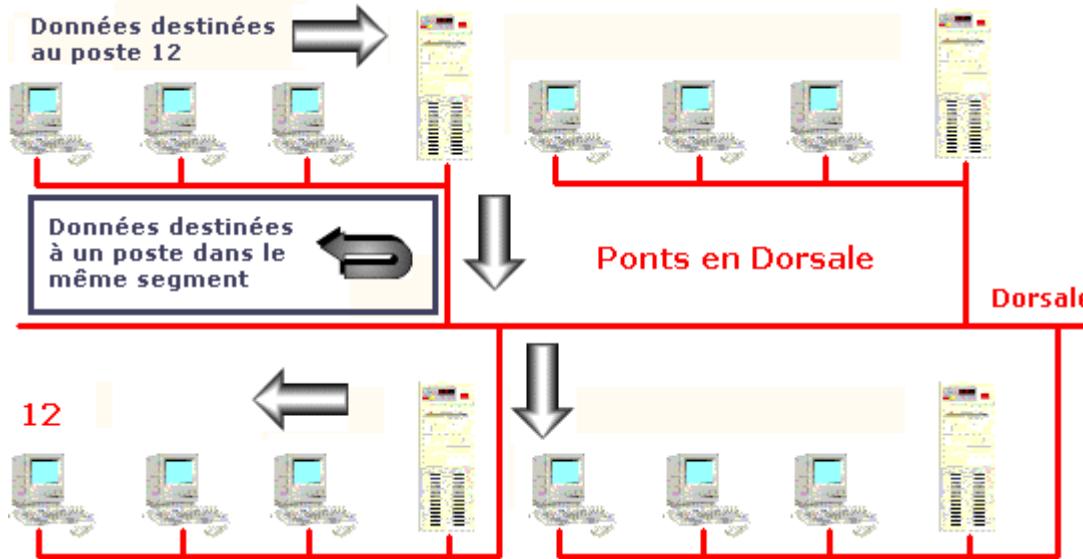
Le Pont en Cascade est à éviter car les données en provenance d'un segment doivent passer par plusieurs Ponts. Ceci a pour conséquence de ralentir la transmission des données, voire même de créer un trafic superflu en cas de rémission par le nœud



### Le Pont en Dorsale

Le Pont en Dorsale coûte plus chère que la configuration précédente car il faut un nombre de Ponts équivalent au nombre de segments + 1. Par contre

elle réduit les problèmes précédemment cités puisque les données ne transitent que par deux Ponts



## **Les Commutateurs**

Un Commutateur peut être considéré comme un Concentrateur intelligent et un Pont. Ils sont gérés souvent par des logiciels. La topologie physique d'un réseau commuté est en étoile. Par contre la topologie logique est spéciale, elle s'appelle une topologie commutée.

Lors de la communication de données entre deux nœuds, le Commutateur ouvre une connexion temporaire virtuelle en fermant les autres ports. De cette façon la bande passante totale est disponible pour cette transmission et les risques de collision sont minimisés.

Certains Commutateurs haut de gamme sont équipés d'un système anti-catastrophe qui leur permet d'isoler une partie d'un réseau en panne afin que les autres parties puissent continuer à fonctionner sans problème.

## **Les Routeurs**

Un Routeur est un Pont sophistiqué capable :

- d'assurer l'interconnexion entre des segments,
- de filtrer le trafic,
- d'isoler une partie du réseau,
- d'explorer les informations d'adressage pour trouver le chemin le plus approprié et le plus rentable pour la transmission des données.

Les Routeurs utilisent une table de routage pour stocker les informations sur :

- les adresses du réseau,
- les solutions de connexion vers d'autres réseaux,
- l'efficacité des différentes routes.

Il existe deux types de Routeur :

- **le Routeur Statique**
  - la table de routage est éditer manuellement,
  - les routes empruntées pour la transmission des données sont toujours les mêmes,
  - il n'y a pas de recherche d'efficacité.
- **le Routeur Dynamique**
  - découvre automatiquement les routes à emprunter dans un réseau.

## Les Passerelles

Ce périphérique, souvent un logiciel, sert à faire une conversion de données :

- entre deux technologies différentes ( Ethernet - Token-Ring ),
- entre deux protocoles différents,
- entre des formats de données différents.

## Comprendre le Chiffrement

## Introduction à la cryptologie

### Définitions

- **La Cryptologie**

- La science qui étudie les aspects scientifiques de ces techniques, c'est-à-dire qu'elle englobe la cryptographie et la cryptanalyse.

- **La Cryptanalyse**

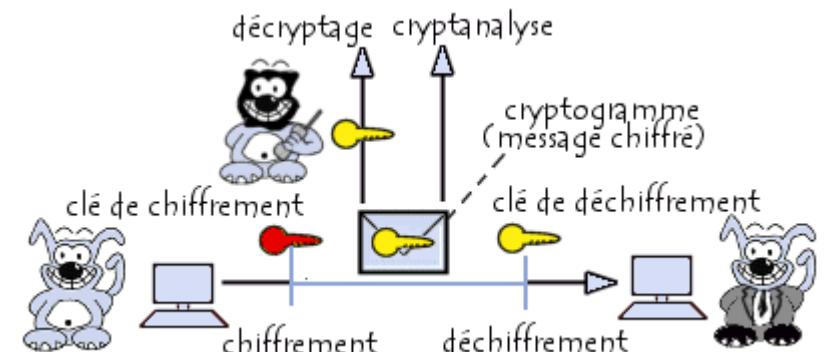
- Lorsque la clef de déchiffrement n'est pas connue de l'attaquant on parle alors de cryptanalyse ou cryptoanalyse (on entend souvent aussi le terme plus familier de cassage).

- **La Cryptographie**

- Un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre inintelligibles sans une action spécifique. Les verbes crypter et chiffrer sont utilisés.

- **Le Décryptement ou Décryptage**

- Est le fait d'essayer de déchiffrer illégitimement le message (que la clé de déchiffrement soit connue ou non de l'attaquant).



### La Cryptographie

La cryptographie apporte quatre points clefs:

- **La confidentialité**

- consiste à rendre l'information inintelligible à d'autres personnes que les acteurs de la transaction.

- L'intégrité
  - consiste à déterminer si les données n'ont pas été altérées durant la communication (de manière fortuite ou intentionnelle).
- L'authentification
  - consiste à assurer l'identité d'un utilisateur.
- La non-répudiation
  - est la garantie qu'aucun des correspondants ne pourra nier la transaction.

La cryptographie est basée sur l'arithmétique. Il s'agit, dans le cas d'un texte, de transformer les lettres qui composent le message en une succession de chiffres (sous forme de bits dans le cas de l'informatique), puis ensuite de faire des calculs sur ces chiffres pour:

- Procéder au chiffrement
  - Le résultat de cette modification (le message chiffré) est appelé cryptogramme (Ciphertext) par opposition au message initial, appelé message en clair (Plaintext)
- Procéder au déchiffrement

Le chiffrement se fait à l'aide d'une clef de chiffrement. Le déchiffrement nécessite une clef de déchiffrement.

On distingue deux types de clefs:

- Les clés symétriques:
  - des clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.
- Les clés asymétriques:
  - des clés utilisées dans le cas du chiffrement asymétrique (aussi appelé chiffrement à clé publique). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

## Le Chiffrement par Substitution

Le chiffrement par substitution consiste à remplacer dans un message une ou plusieurs entités (généralement des lettres) par une ou plusieurs autres entités. On distingue généralement plusieurs types de cryptosystèmes par substitution :

- La substitution **monoalphabétique**
  - consiste à remplacer chaque lettre du message par une autre lettre de l'alphabet
- La substitution **polyalphabétique**

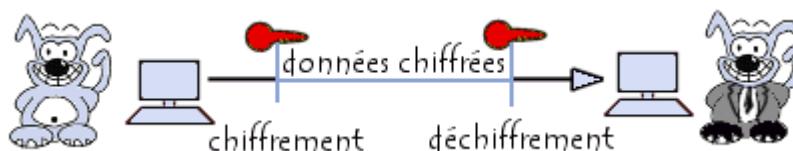
- consiste à utiliser une suite de chiffres monoalphabétique réutilisée périodiquement
- La substitution **homophonique**
  - permet de faire correspondre à chaque lettre du message en clair un ensemble possible d'autres caractères
- La substitution de **polygrammes**
  - consiste à substituer un groupe de caractères (polygramme) dans le message par un autre groupe de caractères

## Algorithmes à clé secrète

### Le Chiffrement Symétrique

Ce système est aussi appelé le système à **Clef Secrète** ou à **clef privée**.

Ce système consiste à effectuer une opération de chiffrement par algorithme mais comporte un inconvénient, à savoir qu'il nécessite un canal sécurisé pour la transmission de la clef de chiffrement/déchiffrement.



**Important** - Le système de Méthode du Masque Jetable (One Time Pad) fût mis au point dans les années 1920. Il utilisait une clef générée aléatoirement à usage unique.

Les algorithmes de chiffrement symétrique couramment utilisés en informatique sont:

- **Data Encryption Standard** (DES),
- **Triple DES** (3DES),
- **RC2**,
- **Blowfish**,
- **International Data Encryption Algorithm** (IDEA),

- Advanced Encryption Standard (AES).

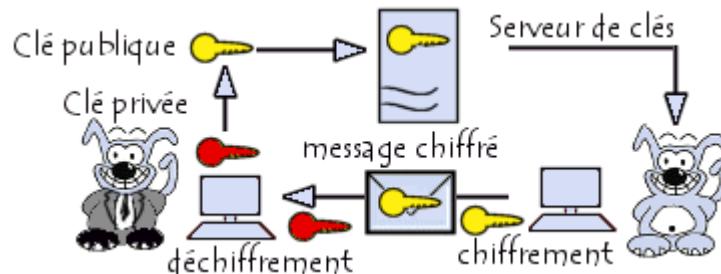
## Algorithmes à clef publique

### Le Chiffrement Asymétrique

Ce système est aussi appelé **Système à Clef Publique**.

Ce système consiste à avoir deux clefs appelées des **bi-clefs**:

- Une clef **publique** pour le chiffrement
- Une clef **secrète** ou **privée** pour le déchiffrement



- L'utilisateur A (celui qui déchiffre) choisit une clef privée.
- A partir de cette clef il génère plusieurs clefs publiques grâce à un algorithme.
- L'utilisateur B (celui qui chiffre) choisit une des clefs publiques à travers un canal non-sécurisé pour chiffrer les données à l'attention de l'utilisateur A.

Ce système est basé sur ce que l'on appelle une **fonction à trappe à sens unique** ou **one-way trap door**.

Il existe toutefois un problème – s'assurer que la clef publique récupérée est bien celle qui correspond au destinataire !

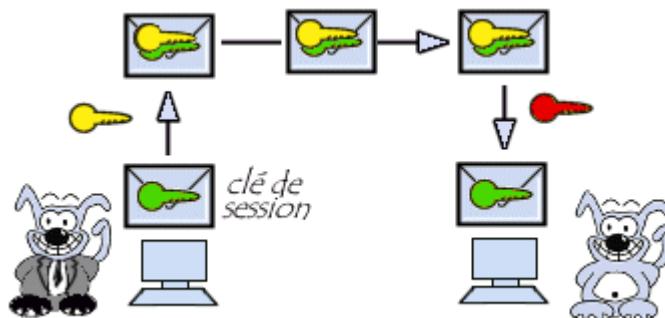
Les algorithmes de chiffrement asymétrique couramment utilisés en informatique sont:

- Digital Signature Algorithm (DSA)

- **Rivest, Shamir, Adleman** (RSA)

## La Clef de Session

Ce système est un compromis entre le système symétrique et le système asymétrique. Il permet l'envoie de données chiffrées à l'aide d'un algorithme de chiffrement symétrique par un canal non-sécurisé et a été mis au point pour palier au problème de lenteur de déchiffrement du système asymétrique.

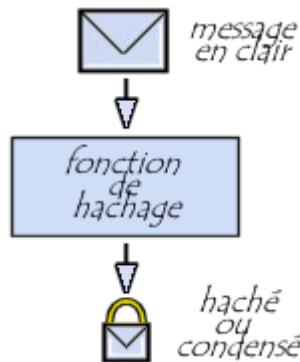


Ce système fonctionne de la façon suivante :

- L'utilisateur A chiffre une clef privée générée aléatoirement, appelée une « clef de session », en utilisant une des clefs publiques de l'utilisateur B.
- L'utilisateur A chiffre les données avec la clef de session.
- L'utilisateur B déchiffre la clef de session en utilisant sa propre clef privée.
- L'utilisateur B déchiffre les données en utilisant la clef de session.

## Fonctions de Hachage

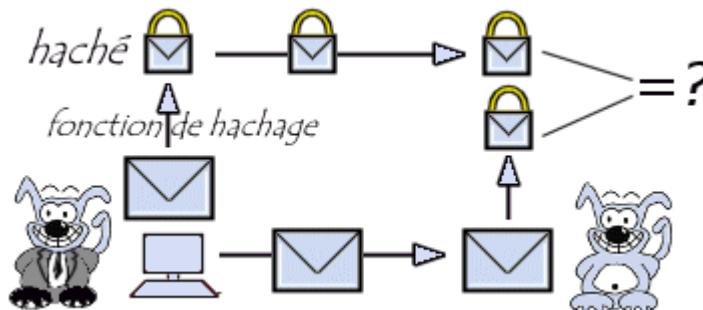
La fonction de **hachage**, aussi appelée une fonction de **condensation**, est à **sens unique** (one way function). Il « condense » un message en clair et produit un haché unique.



Les deux algorithmes de hachage utilisés sont:

- **Message Digest 5** (MD5)
- **Secure Hash Algorithm** (SHA)

Lors de son envoi, le message est accompagné de son haché et il est donc possible de garantir son intégrité:



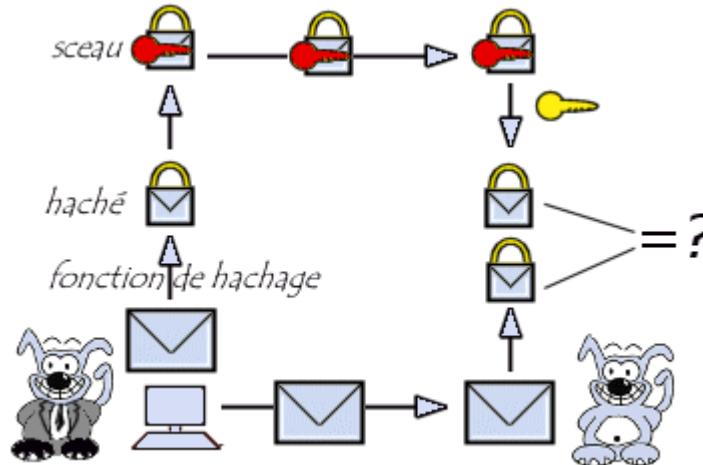
- A la réception du message, le destinataire ou l'utilisateur B calcule le haché du message reçu et le compare avec le haché accompagnant le document.
- Si le message ou le haché a été falsifié durant la communication, les deux empreintes ne correspondront pas.



**Important** - Ce système permet de vérifier que l'empreinte correspond bien au message reçu, mais ne permet pas de prouver que le message a bien été envoyé par l'utilisateur A.

## Signature Numérique

Pour garantir l'authentification du message l'utilisateur A va chiffrer ou **signer** le haché à l'aide de sa clé privée. Le haché signé est appelé un **sceau**.



- L'utilisateur A envoie le sceau au destinataire.
- A la réception du message L'utilisateur B déchiffre le sceau avec la clé publique de l'utilisateur A.
- Il compare le haché obtenu au haché reçu en pièce jointe.

Ce mécanisme de création de sceau est appelé **scellement**.

Ce mécanisme est identique au procédé utilisé par SSH lors d'une connexion

## Utilisation de GnuPG

### Présentation

**GNU Privacy Guard** permet aux utilisateurs de transférer des messages chiffrés et/ou signés.

## Installation

Sous RHEL 9, le paquet gnupg est installé par défaut :

```
[root@centos8 ~]# whereis gpg
gpg: /usr/bin/gpg /usr/share/man/man1/gpg.1.gz
```

## Configuration

Pour initialiser GnuPG, saisissez la commande suivante :

```
[root@centos8 ~]# gpg
gpg: directory '/root/.gnupg' created
gpg: keybox '/root/.gnupg/pubring.kbx' created
gpg: WARNING: no command supplied. Trying to guess what you mean ...
gpg: Go ahead and type your message ...
^C
gpg: signal Interrupt caught ... exiting
```

Pour générer les clefs, saisissez la commande suivante :

```
[root@centos8 ~]# gpg --full-generate-key
gpg (GnuPG) 2.2.20; Copyright (C) 2020 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Please select what kind of key you want:

- (1) RSA and RSA (default)
- (2) DSA and Elgamal
- (3) DSA (sign only)
- (4) RSA (sign only)
- (14) Existing key from card

```
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y
```

GnuPG needs to construct a user ID to identify your key.

```
Real name: I2TCH
Email address: infos@i2tch.co.uk
Comment: Test Key
You selected this USER-ID:
    "I2TCH (Test Key) <infos@i2tch.co.uk>"
```

```
Change (N)ame, (C)omment, (E)mail or (0)key/(Q)uit? 0
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
```

```
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
```

```
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 8B4DEC5CC2B2AC5A marked as ultimately trusted
gpg: directory '/root/.gnupg/openpgp-revocs.d' created
```

```
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/9666229B8B43D80C1832BE0D8B4DEC5CC2B2AC5A.rev'
public and secret key created and signed.

pub    rsa2048 2021-08-24 [SC]
      9666229B8B43D80C1832BE0D8B4DEC5CC2B2AC5A
uid          I2TCH (Test Key) <infos@i2tch.co.uk>
sub    rsa2048 2021-08-24 [E]
```



**Important** - Lorsque le système vous la demande, entrez la passphrase **fenestros**.

La liste de clefs peut être visualisée avec la commande suivante :

```
[root@centos8 ~]# gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
/root/.gnupg/pubring.kbx
-----
pub    rsa2048 2021-08-24 [SC]
      9666229B8B43D80C1832BE0D8B4DEC5CC2B2AC5A
uid          [ultimate] I2TCH (Test Key) <infos@i2tch.co.uk>
sub    rsa2048 2021-08-24 [E]
```



**Important** - Pour importer la clef d'un correspondant dans sa trousse de clefs il convient d'utiliser la commande suivante :

```
# gpg --import la-clef.asc
```

Pour exporter sa clef publique, il convient d'utiliser la commande suivante :

```
[root@centos8 ~]# gpg --export --armor I2TCH > ~/I2TCH.asc
[root@centos8 ~]# cat I2TCH.asc
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBGE1DSgBCACih8Jfs1n1SPiK/wGCyz2WSljsiXdXlnSHaklznxNldpY4Xrj
TPl145L95XJkHsMf++74MVMdGBn1TnG6m+J1iXkV2EbZzxw9rExA5u9W6rtzWIzP
a/90kuQNAfc/sCUoAM10Mq0vpiuc+vSHoJNuqdh4Vv1K3wSg+yQKBXacStZ/7ZS3
0PFXXFCjP6IW4a7h761EcyCXPWhuDfc7qXqLiRjNJS9xKWj0/Hd/0+UYi20XgGB8
VnjMoHodvNvmmsLCvBM8bsxUxT4izFKRHk4xM2AaQurmiU9i1J8n0C51a2Iin0tD
QT1WCryY1pnnNz014BY8VjN2eFWIFh9R9UZhABEBAAG0JEkyVENIIChUZXN0IEtl
eSkgPGluZm9zQGkydGNoLmNvLnVrPokBTgQTAQgA0BYhBJZmIpuLQ9gMGDK+DYtN
7FzCsqxaBQJhJQ0oAhsDBQsJCAcCBhUKCQgLAgQWAgnMBAh4BAheAAAoJEItN7FzC
sqxaFAKh/1ZQrtW6oNsATiG0i+X6obmWfMcRaKZiGcT5TNYdjEvXzDM/ND43nVzy
wBHJR6jZ45M4e+0eQAe01VrqBJGirrgZD0g0m8gXdXr0mygAFmUwQ6E+qYlawx7j
29p2al54zpaarSy2r/y5+hD0KV/0QxzB9xUsm0qhqmFryh+hBBvJXqNVdBH0lk+j
ENK/8BvD5FtjgU6r3pvICWiA+hwSQ2bCT+l2083twP5o19oRE3dTd+pX5/RI5Kgj
+YuD6jtVzCnA2hbjCJ4xVEREBubg/1f9D4IgnZp5QTaznpH6US2rZ1Xhz2P6Jo95
61kuoR4K4H7zvdyE0gbtZf3iDfrAc/i5AQ0EYSUNKAEIALidAGF/Ev18YfokQy5z
Xssxj2UuKRYwR06xr731aBaYKg0ym0/56Aj944WhWmJ0/RyIMpRz51p/yFLtHy1H
nWg0a3WnwGssQbL4UErEe1wUrNb3hLsvFXYDehZTwcr2adfl94Yv4ya0a9vYmb5p
Qu5tAoDQ1PUqZYsR83IjIQinF2ZgQh6+cK+MfojtwarmwhHJnYAhb0ux3WB0FVy
h6SbGxA4SpS/ANqpgR/TPFlXzXI1vVFN9x9QMhMNGjy01oIs8dcYLYoixb970shx
9IucE6Yw7SBfVlJ5ezI+Q+CNEzCJgJ/kUXNST/QWdq/h7lSE2CNhrcYAo0dEAab
pNUAEQEAAyKBNgQYAQgAIBYhBJZmIpuLQ9gMGDK+DYtN7FzCsqxaBQJhJQ0oAhsM
AAoJEItN7FzCsqxadFgH/R3ncPLtfj1RE0bZM6MUbutnQxq4RbBp9JrbqYhFy97o
lWbhMrca8Ts9pCZE3/kFbsNhg3uo7rbECYMvmCJ2Gi8RtM45SAyzezYyR45fa2W
825P+DaUdZ4ahX1jzaNEWgzMjKRt2P84ih1St7oW90c0T/04kCYhmsGfLZPch9+R
W+S8kIoiBJ8ucL5KNy9TA0TTvk4fC7w9plovpU9fJRs7CMg0kKEnTrgkH06bVK65
+4aNWr0LPPNzJaalBMLAghbzcmzRVwsB79AuKciUP/6ZTjyEGXtH/cF5Xxup5qHT
WEhhheTEBxVhlpK40Gs0B6TMSkBGq8LjQ98V3hghYa4=
=0TAN
-----END PGP PUBLIC KEY BLOCK-----
```

Cette clef peut ensuite être jointe à des messages électroniques ou bien être déposée sur un serveur de clefs tel que <http://www.keyserver.net>.

## Signer un message

Créez maintenant un message à signer :

```
[root@centos8 ~]# vi ~/message.txt
[root@centos8 ~]# cat ~/message.txt
This is a test message for gpg
```

Pour signer ce message en format binaire, il convient d'utiliser la commande suivante :

```
[root@centos8 ~]# gpg --default-key I2TCH --detach-sign message.txt
gpg: using "I2TCH" as default secret key for signing
[root@centos8 ~]# ls -l | grep message
-rw-r--r--. 1 root root 31 Aug 24 11:22 message.txt
-rw-r--r--. 1 root root 329 Aug 24 11:23 message.txt.sig
[root@centos8 ~]# cat message.txt.sig
0!f"C
M\^Za%infos@i2tch.co.uk
M\^ZT2oh@<E=n)\jED$kFvA`@7L/4XY0?49U*cje?sh
-p&Za2i?qUuQ懃
9AB|RA?Rk#b2V65mt"vC,:n
/H4&
krZ
a+ 6%60%<z+(qsv[root@centos8 ~]#
```

Pour signer ce message en format ascii, il convient d'utiliser la commande suivante :

```
[root@centos8 ~]# gpg --default-key I2TCH --armor --detach-sign message.txt
gpg: using "I2TCH" as default secret key for signing
[root@centos8 ~]# ls -l | grep message
-rw-r--r--. 1 root root 31 Aug 24 11:22 message.txt
```

```
-rw-r--r--. 1 root root 512 Aug 24 11:24 message.txt.asc
-rw-r--r--. 1 root root 329 Aug 24 11:23 message.txt.sig
[root@centos8 ~]# cat message.txt.asc
-----BEGIN PGP SIGNATURE-----

iQFGBAAwFiEElmYim4tD2AwYMr4Ni03sXMKyrFoFAmElDywSHGluZm9zQGky
dGN0LmNvLnVrAAoJEItN7FzCsqxac1YIAIoAPQ8x2G60HW8yhJKIJxCLrM+gvKz
GsTB/l+vPDEP6fToBnvMkvQwJqqQ7C0m7WkE4M2VWte6RxcpnUVcdwSlkpTKT4ww
Dbwlt7kgwX0MNPr4q0QfAG8azJB40UCRd9aq3nwstdZwmLiQ48zraR/h50W0FN/H
0muyB4khwk2lonE/z7T09BNb8kMajK0CC+ZTSb2e0Hb4U2C1jfzUybfR2v2+ApmC
Dmj4vu2jM5YnElP5Kbz4me/JY5zZbYIFhTb8TMq7kVIuibaB4keERVdd+fk0FY1Z
WFggEw1tSuoc3rZ0y1c0Rj59HoZ9QxaKX8n+wq5+A4k8slt6WzuAu8=
=//z2
-----END PGP SIGNATURE-----
```

Pour vérifier la signature d'un message signé en mode ascii, il convient d'utiliser la commande :

```
[root@centos8 ~]# gpg --verify message.txt.asc
gpg: assuming signed data in 'message.txt'
gpg: Signature made Tue 24 Aug 2021 11:24:28 EDT
gpg:           using RSA key 9666229B8B43D80C1832BE0D8B4DEC5CC2B2AC5A
gpg:           issuer "infos@i2tch.co.uk"
gpg: Good signature from "I2TCH (Test Key) <infos@i2tch.co.uk>" [ultimate]
```

Pour vérifier la signature d'un message signé en mode ascii et produit en dehors du message lui-même, il convient d'utiliser la commande :

```
[root@centos8 ~]# gpg --verify message.txt.asc message.txt
gpg: Signature made Tue 24 Aug 2021 11:24:28 EDT
gpg:           using RSA key 9666229B8B43D80C1832BE0D8B4DEC5CC2B2AC5A
gpg:           issuer "infos@i2tch.co.uk"
gpg: Good signature from "I2TCH (Test Key) <infos@i2tch.co.uk>" [ultimate]
```

Pour signer ce message **dans le message lui-même** en format ascii, il convient d'utiliser la commande suivante :

```
[root@centos8 ~]# gpg --default-key I2TCH --clearsign message.txt
gpg: using "I2TCH" as default secret key for signing
File 'message.txt.asc' exists. Overwrite? (y/N) y
[root@centos8 ~]# ls -l | grep message
-rw-r--r--. 1 root root 31 Aug 24 11:22 message.txt
-rw-r--r--. 1 root root 592 Aug 24 11:28 message.txt.asc
-rw-r--r--. 1 root root 329 Aug 24 11:23 message.txt.sig
[root@centos8 ~]# cat message.txt.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

This is a test message for gpg
-----BEGIN PGP SIGNATURE-----

iQFGBAEBCAAwFiEElmYim4tD2AwYMr4Ni03sXMKyrFoFAmE1EBMSHGlzZm9zQGky
dGNoLmNvLnVrAAoJEItN7FzCsqxaQa0H+gLxI8PTEJtb96q+PmhlsQq2PkITRDFB
bC5vW8CQzXUNA08aqkBE0gA10vX9gJG0Q/aJ07fPrQFWP9g7IYPax/GvmgHCmS7B
Hc5uUL0awGvuIctflk7xCmhgtaFndwCUN685xCPD0dhUMs0rX9Zqj8pKhbwh4Xpz
Q7vY5gPJTN2aj4PL5GkXN/ZzGclFTVN9o5BQuxYnTCB694WzZepf48dMPaIdlDxJ
l2yHf/jZGt2ZE2hoVllvjMN81LhjaqMxIoSTLwUAn+WBrwNreQdERxtQv0waIA7
NNFzGPdi0HGdJhjYJ/v4eFbi5X4gvHVVazz0pY5p48yVgCRAwZHJh/0=
=C30Q
-----END PGP SIGNATURE-----
```

## Chiffrer un message

Pour chiffrer un message, il faut disposer de la clef publique du destinataire du message. Ce dernier utilisera ensuite sa clef privée pour déchiffrer le message. Il convient de préciser le destinataire du message, ou plus précisément la clef publique à utiliser, lors d'un chiffrement :

```
gpg --recipient <destinataire> --encrypt <message>
```

- <destinataire> représente toute information permettant de distinguer sans ambiguïté une clef publique dans votre trousseau. Cette information peut-être le nom ou l'adresse email associé à la clef publique que vous voulez utiliser,

- <message> représente le message à chiffrer.

Par exemple pour chiffrer un message en mode binaire, il convient de saisir la commande suivante :

```
[root@centos8 ~]# gpg --recipient I2TCH --encrypt message.txt
[root@centos8 ~]# ls -l | grep message
-rw-r--r--. 1 root root 31 Aug 24 11:22 message.txt
-rw-r--r--. 1 root root 592 Aug 24 11:28 message.txt.asc
-rw-r--r--. 1 root root 367 Aug 24 11:30 message.txt.gpg
-rw-r--r--. 1 root root 329 Aug 24 11:23 message.txt.sig
[root@centos8 ~]# cat message.txt.gpg
```

EeJ ¶

```
pqa=w_wZI)0,G@"s"+i:(AVG;@GX][]ba9hh%7
Wg7X
o#U>gHEre8K\R*4u0n@"{SIlg6gy].Z{t0'p@k{%I~}pu0-
#f t^S)[ )Xq=#94t;fM C|UVo ,H|+.!4:DmZl0]bI{H[root@centos8 ~]#
```

Et pour chiffrer un message en mode ascii, il convient de saisir la commande suivante :

```
[root@centos8 ~]# gpg --recipient I2TCH --armor --encrypt message.txt
File 'message.txt.asc' exists. Overwrite? (y/N) y
[root@centos8 ~]# ls -l | grep message
-rw-r--r--. 1 root root 31 Aug 24 11:22 message.txt
-rw-r--r--. 1 root root 561 Aug 24 11:32 message.txt.asc
-rw-r--r--. 1 root root 367 Aug 24 11:30 message.txt.gpg
-rw-r--r--. 1 root root 329 Aug 24 11:23 message.txt.sig
[root@centos8 ~]# cat message.txt.asc
-----BEGIN PGP MESSAGE-----
```

```
hQEMA0XsZUog1b4LAQf7BgGL8LMcMbLdD4nS0wc45FLNyj9MXkr0ru01jBRb3UP/
MW6VxWekLrW0XRbFo/dS1Y/KIAYiZ9kDVSYwbbrQx0ql/F4sWBagWA0s/gzeWt6
MrKuOK6pgPdg057AcIm0eUjPL42RHh6enGRdud+GWiZNQKAvPiCNikfhJUza+o1Z
```

```
GyAcq5RMSuoh0p2weai5CwcVqZdrTvKzjkoUrMCwnMxGKjdbNRC3+DKEI9B4L3j
7Dno9DseQcebD3NYEICSt2oJr+xazejiLj4X8nerBrCqV7nK9v7mvxTKCIL5i0BR
duBPFvgJuSVnSJZ+XzBeEQ8q24L3FLV9B5yJnF+e8tJeASweIXfqWaeWN0bfAHC3
dkMtvNUNs6jkmFUGd0NYosNlHW9jFWllpe2Q5Ra13kdZob3oleevU2iGBAx0Gi0Z
yEB3HjqYFKxFj+lCj4KP59055sEpePgAo2qhPhfeMw==
=UDxQ
-----END PGP MESSAGE-----
```

Pour décrypter un message il convient d'utiliser la commande suivante :

```
[root@centos8 ~]# gpg --decrypt message.txt.asc
gpg: encrypted with 2048-bit RSA key, ID 45EC654A20D5BE0B, created 2021-08-24
    "I2TCH (Test Key) <infos@i2tch.co.uk>"
This is a test message for gpg
```

## PKI

On appelle  **PKI** (Public Key Infrastructure, ou en français **infrastructure à clé publique (ICP)**, parfois **infrastructure de gestion de clés (IGC)**) l'ensemble des solutions techniques basées sur la cryptographie à clé publique.

Les cryptosystèmes à clés publiques permettent de s'affranchir de la nécessité d'avoir recours systématiquement à un canal sécurisé pour s'échanger les clés. En revanche, la publication de la clé publique à grande échelle doit se faire en toute confiance pour assurer que :

- La clé publique est bien celle de son propriétaire ;
- Le propriétaire de la clé est digne de confiance ;
- La clé est toujours valide.

Ainsi, il est nécessaire d'associer au bi-clé (ensemble clé publique / clé privée) un certificat délivré par un **tiers de confiance** : l'infrastructure de gestion de clés.

Le tiers de confiance est une entité appelée communément autorité de certification (ou en anglais Certification authority, abrégé CA) chargée d'assurer la véracité des informations contenues dans le certificat de clé publique et de sa validité.

Pour ce faire, l'autorité signe le certificat de clé publique à l'aide de sa propre clé en utilisant le principe de signature numérique.

Le rôle de l'infrastructure de clés publiques est multiple et couvre notamment les champs suivants :

- enregistrer des demandes de clés en vérifiant l'identité des demandeurs ;
- générer les paires de clés (clé privée / clé publique) ;
- garantir la confidentialité des clés privées correspondant aux clés publiques ;
- certifier l'association entre chaque utilisateurs et sa clé publique ;
- révoquer des clés (en cas de perte par son propriétaire, d'expiration de sa date de validité ou de compromission).

Une infrastructure à clé publique est en règle générale composée de trois entités distinctes :

- L'autorité d'enregistrement (AE ou RA pour Recording authority), chargée des formalités administratives telles que la vérification de l'identité des demandeurs, le suivi et la gestion des demandes, etc.) ;
- L'autorité de certification (AC ou CA pour Certification Authority), chargée des tâches techniques de création de certificats. L'autorité de certification est ainsi chargée de la signature des demandes de certificat (CSR pour Certificate Signing Request, parfois appelées PKCS#10, nom du format correspondant). L'autorité de certification a également pour mission la signature des listes de révocations (CRL pour Certificate Revocation List) ;
- L'Autorité de dépôt (Repository) dont la mission est de conserver en sécurité les certificats.

## Certificats X509

Pour palier aux problèmes liés à des clefs publiques piratées, un système de certificats a été mis en place.

Le certificat permet d'associer la clé publique à une entité ou une personne. Les certificats sont délivrés par des Organismes de Certification.

Les certificats sont des fichiers divisés en deux parties :

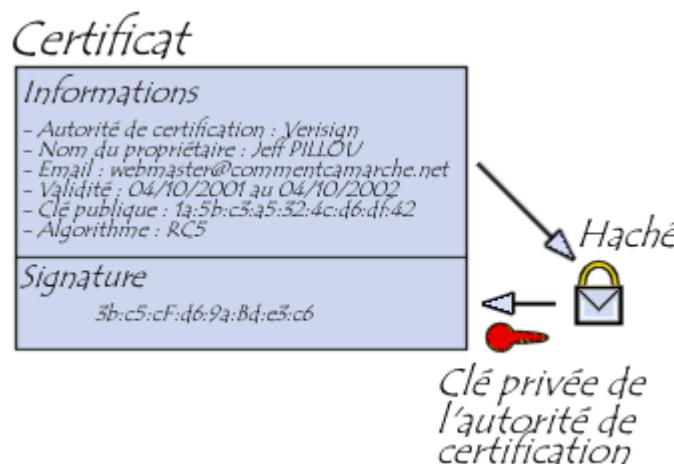
- La partie contenant les informations
- La partie contenant la signature de l'autorité de certification

La structure des certificats est normalisée par le standard [X.509](#) de l'[Union internationale des télécommunications](#).

Elle contient :

- Le nom de l'autorité de certification
- Le nom du propriétaire du certificat
- La date de validité du certificat
- L'algorithme de chiffrement utilisé
- La clé publique du propriétaire

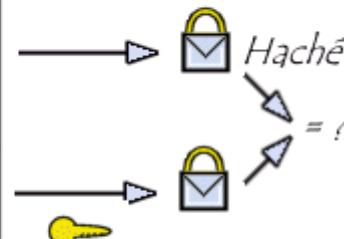
Le Certificat est signé par l'autorité de certification:



La vérification se passe ainsi:

## Certificat

| Informations   |
|--|
| - Autorité de certification : Verisign<br>- Nom du propriétaire : Jeff PILLOU<br>- Email : webmaster@commentcamarche.net<br>- Validité : 04/10/2001 au 04/10/2002<br>- Clé publique : 1a:5b:c5:a5:32:4c:d6:df:42<br>- Algorithme : RC5 |
| Signature  |
| 3b:c5:cF:d6:9a:8d:c3:c6  |



Déchiffrement à l'aide  
de la clé publique de  
l'autorité de certification

## Comprendre IPv4

### En-tête TCP

L'en-tête TCP est codée sur 4 octets soit 32 bits :

| 1er octet   | 2ème octet | 3ème octet            | 4 ème octet |
|-------------|------------|-----------------------|-------------|
| Port source |            | Port destination      |             |
|             |            | Numéro de séquence    |             |
|             |            | Numéro d'acquittement |             |
| Offset      | Flags      |                       | Fenêtre     |
| Checksum    |            | Pointeur Urgent       |             |
| Options     |            | Padding               |             |
|             |            | Données               |             |

Vous noterez que les numéros de ports sont codés sur 16 bits. Cette information nous permet de calculer le nombres de ports maximum en IPv4, soit  $2^{16}$  ports ou 65 535.

L'**Offset** contient la taille de l'en-tête.

Les **Flags** sont :

- URG - Si la valeur est 1 le pointeur urgent est utilisé. Le numéro de séquence et le pointeur urgent indique un octet spécifique.
- ACK - Si la valeur est 1, le paquet est un accusé de réception
- PSH - Si la valeur est 1, les données sont immédiatement présentées à l'application
- RST - Si la valeur est 1, la communication comporte un problème et la connexion est réinitialisée
- SYN - Si la valeur est 1, le paquet est un paquet de synchronisation
- FIN - Si la valeur est 1, le paquet indique la fin de la connexion

La **Fenêtre** est codée sur 16 bits. La Fenêtre est une donnée liée au fonctionnement d'expédition de données appelé le **sliding window** ou la **fenêtre glissante**. Puisque il serait impossible, pour des raisons de performance, d'attendre l'accusé de réception de chaque paquet envoyé, l'expéditeur envoie des paquets par groupe. La taille de cette groupe s'appelle la Fenêtre. Dans le cas d'un problème de réception d'une partie de la Fenêtre, toute la Fenêtre est ré-expédiée.

Le **Checksum** est une façon de calculer si le paquet est complet.

Le **Padding** est un champ pouvant être rempli de valeurs nulles de façon à ce que la taille de l'en-tête soit un multiple de 32

## En-tête UDP

L'en-tête UDP est codée sur 4 octets soit 32 bits :

| 1er octet   | 2ème octet | 3ème octet       | 4 ème octet |
|-------------|------------|------------------|-------------|
| Port source |            | Port destination |             |
| longueur    |            | Checksum         |             |
| Données     |            |                  |             |

L'en-tête UDP a une longueur de 8 octets.

## Fragmentation et Ré-encapsulation

La taille limite d'un paquet TCP, l'en-tête comprise, ne peut pas dépasser **65 535 octets**. Cependant chaque réseau est qualifié par son MTU ( Maximum Tranfer Unit ). Cette valeur est la taille maximum d'un paquet autorisée. L'unité est en **octets**. Pour un réseau Ethernet sa valeur est de 1 500. Quand un paquet doit être expédié sur un réseau ayant un MTU inférieur à sa propre taille, le paquet doit être **fractionné**. A la sortie du réseau, le paquet est reconstitué. Cette reconstitution s'appelle **ré-encapsulation**.

## Adressage

L'adressage IP requiert que chaque périphérique sur le réseau possède une adresse IP unique de 4 octets, soit 32 bits au format XXX.XXX.XXX.XXX. De cette façon le nombre total d'adresses est de  $2^{32} = 4.3$  Milliards.

Les adresses IP sont divisées en 5 classes, de A à E. Les 4 octets des classes A à C sont divisés en deux, une partie qui s'appelle le **Net ID** qui identifie le réseau et une partie qui s'appelle le **Host ID** qui identifie le hôte :

|   | <b>1er octet</b> | <b>2ème octet</b> | <b>3ème octet</b> | <b>4 ème octet</b> |
|---|------------------|-------------------|-------------------|--------------------|
| A | Net ID           |                   |                   | Host ID            |
| B |                  | Net ID            |                   | Host ID            |
| C |                  |                   | Net ID            | Host ID            |
| D |                  |                   |                   | Multicast          |
| E |                  |                   |                   | Réservé            |

L'attribution d'une classe dépend du nombre de hôtes à connecter. Chaque classe est identifiée par un **Class ID** composé de 1 à 3 bits :

| <b>Classe</b> | <b>Bits ID Classe</b> | <b>Valeur ID Classe</b> | <b>Bits ID Réseau</b> | <b>Nb. de Réseaux</b> | <b>Bits ID hôtes</b> | <b>Nb. d'adresses</b> | <b>Octet de Départ</b> |
|---------------|-----------------------|-------------------------|-----------------------|-----------------------|----------------------|-----------------------|------------------------|
| A             | 1                     | 0                       | 7                     | $2^7=128$             | 24                   | $2^{24}=16\ 777\ 216$ | 1 - 126                |
| B             | 2                     | 10                      | 14                    | $2^{14}=16\ 834$      | 16                   | $2^{16}=65\ 535$      | 128 - 191              |
| C             | 3                     | 110                     | 21                    | $2^{21}=2\ 097\ 152$  | 8                    | $2^8=256$             | 192 - 223              |

Dans chaque classe, certaines adresses sont réservées pour un usage privé :

| Classe | IP de Départ | IP de Fin       |
|--------|--------------|-----------------|
| A      | 10.0.0.0     | 10.255.255.255  |
| B      | 172.16.0.0   | 172.31.255.255  |
| C      | 192.168.0.0  | 192.168.255.255 |

Il existe des adresses particulières ne pouvant pas être utilisées pour identifier un hôte :

| Adresse Particulière          | Description                                  |
|-------------------------------|--|
| 169.254.0.0 à 169.254.255.255 | Automatic Private IP Addressing de Microsoft |
| Hôte du réseau courant        | Tous les bits du Net ID sont à 0             |
| Adresse de réseau             | Tous les bits du Host ID sont à 0            |
| Adresse de diffusion          | Tous les bits du Host ID sont à 1            |

L'adresse de réseau identifie le **segment** du réseau entier tandis que l'adresse de diffusion identifie tous les hôtes sur le segment de réseau.

Afin de mieux comprendre l'adresse de réseau et l'adresse de diffusion, prenons le cas de l'adresse 192.168.10.1 en classe C :

|                                  | 1er octet | 2ème octet | 3ème octet | 4 ème octet     |
|----------------------------------|-----------|------------|------------|-----------------|
|                                  | Net ID    |            |            | Host ID         |
| Adresse IP                       | 192       | 168        | 10         | 1               |
| Binaire                          | 11000000  | 10101000   | 000001010  | 00000001        |
| Calcul de l'adresse de réseau    |           |            |            |                 |
| Binaire                          | 11000000  | 10101000   | 000001010  | <b>00000000</b> |
| Adresse réseau                   | 192       | 168        | 10         | 0               |
| Calcul de l'adresse de diffusion |           |            |            |                 |
| Binaire                          | 11000000  | 10101000   | 000001010  | <b>11111111</b> |
| Adresse de diffusion             | 192       | 168        | 10         | 255             |

## Masques de sous-réseaux

Tout comme l'adresse IP, le masque de sous-réseau compte 4 octets ou 32 bits. Les masques de sous-réseaux permettent d'identifier le Net ID et le Host ID :

| Classe | Masque        | Notation CIDR |
|--------|---------------|---------------|
| A      | 255.0.0.0     | /8            |
| B      | 255.255.0.0   | /16           |
| C      | 255.255.255.0 | /24           |

Le terme **CIDR** veut dire **Classless InterDomain Routing**. Le terme Notation CIDR correspond au nombre de bits d'une valeur de 1 dans le masque de sous-réseau.

Quand un hôte souhaite émettre il procède d'abord à l'identification de sa propre adresse réseau par un calcul AND (ET) appliqué à sa propre adresse et son masque de sous-réseau qui stipule :

- $1 \times 1 = 1$
- $0 \times 1 = 0$
- $1 \times 0 = 0$
- $0 \times 0 = 0$

Prenons le cas de l'adresse IP 192.168.10.1 ayant un masque de 255.255.255.0 :

|                       | 1er octet | 2ème octet | 3ème octet | 4 ème octet |
|-----------------------|-----------|------------|------------|-------------|
| Adresse IP            | 192       | 168        | 10         | 1           |
| Binaire               | 11000000  | 10101000   | 00001010   | 00000001    |
| Masque de sous-réseau |           |            |            |             |
| Binaire               | 11111111  | 11111111   | 11111111   | 00000000    |
| Calcul AND            | 11000000  | 10101000   | 00001010   | 00000000    |
| Adresse réseau        | 192       | 168        | 10         | 0           |

Cet hôte essaie de communiquer avec un hôte ayant une adresse IP de 192.168.10.10. Il procède donc au même calcul en appliquant **son propre masque de sous-réseau** à l'adresse IP de l'hôte destinataire :

|                       | 1er octet | 2ème octet | 3ème octet | 4 ème octet |
|-----------------------|-----------|------------|------------|-------------|
| Adresse IP            | 192       | 168        | 10         | 10          |
| Binaire               | 11000000  | 10101000   | 00001010   | 00001010    |
| Masque de sous-réseau |           |            |            |             |

|                | <b>1er octet</b> | <b>2ème octet</b> | <b>3ème octet</b> | <b>4 ème octet</b> |
|----------------|------------------|-------------------|-------------------|--------------------|
| Binaire        | 11111111         | 11111111          | 11111111          | 00000000           |
| Calcul AND     | 11000000         | 10101000          | 00001010          | 00000000           |
| Adresse réseau | 192              | 168               | 10                | 0                  |

Puisque l'adresse réseau est identique dans les deux cas, l'hôte émetteur présume que l'hôte de destination se trouve sur son réseau et envoie les paquets directement sur le réseau sans s'adresser à sa passerelle par défaut.

L'hôte émetteur essaie maintenant de communiquer avec un hôte ayant une adresse IP de 192.168.2.1. Il procède donc au même calcul en appliquant **son propre masque de sous-réseau** à l'adresse IP de l'hôte destinataire :

|                       | <b>1er octet</b> | <b>2ème octet</b> | <b>3ème octet</b> | <b>4 ème octet</b> |
|-----------------------|------------------|-------------------|-------------------|--------------------|
| Adresse IP            | 192              | 168               | 2                 | 1                  |
| Binaire               | 11000000         | 10101000          | 00000010          | 00000001           |
| Masque de sous-réseau |                  |                   |                   |                    |
| Binaire               | 11111111         | 11111111          | 11111111          | 00000000           |
| Calcul AND            | 11000000         | 10101000          | 00000010          | 00000000           |
| Adresse réseau        | 192              | 168               | 2                 | 0                  |

Dans ce cas, l'hôte émetteur constate que le réseau de destination 192.168.2.0 n'est pas identique à son propre réseau 192.168.10.0. Il adresse donc les paquets à la passerelle par défaut.

## VLSM

Puisque le stock de réseaux disponibles sous IPv4 est presque épuisé, une solution a du être trouvée pour créer des sous-réseaux en attendant l'introduction de l'IPv6. Cette solution s'appelle le VLSM ou Variable Length Subnet Masks. Le VLSM exprime les masques de sous-réseaux au format CIDR.

Son principe est simple. Afin de créer des réseaux différents à partir d'une adresse réseau d'une classe donnée, il convient de réduire le nombre d'hôtes. De cette façon les bits 'libérés' du Host ID peuvent être utilisés pour identifier les sous-réseaux.

Pour illustrer ceci, prenons l'exemple d'un réseau 192.168.1.0. Sur ce réseau, nous pouvons mettre  $2^8 - 2$  soit 254 hôtes entre 192.168.1.1 au

192.168.1.254.

Supposons que nous souhaiterions diviser notre réseau en 2 sous-réseaux. Pour coder 2 sous-réseaux, il faut que l'on libère 2 bits du Host ID. Les deux bits libérés auront les valeurs binaires suivantes :

- 00
- 01
- 10
- 11

Les valeurs binaires du quatrième octet de nos adresses de sous-réseaux seront donc :

- 192.168.1.00XXXXXX
- 192.168.1.01XXXXXX
- 192.168.1.10XXXXXX
- 192.168.1.11XXXXXX

où les XXXXXX représentent les bits que nous réservons pour décrire les hôtes dans chacun des sous-réseaux.

Nous ne pouvons pas utiliser les deux sous-réseaux suivants :

- 192.168.1.00XXXXXX
- 192.168.1.11XXXXXX

car ceux-ci correspondent aux débuts de l'adresse réseau 192.168.1.0 et de l'adresse de diffusion 192.168.1.255.

Nous pouvons utiliser les deux sous-réseaux suivants :

- 192.168.1.01XXXXXX
- 192.168.1.10XXXXXX

Pour le premier sous-réseau l'adresse réseau et l'adresse de diffusion sont :

|                               |          |          |          |                  |
|-------------------------------|----------|----------|----------|------------------|
| Sous-réseau #1                | 192      | 168      | 1        | 01XXXXXX         |
| Calcul de l'adresse de réseau |          |          |          |                  |
| Binaire                       | 11000000 | 10101000 | 00000001 | 01 <b>000000</b> |

|                                  |          |          |          |                  |
|----------------------------------|----------|----------|----------|------------------|
| Adresse réseau                   | 192      | 168      | 1        | <b>64</b>        |
| Calcul de l'adresse de diffusion |          |          |          |                  |
| Binaire                          | 11000000 | 10101000 | 00000001 | 01 <b>111111</b> |
| Adresse de diffusion             | 192      | 168      | 1        | <b>127</b>       |

- L'adresse CIDR du réseau est donc 192.168.1.64/26 car le Net ID est codé sur 24+2 bits.
- Le masque de sous-réseau est donc le 11111111.11111111.11111111.11000000 ou le 255.255.255.192
- Nous pouvons avoir  $2^6-2$  soit 62 hôtes.
- La plage valide d'adresses IP est de 192.168.1.65 à 192.168.1.126

Pour le deuxième sous-réseau l'adresse réseau et l'adresse de diffusion sont :

|                                  |          |          |          |                 |
|----------------------------------|----------|----------|----------|-----------------|
| Sous-réseau #2                   | 192      | 168      | 1        | 10XXXXXX        |
| Calcul de l'adresse de réseau    |          |          |          |                 |
| Binaire                          | 11000000 | 10101000 | 00000001 | <b>10000000</b> |
| Adresse réseau                   | 192      | 168      | 1        | <b>128</b>      |
| Calcul de l'adresse de diffusion |          |          |          |                 |
| Binaire                          | 11000000 | 10101000 | 00000001 | <b>10111111</b> |
| Adresse de diffusion             | 192      | 168      | 1        | <b>191</b>      |

- L'adresse CIDR du réseau est donc 192.168.1.128/26 car le Net ID est codé sur 24+2 bits.
- Le masque de sous-réseau est donc le 11111111.11111111.11111111.11000000 ou le 255.255.255.192
- Nous pouvons avoir  $2^6-2$  soit 62 hôtes.
- La plage valide d'adresses IP est de 192.168.1.129 à 192.168.1.190

La valeur qui sépare les sous-réseaux est 64. Cette valeur comporte le nom **incrément**.

## Ports et sockets

Afin que les données arrivent aux applications que les attendent, TCP utilise des numéros de ports sur la couche transport. Le numéros de ports sont divisés en trois groupes :

- **Well Known Ports**
  - De 1 à 1023
- **Registered Ports**
  - De 1024 à 49151
- **Dynamic et/ou Private Ports**
  - De 49152 à 65535

Le couple **numéro IP:numéro de port** s'appelle un **socket**.

## /etc/services

Les ports les plus utilisés sont détaillés dans le fichier **/etc/services** :

```
[root@centos8 ~]# more /etc/services
# /etc/services:
# $Id: services,v 1.49 2017/08/18 12:43:23 ovasik Exp $
#
# Network services, Internet style
# IANA services version: last updated 2016-07-08
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, ``Assigned Numbers'' (October 1994). Not all ports
# are included, only the more common ones.
#
# The latest IANA port assignments can be gotten from
#      http://www.iana.org/assignments/port-numbers
# The Well Known Ports are those from 0 through 1023.
# The Registered Ports are those from 1024 through 49151
```

```
# The Dynamic and/or Private Ports are those from 49152 through 65535
#
# Each line describes one service, and is of the form:
#
# service-name  port/protocol  [aliases ...]  [# comment]

tcpmux      1/tcp          # TCP port service multiplexer
tcpmux      1/udp          # TCP port service multiplexer
rje         5/tcp          # Remote Job Entry
rje         5/udp          # Remote Job Entry
echo        7/tcp
echo        7/udp
discard     9/tcp          sink null
discard     9/udp          sink null
systat      11/tcp         users
systat      11/udp         users
daytime     13/tcp
daytime     13/udp
qotd        17/tcp          quote
qotd        17/udp          quote
chargen     19/tcp          ttystt source
chargen     19/udp          ttystt source
ftp-data    20/tcp
ftp-data    20/udp
# 21 is registered to ftp, but also used by fsp
ftp         21/tcp
ftp         21/udp          fsp fspd
--More-- (0%)
```

Notez que les ports sont listés par deux :

- le port TCP
- le port UDP

La liste la plus complète peut être consultée à l'adresse suivante

<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>.

Pour connaître la liste des sockets ouverts sur l'ordinateur, saisissez la commande suivante :

```
[root@centos8 ~]# netstat -an | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 0.0.0.0:111              0.0.0.0:*              LISTEN
tcp      0      0 192.168.122.1:53           0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:22               0.0.0.0:*              LISTEN
tcp      0      0 127.0.0.1:631              0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:8888              0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:5901              0.0.0.0:*              LISTEN
tcp      0      0 10.0.2.45:22              10.0.2.1:50574         ESTABLISHED
tcp     32      0 10.0.2.45:50280            8.43.85.29:443         CLOSE_WAIT
tcp     32      0 10.0.2.45:50278            8.43.85.29:443         CLOSE_WAIT
tcp      0      0 10.0.2.45:36844            44.238.3.246:443         ESTABLISHED
tcp6     0      0 :::111                  :::*                  LISTEN
tcp6     0      0 :::22                   :::*                  LISTEN
tcp6     0      0 ::1:631                 :::*                  LISTEN
tcp6     0      0 :::5901                 :::*                  LISTEN
udp      0      0 0.0.0.0:25826             0.0.0.0:*
udp      0      0 0.0.0.0:5353              0.0.0.0:*
udp      0      0 0.0.0.0:36264             0.0.0.0:*
udp      0      0 192.168.122.1:53           0.0.0.0:*
udp      0      0 0.0.0.0:67               0.0.0.0:*
udp      0      0 0.0.0.0:111              0.0.0.0:*
udp      0      0 127.0.0.1:323              0.0.0.0:*
udp6     0      0 :::5353                 :::*                  LISTEN
udp6     0      0 :::42631                :::*                  LISTEN
udp6     0      0 ::::111                 :::*                  LISTEN
udp6     0      0 ::1:323                 :::*                  LISTEN
raw6     0      0 ::::58                  :::*                  LISTEN
```

## Active UNIX domain sockets (servers and established)

| Proto    | RefCnt | Flags   | Type   | State     | I-Node | Path                                  |
|----------|--------|---------|--------|-----------|--------|---------------------------------------|
| unix     | 2      | [ ACC ] | STREAM | LISTENING | 37076  | @/tmp/.ICE-unix/1873                  |
| unix     | 2      | [ ACC ] | STREAM | LISTENING | 25881  | @irqbalance907.sock                   |
| unix     | 2      | [ ACC ] | STREAM | LISTENING | 37130  | /run/user/1000/keyring/control        |
| unix     | 2      | [ ACC ] | STREAM | LISTENING | 37132  | /run/user/1000/keyring/ssh            |
| unix     | 2      | [ ACC ] | STREAM | LISTENING | 32014  | /run/gssproxy.sock                    |
| unix     | 2      | [ ACC ] | STREAM | LISTENING | 30479  | /run/user/42/bus                      |
| unix     | 2      | [ ACC ] | STREAM | LISTENING | 30481  | /run/user/42/pulse/native             |
| unix     | 2      | [ ACC ] | STREAM | LISTENING | 5617   | @/org/kernel/linux/storage/multipathd |
| unix     | 2      | [ ACC ] | STREAM | LISTENING | 1811   | /var/run/.heim_org.h5l.kcm-socket     |
| unix     | 2      | [ ACC ] | STREAM | LISTENING | 30484  | /run/user/42/pipewire-0               |
| --More-- |        |         |        |           |        |                                       |

Pour connaître la liste des applications ayant ouvert un port sur l'ordinateur, saisissez la commande suivante :

| [root@centos8 ~]# netstat -anp   more                 |        |        |                  |                  |             |                     |
|---|--------|--------|------------------|------------------|-------------|---------------------|
| Active Internet connections (servers and established) |        |        |                  |                  |             |                     |
| Proto   | Recv-Q | Send-Q | Local Address    | Foreign Address  | State       | PID/Program name    |
| tcp   | 0      | 0      | 0.0.0.0:111      | 0.0.0.0:*        | LISTEN      | 1/systemd           |
| tcp   | 0      | 0      | 192.168.122.1:53 | 0.0.0.0:*        | LISTEN      | 1859/dnsmasq        |
| tcp   | 0      | 0      | 0.0.0.0:22       | 0.0.0.0:*        | LISTEN      | 1027/sshd           |
| tcp   | 0      | 0      | 127.0.0.1:631    | 0.0.0.0:*        | LISTEN      | 1030/cupsd          |
| tcp   | 0      | 0      | 0.0.0.0:8888     | 0.0.0.0:*        | LISTEN      | 804664/python2      |
| tcp   | 0      | 0      | 0.0.0.0:5901     | 0.0.0.0:*        | LISTEN      | 1797/Xvnc           |
| tcp   | 0      | 0      | 10.0.2.45:22     | 10.0.2.1:50574   | ESTABLISHED | 841191/sshd: traine |
| tcp   | 32     | 0      | 10.0.2.45:50280  | 8.43.85.29:443   | CLOSE_WAIT  | 2071/gnome-shell    |
| tcp   | 32     | 0      | 10.0.2.45:50278  | 8.43.85.29:443   | CLOSE_WAIT  | 1903/gnome-shell    |
| tcp   | 0      | 0      | 10.0.2.45:36844  | 44.238.3.246:443 | ESTABLISHED | 2903/firefox        |
| tcp6  | 0      | 0      | :::111           | :::*             | LISTEN      | 1/systemd           |
| tcp6  | 0      | 0      | :::22            | :::*             | LISTEN      | 1027/sshd           |

|      |   |                    |           |   |        |                     |
|------|---|--------------------|-----------|---|--------|---------------------|
| tcp6 | 0 | 0 ::1:631          | :::*      |   | LISTEN | 1030/cupsd          |
| tcp6 | 0 | 0 :::5901          | :::*      |   | LISTEN | 1797/Xvnc           |
| udp  | 0 | 0 0.0.0.0:25826    | 0.0.0.0:* |   |        | 804615/collectd     |
| udp  | 0 | 0 0.0.0.0:5353     | 0.0.0.0:* |   |        | 905/avahi-daemon: r |
| udp  | 0 | 0 0.0.0.0:36264    | 0.0.0.0:* |   |        | 905/avahi-daemon: r |
| udp  | 0 | 0 192.168.122.1:53 | 0.0.0.0:* |   |        | 1859/dnsmasq        |
| udp  | 0 | 0 0.0.0.0:67       | 0.0.0.0:* |   |        | 1859/dnsmasq        |
| udp  | 0 | 0 0.0.0.0:111      | 0.0.0.0:* |   |        | 1/systemd           |
| udp  | 0 | 0 127.0.0.1:323    | 0.0.0.0:* |   |        | 909/chrony          |
| udp6 | 0 | 0 :::5353          | :::*      |   |        | 905/avahi-daemon: r |
| udp6 | 0 | 0 :::42631         | :::*      |   |        | 905/avahi-daemon: r |
| udp6 | 0 | 0 :::111           | :::*      |   |        | 1/systemd           |
| udp6 | 0 | 0 ::1:323          | :::*      |   |        | 909/chrony          |
| raw6 | 0 | 0 :::58            | :::*      | 7 |        | 1002/NetworkManager |

## Active UNIX domain sockets (servers and established)

| Proto                                 | RefCnt | Flags   | Type   | State     | I-Node | PID/Program name    | Path                              |
|---------------------------------------|--------|---------|--------|-----------|--------|---------------------|-----------------------------------|
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 37076  | 1873/gnome-session- | @/tmp/.ICE-unix/1873              |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 25881  | 907/irqbalance      | @irqbalance907.sock               |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 37130  | 2063/gnome-keyring- | /run/user/1000/keyring/control    |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 37132  | 2063/gnome-keyring- | /run/user/1000/keyring/ssh        |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 32014  | 1040/gssproxy       | /run/gssproxy.sock                |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 30479  | 1439/systemd        | /run/user/42/bus                  |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 30481  | 1439/systemd        | /run/user/42/pulse/native         |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 5617   | 1/systemd           |                                   |
| @/org/kernel/linux/storage/multipathd |        |         |        |           |        |                     |                                   |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 1811   | 1/systemd           | /var/run/.heim_org.h5l.kcm-socket |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 30484  | 1439/systemd        | /run/user/42/pipewire-0           |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 1813   | 1/systemd           | /run/avahi-daemon/socket          |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 1817   | 1/systemd           | /run/libvirt/virtlockd-sock       |
| unix                                  | 2      | [ ACC ] | STREAM | LISTENING | 34456  | 1902/gnome-session- | @/tmp/.ICE-unix/1902              |
| --More--                              |        |         |        |           |        |                     |                                   |

La commande **ss** peut aussi être utilisée pour connaître la liste des sockets ouverts :

| [root@centos8 ~]# ss -ta |        |         |                      |
|--------------------------|--------|---------|----------------------|
| State                    | Recv-Q | Process | Send-Q               |
| Peer Address:Port        |        |         | Local Address:Port   |
| LISTEN                   | 0      |         | 5                    |
| 0.0.0.0:*                |        |         | 0.0.0.0:5901         |
| LISTEN                   | 0      |         | 128                  |
| 0.0.0.0:*                |        |         | 0.0.0.0:sunrpc       |
| LISTEN                   | 0      |         | 32                   |
| 0.0.0.0:*                |        |         | 192.168.122.1:domain |
| LISTEN                   | 0      |         | 128                  |
| 0.0.0.0:*                |        |         | 0.0.0.0:ssh          |
| LISTEN                   | 0      |         | 5                    |
| 0.0.0.0:*                |        |         | 127.0.0.1:ipp        |
| ESTAB                    | 0      |         | 0                    |
| 151.101.193.91:https     |        |         | 10.0.2.45:50438      |
| ESTAB                    | 0      |         | 0                    |
| 10.0.2.1:52744           |        |         | 10.0.2.45:ssh        |
| ESTAB                    | 0      |         | 0                    |
| 35.190.72.216:https      |        |         | 10.0.2.45:46986      |
| ESTAB                    | 0      |         | 0                    |
| 151.101.193.91:https     |        |         | 10.0.2.45:50442      |
| LISTEN                   | 0      |         | 5                    |
| [::]:*                   |        |         | [::]:5901            |
| LISTEN                   | 0      |         | 128                  |
| [::]:*                   |        |         | [::]:sunrpc          |
| LISTEN                   | 0      |         | 128                  |
| [::]:*                   |        |         | [::]:ssh             |
| LISTEN                   | 0      |         | 5                    |
| [::]:*                   |        |         | [::1]:ipp            |

## Résolution d'adresses Ethernet

Chaque protocole peut être encapsulé dans une **trame** Ethernet. Lorsque la trame doit être transportée de l'expéditeur au destinataire, ce premier doit

connaitre l'adresse Ethernet du dernier. L'adresse Ethernet est aussi appelée l'adresse **Physique** ou l'adresse **MAC**.

Pour connaître l'adresse Ethernet du destinataire, l'expéditeur fait appel au protocol **ARP**. Les informations reçues sont stockées dans une table. Pour visualiser ces informations, il convient d'utiliser la commande suivante :

```
[root@centos8 ~]# arp -a
_gateway (10.0.2.1) at 42:8e:e7:de:a9:b4 [ether] on ens18
```

## Options de la commande

Les options de cette commande sont :

```
[root@centos8 ~]# arp --help
Usage:
arp [-vn]  [<HW>] [-i <if>] [-a] [<hostname>]           <-Display ARP cache
arp [-v]    [-i <if>] -d  <host> [pub]                  <-Delete ARP entry
arp [-vnD]  [<HW>] [-i <if>] -f  [<filename>]            <-Add entry from file
arp [-v]  [<HW>] [-i <if>] -s  <host> <hwaddr> [temp]      <-Add entry
arp [-v]  [<HW>] [-i <if>] -Ds <host> <if> [netmask <nmm>] pub   <-'-'

-a                      display (all) hosts in alternative (BSD) style
-e                      display (all) hosts in default (Linux) style
-s, --set                set a new ARP entry
-d, --delete              delete a specified entry
-v, --verbose             be verbose
-n, --numeric            don't resolve names
-i, --device              specify network interface (e.g. eth0)
-D, --use-device          read <hwaddr> from given device
-A, -p, --protocol        specify protocol family
-f, --file                read new entries from file or from /etc/ethers

<HW>=Use '-H <hw>' to specify hardware address type. Default: ether
List of possible hardware types (which support ARP):
```

```
ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
netrom (AMPR NET/ROM) rose (AMPR ROSE) arcnet (ARCnet)
dlci (Frame Relay DLCI) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
irda (IrLAP) x25 (generic X.25) infiniband (InfiniBand)
eui64 (Generic EUI-64)
```

## Comprendre IPv6

### Présentation

IPv6 peut être utilisé en parallèle avec IPv4 dans un modèle à double pile. Dans cette configuration, une interface réseau peut avoir une ou plusieurs adresses IPv6 ainsi que des adresses IPv4. RHEL 9 fonctionne par défaut en mode double pile.

### Adresses IPv6

Une adresse IPv6 est un nombre de 128 bits, normalement exprimé sous la forme de huit groupes de quatre **nibbles** (demi-octets) hexadécimaux séparés par deux points. Chaque nibble représente quatre bits de l'adresse IPv6, de sorte que chaque groupe représente 16 bits de l'adresse IPv6.

```
2001:0db8:0000:0010:0000:0000:0000:0001
```

Pour faciliter l'écriture des adresses IPv6, il n'est pas nécessaire d'écrire les zéros en tête d'un groupe séparé par deux points. Cependant, au moins un chiffre hexadécimal doit être écrit dans chaque groupe séparé par des deux-points :

```
2001:db8:0:10:0:0:0:1
```

En vertu de ces règles, 2001:db8::0010:0:0:1 serait une autre façon moins pratique d'écrire l'adresse de l'exemple, mais il s'agit d'une représentation valide de la même adresse

Le conseils pour rédiger des adresses lisibles de manière cohérente sont :

- Supprimer les zéros initiaux dans un groupe.
- Utiliser :: pour raccourcir autant que possible.
- Si une adresse contient deux groupes de zéros consécutifs de même longueur, il est préférable de raccourcir les groupes de zéros les plus à gauche en :: et les groupes les plus à droite en :0 : pour chaque groupe.
- Bien que cela soit autorisé, n'utiliser pas :: pour raccourcir un groupe de zéros. Utiliser plutôt :0 : et conserver :: pour les groupes de zéros consécutifs.
- Utiliser toujours des lettres minuscules pour les nombres hexadécimaux de a à f

Par exemple :

```
2001:db8:0:10::1
```

Dernièrement, un socket IPv6 doit comporter les caractères [ ] autour de l'adresse IPv6 :

```
[2001:db8:0:10::1]:80
```

Une adresse unicast IPv6 normale est divisée en deux parties :

- Le préfixe de réseau,
  - Le préfixe identifie le sous-réseau.
- L'identifiant d'interface,
  - Deux interfaces réseau sur le même sous-réseau ne peuvent pas avoir le même identifiant,
  - Un identifiant d'interface identifie une interface particulière sur le sous-réseau.

## Masque de Sous-réseau

Contrairement à l'IPv4, l'IPv6 dispose d'un masque de sous-réseau standard de /64, utilisé pour presque toutes les adresses normales. Cela signifie qu'un seul sous-réseau peut contenir autant d'hôtes que nécessaire. Généralement, le fournisseur de réseau attribue un préfixe plus court à une organisation, par exemple, /48. Cela laisse au reste du réseau la possibilité d'attribuer des sous-réseaux (toujours de longueur /64) à partir du préfixe attribué. Pour un préfixe /48, il reste 16 bits pour les sous-réseaux, soit 65536 sous-réseaux.

Par exemple, dans le cas de l'adresse **2001:0db8:0000:0001:0000:0000:0000:0001**, exprimée en tant que **2001:0db8:0:1/64**, la partie NetID est **2001:0db8:0000:0001** et la partie HostID est **0000:0000:0000:0001**.

En regardant le NetID, la partie **2001:0db8:0000**, exprimée en tant que **2001:db8::/48** représente l'allocation fournie, tandis que **0001/16** représente le sous-réseau.

## Adresses IPv6 Réservées

Les adresses IPv6 réservées à une utilisation spécifique sont :

| Adresse   | Description   |
|-----------|---|
| ::1/128   | L'adresse loopback similaire à l'adresse 127.0.0.1/8  |
| ::        | L'adresse d'écoute global similaire à l'adresse 0.0.0.0   |
| ::/0      | La route par défaut similaire à l'adresse 0.0.0.0/0   |
| 2000::/3  | Cet espace d'adressage concerne les adresses réseaux allouées par l'IANA, allant de 2000::/16 à 3fff::/16 |
| fd00::/8  | Issue de la RFC 4193, ceci est similaire à la RFC 1918, c'est-à-dire un espace d'adressage privé.         |
| fe80::/10 | Adresses Link-local   |
| ff00::/8  | Adresses Multicast  |

## L'Adresse Link-local

Chaque interface sur un réseau est configurée automatiquement avec une adresse Link-local :

```
[root@centos8 ~]# ifconfig
ens18: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.45 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::8af3:5782:3598:aa0f prefixlen 64 scopeid 0x20<link>
            ether 1e:2c:5e:55:02:2b txqueuelen 1000 (Ethernet)
            RX packets 175 bytes 30014 (29.3 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 179 bytes 24197 (23.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens19: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
```

```
ether ee:9d:9e:c3:04:f9  txqueuelen 1000  (Ethernet)
RX packets 0  bytes 0 (0.0 B)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 0  bytes 0 (0.0 B)
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
inet 127.0.0.1  netmask 255.0.0.0
inet6 ::1  prefixlen 128  scopeid 0x10<host>
loop  txqueuelen 1000  (Local Loopback)
RX packets 0  bytes 0 (0.0 B)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 0  bytes 0 (0.0 B)
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
ether 52:54:00:79:02:66  txqueuelen 1000  (Ethernet)
RX packets 0  bytes 0 (0.0 B)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 0  bytes 0 (0.0 B)
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Pour tester la connectivité de l'adresse Link-local, il convient d'utiliser la commande **ping6** :

```
[root@centos8 ~]# ping6 -c4 fe80::8af3:5782:3598:aa0f%ens18
PING fe80::8af3:5782:3598:aa0f%ens18(fe80::8af3:5782:3598:aa0f%ens18) 56 data bytes
64 bytes from fe80::8af3:5782:3598:aa0f%ens18: icmp_seq=1 ttl=64 time=0.093 ms
64 bytes from fe80::8af3:5782:3598:aa0f%ens18: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from fe80::8af3:5782:3598:aa0f%ens18: icmp_seq=3 ttl=64 time=0.115 ms
64 bytes from fe80::8af3:5782:3598:aa0f%ens18: icmp_seq=4 ttl=64 time=0.051 ms

--- fe80::8af3:5782:3598:aa0f%ens18 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3106ms
```

```
rtt min/avg/max/mdev = 0.051/0.079/0.115/0.027 ms
```



**Important :** Notez qu'à la fin de l'adresse, il faut ajouter le **scope** qui est représenté par le caractère % suivi par le nom de l'interface.

## DHCPv6

DHCPv6 ne fonctionne pas de la même façon que DHCPv4 parce qu'il n'existe pas d'adresses de diffusion sous IPv6.

En résumé, l'hôte envoie une requête DHCPv6 à l'adresse multicast **all-dhcp-servers, ff02::1:2** sur le port **547/udp**. En retour, l'hôte reçoit les informations demandées sur le port **546/udp** de son adresse Link-local.

## Configurer le Réseau

RHEL/CentOS 8 utilise **Network Manager** pour gérer le réseau. Network Manager est composé de deux éléments :

- un service qui gère les connexions réseaux et rapporte leurs états,
- des front-ends qui passent par un API de configuration du service.



**Important :** Notez qu'avec cette version de NetworkManager, IPv6 est activée par défaut.

Le service NetworkManager doit toujours être lancé :

```
[root@centos8 ~]# systemctl status NetworkManager.service
● NetworkManager.service - Network Manager
  Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; vendor pr>
```

```
Active: active (running) since Thu 2021-07-22 05:05:29 EDT; 1 months 7 days ago
  Docs: man:NetworkManager(8)
Main PID: 1002 (NetworkManager)
  Tasks: 3 (limit: 23535)
Memory: 6.8M
CGroup: /system.slice/NetworkManager.service
        └─1002 /usr/sbin/NetworkManager --no-daemon
```

Warning: Journal has been rotated since unit was started. Log output is incomplete or>
lines 1-11/11 (END)
[<sup>^</sup>q]

## La Commande nmcli

La commande **nmcli** (Network Manager Command Line Interface) est utilisée pour configurer NetworkManager.

Les options et les sous-commandes peuvent être consultées en utilisant les commandes suivantes :

```
[root@centos8 ~]# nmcli help
Usage: nmcli [OPTIONS] OBJECT { COMMAND | help }

OPTIONS
-a, --ask                         ask for missing parameters
-c, --colors auto|yes|no           whether to use colors in output
-e, --escape yes|no                escape columns separators in values
-f, --fields <field,...>|all|common  specify fields to output
-g, --get-values <field,...>|all|common shortcut for -m tabular -t -f
-h, --help                          print this help
-m, --mode tabular|multiline      output mode
-o, --overview                     overview mode
-p, --pretty                        pretty output
-s, --show-secrets                  allow displaying passwords
-t, --terse                         terse output
```

```
-v, --version                      show program version
-w, --wait <seconds>              set timeout waiting for finishing operations
```

## OBJECT

|              |  |
|--------------|--|
| g[eneral]    | NetworkManager's general status and operations |
| n[etworking] | overall networking control                     |
| r[adio]      | NetworkManager radio switches                  |
| c[onnection] | NetworkManager's connections                   |
| d[evice]     | devices managed by NetworkManager              |
| a[gent]      | NetworkManager secret agent or polkit agent    |
| m[onitor]    | monitor NetworkManager changes                 |

# LAB #1 - Configuration du Réseau

## 1.1 - Connections et Profils

NetworkManager inclus la notion de **connections** ou **profils** permettant des configurations différentes en fonction de la localisation. Pour voir les connections actuelles, utilisez la commande **nmcli c** avec la sous-commande **show** :

```
[root@centos8 ~]# nmcli c show
NAME      UUID                                  TYPE      DEVICE
ens18     fc4a4d23-b15e-47a7-bcfa-b2e08f49553e  ethernet  ens18
virbr0   03f6c432-2a09-47e7-9693-208431a572ee  bridge    virbr0
```

Créez donc un profil IP fixe rattaché au périphérique **ens18** :

```
[root@centos8 ~]# nmcli connection add con-name ip_fixe iface ens18 type ethernet ip4 10.0.2.46/24 gw4 10.0.2.1
Connection 'ip_fixe' (0f48c74d-5d16-4c37-8220-24644507b589) successfully added.
```

Constatez sa présence :

```
[root@centos8 ~]# nmcli c show
NAME      UUID
ens18    fc4a4d23-b15e-47a7-bcfa-b2e08f49553e
virbr0   03f6c432-2a09-47e7-9693-208431a572ee
ip_fixe  0f48c74d-5d16-4c37-8220-24644507b589
TYPE      DEVICE
ethernet  ens18
bridge    virbr0
ethernet  --
```

Notez que la sortie n'indique pas que le profil **ip\_fixe** soit associé au périphérique **ens18** car le profil **ip\_fixe** n'est pas activé :

```
[root@centos8 ~]# nmcli d show
GENERAL.DEVICE:                         ens18
GENERAL.TYPE:                            ethernet
GENERAL.HWADDR:                          4E:B1:31:BD:5D:B2
GENERAL.MTU:                             1500
GENERAL.STATE:                           100 (connected)
GENERAL.CONNECTION:                      ens18
GENERAL.CON-PATH:                        /org/freedesktop/NetworkManager/ActiveConnect>
WIRED-PROPERTIES.CARRIER:                on
IP4.ADDRESS[1]:                          10.0.2.45/24
IP4.GATEWAY:                            10.0.2.1
IP4.ROUTE[1]:                           dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 10.0.2.1, mt = 100
IP4.DNS[1]:                             8.8.8.8
IP6.ADDRESS[1]:                          fe80::86b6:8d39:cab2:d84d/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]:                           dst = ff00::/8, nh = ::, mt = 256, table=255

GENERAL.DEVICE:                          virbr0
GENERAL.TYPE:                            bridge
GENERAL.HWADDR:                          52:54:00:79:02:66
GENERAL.MTU:                             1500
GENERAL.STATE:                           100 (connected (externally))
GENERAL.CONNECTION:                     virbr0
GENERAL.CON-PATH:                        /org/freedesktop/NetworkManager/ActiveConnect>
```

```
IP4.ADDRESS[1]:          192.168.122.1/24
IP4.GATEWAY:             --
IP4.ROUTE[1]:            dst = 192.168.122.0/24, nh = 0.0.0.0, mt = 0
IP6.GATEWAY:              --

GENERAL.DEVICE:          lo
GENERAL.TYPE:             loopback
GENERAL.HWADDR:           00:00:00:00:00:00
GENERAL.MTU:               65536
GENERAL.STATE:             10 (unmanaged)
GENERAL.CONNECTION:        --
GENERAL.CON-PATH:          --
IP4.ADDRESS[1]:          127.0.0.1/8
IP4.GATEWAY:              --
IP6.ADDRESS[1]:            ::1/128
IP6.GATEWAY:              --
IP6.ROUTE[1]:              dst = ::1/128, nh = ::, mt = 256

GENERAL.DEVICE:           virbr0-nic
GENERAL.TYPE:              tun
GENERAL.HWADDR:            52:54:00:79:02:66
GENERAL.MTU:                1500
GENERAL.STATE:              10 (unmanaged)
GENERAL.CONNECTION:        --
GENERAL.CON-PATH:          --
lines 28-50/50 (END)
[q]
```

Pour activer le profil ip\_fixe, utilisez la commande suivante :

```
[root@centos8 ~]# nmcli connection up ip_fixe
```

Notez que votre terminal est bloqué à cause du changement de l'adresse IP.



**A faire** - Revenez à l'accueil du cloud IT Training et re-connectez-vous à la VM en tant que trainee en utilisant la connexion **CentOS8\_SSH\_10.0.2.46**.

Le profil ip\_fixe est maintenant activé tandis que le profil enp0s3 a été désactivé :

```
[root@centos8 ~]# nmcli c show
NAME      UUID                                  TYPE      DEVICE
ip_fixe   0f48c74d-5d16-4c37-8220-24644507b589  ethernet  ens18
virbr0    03f6c432-2a09-47e7-9693-208431a572ee  bridge    virbr0
ens18     fc4a4d23-b15e-47a7-bcfa-b2e08f49553e  ethernet  --
[root@centos8 ~]# nmcli d show
GENERAL.DEVICE:                         ens18
GENERAL.TYPE:                            ethernet
GENERAL.HWADDR:                          4E:B1:31:BD:5D:B2
GENERAL.MTU:                             1500
GENERAL.STATE:                           100 (connected)
GENERAL.CONNECTION:                      ip_fixe
GENERAL.CON-PATH:                        /org/freedesktop/NetworkManager/ActiveConnect>
WIRED-PROPERTIES.CARRIER:                on
IP4.ADDRESS[1]:                          10.0.2.46/24
IP4.GATEWAY:                            10.0.2.1
IP4.ROUTE[1]:                           dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 0.0.0.0/0, nh = 10.0.2.1, mt = 100
IP6.ADDRESS[1]:                          fe80::5223:aeel:998e:9f27/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]:                           dst = ff00::/8, nh = ::, mt = 256, table=255

GENERAL.DEVICE:                          virbr0
GENERAL.TYPE:                           bridge
GENERAL.HWADDR:                         52:54:00:79:02:66
```

```
GENERAL.MTU:                      1500
GENERAL.STATE:                    100 (connected (externally))
GENERAL.CONNECTION:               virbr0
GENERAL.CON-PATH:                 /org/freedesktop/NetworkManager/ActiveConnect>
IP4.ADDRESS[1]:                  192.168.122.1/24
IP4.GATEWAY:                     --
IP4.ROUTE[1]:                    dst = 192.168.122.0/24, nh = 0.0.0.0, mt = 0
IP6.GATEWAY:                     --
GENERAL.DEVICE:                  lo
GENERAL.TYPE:                     loopback
GENERAL.HWADDR:                  00:00:00:00:00:00
GENERAL.MTU:                      65536
GENERAL.STATE:                   10 (unmanaged)
GENERAL.CONNECTION:              --
GENERAL.CON-PATH:                --
IP4.ADDRESS[1]:                  127.0.0.1/8
IP4.GATEWAY:                     --
IP6.ADDRESS[1]:                  ::1/128
IP6.GATEWAY:                     --
IP6.ROUTE[1]:                    dst = ::1/128, nh = ::, mt = 256
GENERAL.DEVICE:                  virbr0-nic
GENERAL.TYPE:                     tun
GENERAL.HWADDR:                  52:54:00:79:02:66
GENERAL.MTU:                      1500
GENERAL.STATE:                   10 (unmanaged)
GENERAL.CONNECTION:              --
GENERAL.CON-PATH:                --
lines 27-49/49 (END)
[q]
```

Pour consulter les paramètres du profil **ens18**, utilisez la commande suivante :

```
[root@centos8 ~]# nmcli -p connection show ens18
=====
              Connection profile details (ens18)
=====

connection.id:                  ens18
connection.uuid:                fc4a4d23-b15e-47a7-bcfa-b2e08f49553e
connection.stable-id:           --
connection.type:                802-3-ethernet
connection.interface-name:      ens18
connection.autoconnect:         yes
connection.autoconnect-priority: 0
connection.autoconnect-retries: -1 (default)
connection.multi-connect:       0 (default)
connection.auth-retries:        -1
connection.timestamp:           1630224060
connection.read-only:           no
connection.permissions:         --
connection.zone:                --
connection.master:              --
connection.slave-type:          --
connection.autoconnect-slaves:  -1 (default)
connection.secondaries:          --
connection.gateway-ping-timeout: 0
connection.metered:              unknown
connection.lldp:                default
connection.mdns:                -1 (default)
connection.llmnr:                -1 (default)
connection.wait-device-timeout:  -1
-----
802-3-ethernet.port:            --
802-3-ethernet.speed:           0
802-3-ethernet.duplex:          --
802-3-ethernet.auto-negotiate: no
802-3-ethernet.mac-address:     --
```

```
802-3-ethernet.cloned-mac-address:      --
802-3-ethernet.generate-mac-address-mask:--
802-3-ethernet.mac-address-blacklist:    --
802-3-ethernet.mtu:                     auto
802-3-ethernet.s390-subchannels:        --
802-3-ethernet.s390-nettype:            --
802-3-ethernet.s390-options:           --
802-3-ethernet.wake-on-lan:             default
802-3-ethernet.wake-on-lan-password:   --

-----
ipv4.method:                           manual
ipv4.dns:                             8.8.8.8
ipv4.dns-search:                      ittraining.loc
ipv4.dns-options:                     --
ipv4.dns-priority:                   0
ipv4.addresses:                       10.0.2.45/24
ipv4.gateway:                         10.0.2.1
ipv4.routes:                          --
ipv4.route-metric:                    -1
ipv4.route-table:                     0 (unspec)
ipv4.routing-rules:                  --
ipv4.ignore-auto-routes:              no
ipv4.ignore-auto-dns:                no
ipv4.dhcp-client-id:                 --
ipv4.dhcp-iaid:                       --
ipv4.dhcp-timeout:                   0 (default)
ipv4.dhcp-send-hostname:              yes
ipv4.dhcp-hostname:                  --
ipv4.dhcp-fqdn:                       --
ipv4.dhcp-hostname-flags:             0x0 (none)
ipv4.never-default:                  no
ipv4.may-fail:                        yes
ipv4.dad-timeout:                   -1 (default)
ipv4.dhcp-vendor-class-identifier:   --
```

```
ipv4.dhcp-reject-servers:          --
-----
ipv6.method:                      auto
ipv6.dns:                         --
ipv6.dns-search:                  --
ipv6.dns-options:                 --
ipv6.dns-priority:                0
ipv6.addresses:                   --
ipv6.gateway:                     --
ipv6.routes:                      --
ipv6.route-metric:                -1
ipv6.route-table:                 0 (unspec)
ipv6.routing-rules:               --
ipv6.ignore-auto-routes:          no
ipv6.ignore-auto-dns:             no
ipv6.never-default:               no
ipv6.may-fail:                    yes
ipv6.ip6-privacy:                 0 (disabled)
ipv6.addr-gen-mode:               stable-privacy
ipv6.ra-timeout:                  0 (default)
ipv6.dhcp-duid:                   --
ipv6.dhcp-iaid:                   --
ipv6.dhcp-timeout:                0 (default)
ipv6.dhcp-send-hostname:          yes
ipv6.dhcp-hostname:               --
ipv6.dhcp-hostname-flags:         0x0 (none)
ipv6.token:                       --
-----
proxy.method:                     none
proxy.browser-only:              no
proxy.pac-url:                   --
proxy.pac-script:                 --
-----
lines 56-100/100 (END)
```

[q]

De même, pour consulter les paramètres du profil **ip\_fixe**, utilisez la commande suivante :

```
[root@centos8 ~]# nmcli -p connection show ip_fixe
=====
              Connection profile details (ip_fixe)
=====

connection.id:          ip_fixe
connection.uuid:        0f48c74d-5d16-4c37-8220-24644507b589
connection.stable-id:   --
connection.type:         802-3-ethernet
connection.interface-name: ens18
connection.autoconnect: yes
connection.autoconnect-priority: 0
connection.autoconnect-retries: -1 (default)
connection.multi-connect: 0 (default)
connection.auth-retries: -1
connection.timestamp:    1630224329
connection.read-only:   no
connection.permissions: --
connection.zone:         --
connection.master:       --
connection.slave-type:   --
connection.autoconnect-slaves: -1 (default)
connection.secondaries:  --
connection.gateway-ping-timeout: 0
connection.metered:     unknown
connection.lldp:        default
connection.mdns:        -1 (default)
connection.llmnr:        -1 (default)
connection.wait-device-timeout: -1
-----
802-3-ethernet.port:   --
```

```
802-3-ethernet.speed:          0
802-3-ethernet.duplex:        --
802-3-ethernet.auto-negotiate: no
802-3-ethernet.mac-address:    --
802-3-ethernet.cloned-mac-address: --
802-3-ethernet.generate-mac-address-mask: --
802-3-ethernet.mac-address-blacklist: --
802-3-ethernet.mtu:           auto
802-3-ethernet.s390-subchannels: --
802-3-ethernet.s390-nettype:   --
802-3-ethernet.s390-options:  --
802-3-ethernet.wake-on-lan:    default
802-3-ethernet.wake-on-lan-password: --

-----
ipv4.method:                  manual
ipv4.dns:                     --
ipv4.dns-search:              --
ipv4.dns-options:             --
ipv4.dns-priority:            0
ipv4.addresses:                10.0.2.46/24
ipv4.gateway:                 10.0.2.1
ipv4.routes:                   --
ipv4.route-metric:             -1
ipv4.route-table:              0 (unspec)
ipv4.routing-rules:            --
ipv4.ignore-auto-routes:       no
ipv4.ignore-auto-dns:          no
ipv4.dhcp-client-id:          --
ipv4.dhcp-iaid:                --
ipv4.dhcp-timeout:             0 (default)
ipv4.dhcp-send-hostname:       yes
ipv4.dhcp-hostname:            --
ipv4.dhcp-fqdn:                --
ipv4.dhcp-hostname-flags:      0x0 (none)
```

```
ipv4.never-default:          no
ipv4.may-fail:               yes
ipv4.dad-timeout:            -1 (default)
ipv4.dhcp-vendor-class-identifier:  --
ipv4.dhcp-reject-servers:    --

-----
ipv6.method:                 auto
ipv6.dns:                    --
ipv6.dns-search:              --
ipv6.dns-options:             --
ipv6.dns-priority:            0
ipv6.addresses:               --
ipv6.gateway:                --
ipv6.routes:                  --
ipv6.route-metric:             -1
ipv6.route-table:              0 (unspec)
ipv6.routing-rules:            --
ipv6.ignore-auto-routes:      no
ipv6.ignore-auto-dns:         no
ipv6.never-default:           no
ipv6.may-fail:                yes
ipv6.ip6-privacy:              -1 (unknown)
ipv6.addr-gen-mode:            stable-privacy
ipv6.ra-timeout:               0 (default)
ipv6.dhcp-duid:                --
ipv6.dhcp-iaid:                --
ipv6.dhcp-timeout:              0 (default)
ipv6.dhcp-send-hostname:       yes
ipv6.dhcp-hostname:             --
ipv6.dhcp-hostname-flags:      0x0 (none)
ipv6.token:                   --

-----
proxy.method:                 none
proxy.browser-only:            no
```

```
proxy.pac-url:          --
proxy.pac-script:        --
=====
=====
Activate connection details (0f48c74d-5d16-4c37-8220-24644507b589)
=====
GENERAL.NAME:            ip_fixe
GENERAL.UUID:           0f48c74d-5d16-4c37-8220-24644507b589
GENERAL.DEVICES:         ens18
GENERAL.IP-IFACE:        ens18
GENERAL.STATE:           activated
GENERAL.DEFAULT:          yes
GENERAL.DEFAULT6:         no
GENERAL.SPEC-OBJECT:      --
GENERAL.VPN:              no
GENERAL.DBUS-PATH:        /org/freedesktop/NetworkManager/ActiveConnection/4
GENERAL.CON-PATH:         /org/freedesktop/NetworkManager/Settings/4
GENERAL.ZONE:             --
GENERAL.MASTER-PATH:      --
=====
IP4.ADDRESS[1]:          10.0.2.46/24
IP4.GATEWAY:              10.0.2.1
IP4.ROUTE[1]:             dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:             dst = 0.0.0.0/0, nh = 10.0.2.1, mt = 100
=====
IP6.ADDRESS[1]:          fe80::5223:aee1:998e:9f27/64
IP6.GATEWAY:              --
IP6.ROUTE[1]:             dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]:             dst = ff00::/8, nh = ::, mt = 256, table=255
=====
lines 83-127/127 (END)
[q]
```

Pour consulter la liste profils associés à un périphérique, utilisez la commande suivante :

```
[root@centos8 ~]# nmcli -f CONNECTIONS device show ens18
CONNECTIONS.AVAILABLE-CONNECTION-PATHS:
/org/freedesktop/NetworkManager/Settings/1,/org/freedesktop/NetworkManager/Settings/4
CONNECTIONS.AVAILABLE-CONNECTIONS[1]:    fc4a4d23-b15e-47a7-bcfa-b2e08f49553e | ens18
CONNECTIONS.AVAILABLE-CONNECTIONS[2]:    0f48c74d-5d16-4c37-8220-24644507b589 | ip_fixe
```

Les fichiers de configuration pour le périphérique **ens18** se trouvent dans le répertoire **/etc/sysconfig/network-scripts/** :

```
[root@centos8 ~]# ls -l /etc/sysconfig/network-scripts/ | grep ifcfg
-rw-r--r--. 1 root root 417 Jun 16 06:39 ifcfg-ens18
-rw-r--r--. 1 root root 326 Aug 29 03:58 ifcfg-ip_fixe
```

## 1.2 - Résolution des Noms

L'étude du fichier **/etc/sysconfig/network-scripts/ifcfg-ip\_fixe** démontre l'absence de directives concernant les DNS :

```
[root@centos8 ~]# cat /etc/sysconfig/network-scripts/ifcfg-ip_fixe
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=10.0.2.46
PREFIX=24
GATEWAY=10.0.2.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ip_fixe
```

```
UUID=0f48c74d-5d16-4c37-8220-24644507b589
DEVICE=ens18
ONBOOT=yes
```

La résolution des noms est donc inactive :

```
[root@centos8 ~]# ping www.free.fr
ping: www.free.fr: Name or service not known
```

Modifiez donc la configuration du profil **ip\_fixe** :

```
[root@centos8 ~]# nmcli connection mod ip_fixe ipv4.dns 8.8.8.8
```

L'étude du fichier **/etc/sysconfig/network-scripts/ifcfg-ip\_fixe** démontre que la directive concernant le serveur DNS a été ajoutée :

```
[root@centos8 ~]# cat /etc/sysconfig/network-scripts/ifcfg-ip_fixe
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=10.0.2.46
PREFIX=24
GATEWAY=10.0.2.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ip_fixe
UUID=0f48c74d-5d16-4c37-8220-24644507b589
DEVICE=ens18
ONBOOT=yes
```

## DNS1=8.8.8.8

Afin que la modification du serveur DNS soit prise en compte, re-démarrez le service NetworkManager :

```
root@centos8 ~]# systemctl restart NetworkManager.service
[root@centos8 ~]# systemctl status NetworkManager.service
● NetworkManager.service - Network Manager
    Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
    Active: active (running) since Sun 2021-08-29 04:15:11 EDT; 8s ago
      Docs: man:NetworkManager(8)
   Main PID: 973390 (NetworkManager)
     Tasks: 4 (limit: 23535)
    Memory: 4.6M
      CGroup: /system.slice/NetworkManager.service
              └─973390 /usr/sbin/NetworkManager --no-daemon

Aug 29 04:15:12 centos8.ittraining.loc NetworkManager[973390]: <info> [1630224912.2235] device (ens18): state
change: ip-check -> secondaries (reas>
Aug 29 04:15:12 centos8.ittraining.loc NetworkManager[973390]: <info> [1630224912.2237] device (virbr0): state
change: secondaries -> activated (re>
Aug 29 04:15:12 centos8.ittraining.loc NetworkManager[973390]: <info> [1630224912.2241] manager: NetworkManager
state is now CONNECTED_LOCAL
Aug 29 04:15:12 centos8.ittraining.loc NetworkManager[973390]: <info> [1630224912.2251] policy: set 'ip_fixe'
(ens18) as default for IPv4 routing a>
Aug 29 04:15:12 centos8.ittraining.loc NetworkManager[973390]: <info> [1630224912.3090] device (virbr0):
Activation: successful, device activated.
Aug 29 04:15:12 centos8.ittraining.loc NetworkManager[973390]: <info> [1630224912.3098] device (ens18): state
change: secondaries -> activated (rea>
Aug 29 04:15:12 centos8.ittraining.loc NetworkManager[973390]: <info> [1630224912.3102] manager: NetworkManager
state is now CONNECTED_SITE
Aug 29 04:15:12 centos8.ittraining.loc NetworkManager[973390]: <info> [1630224912.3111] device (ens18):
Activation: successful, device activated.
Aug 29 04:15:12 centos8.ittraining.loc NetworkManager[973390]: <info> [1630224912.3116] manager: NetworkManager
state is now CONNECTED_GLOBAL
```

```
Aug 29 04:15:12 centos8.ittraining.loc NetworkManager[973390]: <info>  [1630224912.3121] manager: startup complete
lines 1-20/20 (END)
[q]
```

Vérifiez que le fichier **/etc/resolv.conf** ait été modifié par NetworkManager :

```
[root@centos8 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
search ittraining.loc
nameserver 8.8.8.8
```

Dernièrement vérifiez la resolution des noms :

```
[root@centos8 ~]# ping www.free.fr
PING www.free.fr (212.27.48.10) 56(84) bytes of data.
64 bytes from www.free.fr (212.27.48.10): icmp_seq=1 ttl=47 time=29.3 ms
64 bytes from www.free.fr (212.27.48.10): icmp_seq=2 ttl=47 time=29.4 ms
64 bytes from www.free.fr (212.27.48.10): icmp_seq=3 ttl=47 time=29.4 ms
64 bytes from www.free.fr (212.27.48.10): icmp_seq=4 ttl=47 time=29.4 ms
^C
--- www.free.fr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 29.266/29.377/29.428/0.183 ms
```



**Important** : Notez qu'il existe un front-end graphique en mode texte, **nmtui**, pour configurer NetworkManager.

### 1.3 - Ajouter une Deuxième Adresse IP à un Profil

Pour ajouter une deuxième adresse IP à un profil sous RHEL/CentOS 8, il convient d'utiliser la commande suivante :

```
[root@centos8 ~]# nmcli connection mod ip_fixe +ipv4.addresses 192.168.1.2/24
```

Rechargez la configuration du profil :

```
[root@centos8 ~]# nmcli con up ip_fixe
```

Saisissez ensuite la commande suivante :

```
[root@centos8 ~]# nmcli connection show ip_fixe
connection.id:                      ip_fixe
connection.uuid:                     0f48c74d-5d16-4c37-8220-24644507b589
connection.stable-id:                --
connection.type:                    802-3-ethernet
connection.interface-name:          ens18
connection.autoconnect:             yes
connection.autoconnect-priority:    0
connection.autoconnect-retries:     -1 (default)
connection.multi-connect:           0 (default)
connection.auth-retries:            -1
connection.timestamp:               1630225792
connection.read-only:                no
connection.permissions:              --
connection.zone:                   --
connection.master:                  --
connection.slave-type:              --
connection.autoconnect-slaves:     -1 (default)
connection.secondaries:              --
connection.gateway-ping-timeout:   0
connection.metered:                 unknown
```

```
connection.lldp:           default
connection.mdns:            -1 (default)
connection.llmnr:           -1 (default)
connection.wait-device-timeout: -1
802-3-ethernet.port:        --
802-3-ethernet.speed:       0
802-3-ethernet.duplex:      --
802-3-ethernet.auto-negotiate: no
802-3-ethernet.mac-address:  --
802-3-ethernet.cloned-mac-address: --
802-3-ethernet.generate-mac-address-mask: --
802-3-ethernet.mac-address-blacklist: --
802-3-ethernet.mtu:         auto
802-3-ethernet.s390-subchannels: --
802-3-ethernet.s390-nettype:   --
802-3-ethernet.s390-options:  --
802-3-ethernet.wake-on-lan:    default
802-3-ethernet.wake-on-lan-password: --
ipv4.method:                manual
ipv4.dns:                   8.8.8.8
ipv4.dns-search:             --
ipv4.dns-options:            --
ipv4.dns-priority:           0
ipv4.addresses:              10.0.2.46/24, 192.168.1.2/24
ipv4.gateway:                10.0.2.1
ipv4.routes:                 --
ipv4.route-metric:            -1
ipv4.route-table:             0 (unspec)
ipv4.routing-rules:           --
ipv4.ignore-auto-routes:      no
ipv4.ignore-auto-dns:          no
ipv4.dhcp-client-id:          --
ipv4.dhcp-iaid:                --
ipv4.dhcp-timeout:             0 (default)
```

```
ipv4.dhcp-send-hostname: yes
ipv4.dhcp-hostname: --
ipv4.dhcp-fqdn: --
ipv4.dhcp-hostname-flags: 0x0 (none)
ipv4.never-default: no
ipv4.may-fail: yes
ipv4.dad-timeout: -1 (default)
ipv4.dhcp-vendor-class-identifier: --
ipv4.dhcp-reject-servers: --
ipv6.method: auto
ipv6.dns: --
ipv6.dns-search: --
ipv6.dns-options: --
ipv6.dns-priority: 0
ipv6.addresses: --
ipv6.gateway: --
ipv6.routes: --
ipv6.route-metric: -1
ipv6.route-table: 0 (unspec)
ipv6.routing-rules: --
ipv6.ignore-auto-routes: no
ipv6.ignore-auto-dns: no
ipv6.never-default: no
ipv6.may-fail: yes
ipv6.ip6-privacy: -1 (unknown)
ipv6.addr-gen-mode: stable-privacy
ipv6.ra-timeout: 0 (default)
ipv6.dhcp-duid: --
ipv6.dhcp-iaid: --
ipv6.dhcp-timeout: 0 (default)
ipv6.dhcp-send-hostname: yes
ipv6.dhcp-hostname: --
ipv6.dhcp-hostname-flags: 0x0 (none)
ipv6.token: --
```

```
proxy.method: none
proxy.browser-only: no
proxy.pac-url: --
proxy.pac-script: --
GENERAL.NAME: ip_fixe
GENERAL.UUID: 0f48c74d-5d16-4c37-8220-24644507b589
GENERAL.DEVICES: ens18
GENERAL.IP-IFACE: ens18
GENERAL.STATE: activated
GENERAL.DEFAULT: yes
GENERAL.DEFAULT6: no
GENERAL.SPEC-OBJECT: --
GENERAL.VPN: no
GENERAL.DBUS-PATH: /org/freedesktop/NetworkManager/ActiveConnection/3
GENERAL.CON-PATH: /org/freedesktop/NetworkManager/Settings/2
GENERAL.ZONE: --
GENERAL.MASTER-PATH: --
IP4.ADDRESS[1]: 10.0.2.46/24
IP4.ADDRESS[2]: 192.168.1.2/24
IP4.GATEWAY: 10.0.2.1
IP4.ROUTE[1]: dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]: dst = 192.168.1.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[3]: dst = 0.0.0.0/0, nh = 10.0.2.1, mt = 100
IP4.DNS[1]: 8.8.8.8
IP6.ADDRESS[1]: fe80::5223:aee1:998e:9f27/64
IP6.GATEWAY: --
IP6.ROUTE[1]: dst = fe80::/64, nh = ::, mt = 100
IP6.ROUTE[2]: dst = ff00::/8, nh = ::, mt = 256, table=255
lines 72-116/116 (END)
[q]
```



**Important :** Notez l'ajout de l'adresse secondaire à la ligne **ipv4.addresses:** ainsi que l'ajout de la ligne **IP4.ADDRESS[2]:**

Consultez maintenant le contenu du fichier **/etc/sysconfig/network-scripts/ifcfg-ip\_fixe** :

```
[root@centos8 ~]# cat /etc/sysconfig/network-scripts/ifcfg-ip_fixe
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=10.0.2.46
PREFIX=24
GATEWAY=10.0.2.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ip_fixe
UUID=0f48c74d-5d16-4c37-8220-24644507b589
DEVICE=ens18
ONBOOT=yes
DNS1=8.8.8.8
IPADDR1=192.168.1.2
PREFIX1=24
```



**Important** : Notez l'ajout de la ligne **IPADDR1=192.168.1.2**.

## 1.4 - La Commande hostname

La procédure de la modification du hostname est simplifiée et sa prise en compte est immédiate :

```
[root@centos8 ~]# hostname  
centos8.ittraining.loc  
  
[root@centos8 ~]# nmcli general hostname centos.ittraining.loc  
  
[root@centos8 ~]# cat /etc/hostname  
centos.ittraining.loc  
  
[root@centos8 ~]# hostname  
centos.ittraining.loc  
  
[root@centos8 ~]# nmcli general hostname centos8.ittraining.loc  
  
[root@centos8 ~]# cat /etc/hostname  
centos8.ittraining.loc  
  
[root@centos8 ~]# hostname  
centos8.ittraining.loc
```

## 1.5 - La Commande ip

Sous RHEL/CentOS 8 la commande **ip** est préférée par rapport à la commande ifconfig :

```
[root@centos8 ~]# ip address  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 4e:b1:31:bd:b2 brd ff:ff:ff:ff:ff:ff
```

```

inet 10.0.2.46/24 brd 10.0.2.255 scope global noprefixroute ens18
    valid_lft forever preferred_lft forever
inet 192.168.1.2/24 brd 192.168.1.255 scope global noprefixroute ens18
    valid_lft forever preferred_lft forever
inet6 fe80::5223:aee1:998e:9f27/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:79:02:66 brd ff:ff:ff:ff:ff:ff
        inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
            valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:79:02:66 brd ff:ff:ff:ff:ff:ff

```

## Options de la Commande ip

Les options de cette commande sont :

```

[root@centos8 ~]# ip --help
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
      ip [ -force ] -batch filename
where OBJECT := { link | address | addrlabel | route | rule | neigh | ntable |
                  tunnel | tuntap | maddress | mroute | mrule | monitor | xfrm |
                  netns | l2tp | fou | macsec | tcp_metrics | token | netconf | ila |
                  vrf | sr | nexthop | mptcp }
OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
             -h[uman-readable] | -iec | -j[son] | -p[retty] |
             -f[amily] { inet | inet6 | mpls | bridge | link } |
             -4 | -6 | -I | -D | -M | -B | -0 |
             -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
             -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
             -rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
             -c[olor]} 
```

## 1.6 - Activer/Désactiver une Interface Manuellement

Deux commandes existent pour désactiver et activer manuellement une interface réseau :

```
# nmcli device disconnect enp0s3  
# nmcli device connect enp0s3
```



**Important** : Veuillez ne **PAS** exécuter ces deux commandes.

## 1.7 - Routage Statique

### La commande ip

Sous RHEL/CentOS 8, pour supprimer la route vers le réseau 192.168.1.0 il convient d'utiliser la commande ip et non pas la commande route :

```
[root@centos8 ~]# ip route  
default via 10.0.2.1 dev ens18 proto static metric 100  
10.0.2.0/24 dev ens18 proto kernel scope link src 10.0.2.46 metric 100  
192.168.1.0/24 dev ens18 proto kernel scope link src 192.168.1.2 metric 100  
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown  
  
root@centos8 ~]# ip route del 192.168.1.0/24 via 0.0.0.0  
[root@centos8 ~]# ip route  
default via 10.0.2.1 dev ens18 proto static metric 100  
10.0.2.0/24 dev ens18 proto kernel scope link src 10.0.2.46 metric 100  
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
```

Pour ajouter la route vers le réseau 192.168.1.0 :

```
[root@centos8 ~]# ip route add 192.168.1.0/24 via 10.0.2.1
[root@centos8 ~]# ip route
default via 10.0.2.1 dev ens18 proto static metric 100
10.0.2.0/24 dev ens18 proto kernel scope link src 10.0.2.46 metric 100
192.168.1.0/24 via 10.0.2.1 dev ens18
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
```



**Important** - La commande utilisée pour ajouter une passerelle par défaut prend la forme suivante **ip route add default via adresse ip**.

## Désactiver/Activer le routage sur le serveur

Pour désactiver le routage sur le serveur, il convient de désactiver la retransmission des paquets.

Pour IPv4 :

```
[root@centos8 ~]# cat /proc/sys/net/ipv4/ip_forward
1
[root@centos8 ~]# echo 0 > /proc/sys/net/ipv4/ip_forward
[root@centos8 ~]# cat /proc/sys/net/ipv4/ip_forward
0
```

Pour activer le routage sur le serveur, il convient d'activer la retransmission des paquets:

```
[root@centos8 ~]# echo 1 > /proc/sys/net/ipv4/ip_forward
[root@centos8 ~]# cat /proc/sys/net/ipv4/ip_forward
1
```

Pour IPv6 :

```
[root@centos8 ~]# cat /proc/sys/net/ipv6/conf/all/forwarding  
1  
[root@centos8 ~]# echo "0" > /proc/sys/net/ipv6/conf/all/forwarding  
[root@centos8 ~]# cat /proc/sys/net/ipv6/conf/all/forwarding  
0
```

Pour activer le routage sur le serveur, il convient d'activer la retransmission des paquets:

```
[root@centos8 ~]# echo "1" > /proc/sys/net/ipv6/conf/all/forwarding  
[root@centos8 ~]# cat /proc/sys/net/ipv6/conf/all/forwarding  
1
```

## LAB #2 - Diagnostique du Réseau

### 2.1 - ping

Pour tester l'accessibilité d'une machine, vous devez utiliser la commande **ping** :

```
[root@centos8 ~]# ping -c4 10.0.2.1  
PING 10.0.2.1 (10.0.2.1) 56(84) bytes of data.  
64 bytes from 10.0.2.1: icmp_seq=1 ttl=64 time=0.104 ms  
64 bytes from 10.0.2.1: icmp_seq=2 ttl=64 time=0.325 ms  
64 bytes from 10.0.2.1: icmp_seq=3 ttl=64 time=0.250 ms  
64 bytes from 10.0.2.1: icmp_seq=4 ttl=64 time=0.123 ms  
  
--- 10.0.2.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3083ms  
rtt min/avg/max/mdev = 0.104/0.200/0.325/0.092 ms
```

## Options de la commande ping

Les options de cette commande sont :

```
[root@centos8 ~]# ping --help
ping: invalid option -- '-'

Usage: ping [-aAbBdDfhLn0qrRUvV64] [-c count] [-i interval] [-I interface]
           [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
           [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
           [-w deadline] [-W timeout] [hop1 ...] destination

Usage: ping -6 [-aAbBdDfhLn0qrRUvV] [-c count] [-i interval] [-I interface]
           [-l preload] [-m mark] [-M pmtudisc_option]
           [-N nodeinfo_option] [-p pattern] [-Q tclass] [-s packetsize]
           [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline]
           [-W timeout] destination
```

## 2.2 - netstat -i

Pour visualiser les statistiques réseaux, vous disposez de la commande **netstat** :

```
[root@centos8 ~]# netstat -i
Kernel Interface table
Iface      MTU     RX-OK RX-ERR RX-DRP RX-OVR     TX-OK TX-ERR TX-DRP TX-OVR Flg
ens18      1500    476056     0     0 0       363562     0     0     0 BMRU
lo         65536    10936     0     0 0       10936     0     0     0 LRU
virbr0     1500      0     0     0 0          0     0     0     0 BMU
```

## Options de la commande netstat

Les options de cette commande sont :

```
[root@centos8 ~]# netstat --help
usage: netstat [-vWeenNcCF] [<Af>] -r           netstat {-V|--version|-h|--help}
               netstat [-vWnNcaeol] [<Socket> ...]
               netstat { [-vWeenNac] -I[<Iface>] | [-veenNac] -i | [-cnNe] -M | -s [-6tuw] } [delay]

-r, --route          display routing table
-I, --interfaces=<Iface> display interface table for <Iface>
-i, --interfaces    display interface table
-g, --groups        display multicast group memberships
-s, --statistics   display networking statistics (like SNMP)
-M, --masquerade   display masqueraded connections

-v, --verbose       be verbose
-W, --wide          don't truncate IP addresses
-n, --numeric       don't resolve names
--numeric-hosts    don't resolve host names
--numeric-ports    don't resolve port names
--numeric-users    don't resolve user names
-N, --symbolic     resolve hardware names
-e, --extend         display other/more information
-p, --programs      display PID/Program name for sockets
-o, --timers        display timers
-c, --continuous    continuous listing

-l, --listening    display listening server sockets
-a, --all           display all sockets (default: connected)
-F, --fib            display Forwarding Information Base (default)
-C, --cache          display routing cache instead of FIB
-Z, --context        display SELinux security context for sockets

<Socket>={-t|--tcp} {-u|--udp} {-U|--udplite} {-S|--sctp} {-w|--raw}
           {-x|--unix} --ax25 --ipx --netrom
<AF>=Use '-6|-4' or '-A <af>' or '--<af>'; default: inet
List of possible address families (which support routing):
```

```
inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
x25 (CCITT X.25)
```

## 2.3 - traceroute

La commande ping est à la base de la commande **traceroute**. Cette commande sert à découvrir la route empruntée pour accéder à un site donné :

```
[root@centos8 ~]# traceroute www.ittraining.network
bash: traceroute: command not found...
Install package 'traceroute' to provide command 'traceroute'? [N/y] y

* Waiting in queue...
The following packages have to be installed:
traceroute-3:2.1.0-6.el8.x86_64          Traces the route taken by packets over an IPv4/IPv6 network
Proceed with changes? [N/y] y

* Waiting in queue...
* Waiting for authentication...
* Waiting in queue...
* Downloading packages...
* Requesting data...
* Testing changes...
* Installing packages...
traceroute to www.ittraining.network (109.228.56.52), 30 hops max, 60 byte packets
1 _gateway (10.0.2.1)  0.132 ms  0.101 ms  0.078 ms
2 79.137.68.252 (79.137.68.252)  0.542 ms  0.656 ms  0.809 ms
3 10.50.24.61 (10.50.24.61)  0.238 ms  0.219 ms 10.50.24.60 (10.50.24.60)  0.239 ms
4 10.50.0.16 (10.50.0.16)  0.172 ms 10.50.0.22 (10.50.0.22)  0.194 ms  0.173 ms
5 10.73.248.192 (10.73.248.192)  0.766 ms 10.73.248.194 (10.73.248.194)  0.730 ms 10.73.248.192 (10.73.248.192)
0.757 ms
```

```
6 waw-wa2-sbb1-nc5.pl.eu (91.121.131.150) 1.102 ms 1.396 ms 1.099 ms
7 fra-fr5-sbb1-nc5.de.eu (213.251.128.113) 18.309 ms fra-fr5-sbb2-nc5.de.eu (54.36.50.116) 21.881 ms fra-fr5-
sbb1-nc5.de.eu (213.251.128.113) 16.764 ms
8 10.200.0.6 (10.200.0.6) 20.922 ms 10.200.0.0 (10.200.0.0) 16.959 ms 10.200.0.4 (10.200.0.4) 21.143 ms
9 decix.bb-a.fra3.fra.de.oneandone.net (80.81.192.123) 18.789 ms decix.bb-c.act.fra.de.oneandone.net
(80.81.193.123) 20.310 ms decix.bb-a.fra3.fra.de.oneandone.net (80.81.192.123) 18.693 ms
10 ae-14.bb-b.fr7.fra.de.oneandone.net (212.227.120.149) 22.222 ms 22.206 ms 22.257 ms
11 port-channel-3.gw-ngcs-1.dc1.con.glo.gb.oneandone.net (88.208.255.131) 39.660 ms 39.679 ms ae-19.bb-
b.thn.lon.gb.oneandone.net (212.227.120.33) 33.973 ms
12 109.228.63.209 (109.228.63.209) 37.363 ms port-channel-3.gw-ngcs-1.dc1.con.glo.gb.oneandone.net
(88.208.255.131) 39.534 ms 109.228.63.209 (109.228.63.209) 37.901 ms
13 * 109.228.63.209 (109.228.63.209) 38.014 ms 37.991 ms
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 * * *^C
```

## Options de la commande traceroute

Les options de cette commande sont :

```
[root@centos8 ~]# traceroute --help
```

Usage:

```
traceroute [ -46dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i device ] [ -m max_ttl ] [ -N squeries ] [ -p port ] [ -t tos ] [ -l flow_label ] [ -w MAX,HERE,NEAR ] [ -q nqueries ] [ -s src_addr ] [ -z sendwait ] [ --fwmark=num ] host [ packetlen ]
```

Options:

|                                    |  |
|------------------------------------|--|
| -4                                 | Use IPv4   |
| -6                                 | Use IPv6   |
| -d --debug                         | Enable socket level debugging  |
| -F --dont-fragment                 | Do not fragment packets  |
| -f first_ttl --first=first_ttl     | Start from the first_ttl hop (instead from 1)  |
| -g gate,... --gateway=gate,...     | Route packets through the specified gateway<br>(maximum 8 for IPv4 and 127 for IPv6)   |
| -I --icmp                          | Use ICMP ECHO for tracerouting   |
| -T --tcp                           | Use TCP SYN for tracerouting (default port is 80)  |
| -i device --interface=device       | Specify a network interface to operate with  |
| -m max_ttl --max-hops=max_ttl      | Set the max number of hops (max TTL to be reached). Default is 30  |
| -N squeries --sim-queries=squeries | Set the number of probes to be tried simultaneously (default is 16)  |
| -n                                 | Do not resolve IP addresses to their domain names  |
| -p port --port=port                | Set the destination port to use. It is either initial udp port value for "default" method (incremented by each probe, default is 33434), or initial seq for "icmp" (incremented as well, default from 1), or some constant destination port for other methods (with default of 80 for "tcp", 53 for "udp", etc.) |
| -t tos --tos=tos                   | Set the TOS (IPv4 type of service) or TC (IPv6   |

```
traffic class) value for outgoing packets
-l flow_label --flowlabel=flow_label
    Use specified flow_label for IPv6 packets
-w MAX,HERE,NEAR --wait=MAX,HERE,NEAR
    Wait for a probe no more than HERE (default 3)
    times longer than a response from the same hop,
    or no more than NEAR (default 10) times than some
    next hop, or MAX (default 5.0) seconds (float
    point values allowed too)
-q nqueries --queries=nqueries
    Set the number of probes per each hop. Default is
    3
-r
    Bypass the normal routing and send directly to a
    host on an attached network
-s src_addr --source=src_addr
    Use source src_addr for outgoing packets
-z sendwait --sendwait=sendwait
    Minimal time interval between probes (default 0).
    If the value is more than 10, then it specifies a
    number in milliseconds, else it is a number of
    seconds (float point values allowed too)
-e --extensions
    Show ICMP extensions (if present), including MPLS
-A --as-path-lookups
    Perform AS path lookups in routing registries and
    print results directly after the corresponding
    addresses
-M name --module=name
    Use specified module (either builtin or external)
    for traceroute operations. Most methods have
    their shortcuts (`-I' means '-M icmp' etc.)
-O OPTS,... --options=OPTS,....
    Use module-specific option OPTS for the
    traceroute module. Several OPTS allowed,
    separated by comma. If OPTS is "help", print info
    about available options
--sport=num
    Use source port num for outgoing packets. Implies
```

```
          ` -N 1'
--fwmark=num      Set firewall mark for outgoing packets
-U   --udp        Use UDP to particular port for tracerouting
                  (instead of increasing the port per each probe),
                  default port is 53
-UL              Use UDPLITE for tracerouting (default dest port
                  is 53)
-D   --dccp       Use DCCP Request for tracerouting (default port
                  is 33434)
-P prot  --protocol=prot  Use raw packet of protocol prot for tracerouting
--mtu            Discover MTU along the path being traced. Implies
                  ` -F -N 1'
--back           Guess the number of hops in the backward path and
                  print if it differs
-V   --version    Print version info and exit
--help           Read this help and exit
```

**Arguments:**

|           |  |
|-----------|--|
| + host    | The host to traceroute to  |
| packetlen | The full packet length (default is the length of an IP header plus 40). Can be ignored or increased to a minimal allowed value |

## LAB #3 - Connexions à Distance

### 3.1 - Telnet



**Important** - Si la commande **telnet** n'est pas installée sous CentOS 8, installez-le à l'aide de la commande **dnf install telnet** en tant que root.

La commande **telnet** est utilisée pour établir une connexion à distance avec un serveur telnet :

```
# telnet numero_ip
```



**Important** - Le service telnet revient à une redirection des canaux standards d'entrée et de sortie. Notez que la connexion n'est **pas** sécurisée. Pour fermer la connexion, il faut saisir la commande **exit**. La commande telnet n'offre pas de services de transfert de fichiers. Pour cela, il convient d'utiliser la commande **ftp**.

## Options de la commande telnet

Les options de cette commande sont :

```
[[root@centos8 ~]# telnet --help
telnet: invalid option -- '-'
Usage: telnet [-4] [-6] [-8] [-E] [-L] [-S tos] [-a] [-c] [-d] [-e char] [-l user]
           [-n tracefile] [-b hostalias] [-r]
           [host-name [port]]
```

## 3.2 - wget

La commande **wget** est utilisée pour récupérer un fichier via http, https ou ftp :

```
[root@centos8 ~]# wget https://www.dropbox.com/s/wk79lkfr6f12u9j/wget_file.txt
...
[root@centos8 ~]# cat wget_file.txt
This is a file retrieved by the wget command.
```

## Options de la commande wget

Les options de cette commande sont :

```
[root@centos8 ~]# wget --help
GNU Wget 1.19.5, a non-interactive network retriever.
Usage: wget [OPTION]... [URL]...
```

Mandatory arguments to long options are mandatory for short options too.

### Startup:

|                       |                                      |
|-----------------------|--------------------------------------|
| -V, --version         | display the version of Wget and exit |
| -h, --help            | print this help                      |
| -b, --background      | go to background after startup       |
| -e, --execute=COMMAND | execute a `wgetrc'-style command     |

### Logging and input file:

|                          |   |
|--------------------------|---|
| -o, --output-file=FILE   | log messages to FILE                                      |
| -a, --append-output=FILE | append messages to FILE                                   |
| -d, --debug              | print lots of debugging information                       |
| -q, --quiet              | quiet (no output)   |
| -v, --verbose            | be verbose (this is the default)                          |
| -nv, --no-verbose        | turn off verboseness, without being quiet                 |
| --report-speed=TYPE      | output bandwidth as TYPE. TYPE can be bits                |
| -i, --input-file=FILE    | download URLs found in local or external FILE             |
| --input-metalink=FILE    | download files covered in local Metalink FILE             |
| -F, --force-html         | treat input file as HTML                                  |
| -B, --base=URL           | resolves HTML input-file links (-i -F)<br>relative to URL |
| --config=FILE            | specify config file to use                                |
| --no-config              | do not read any config file                               |
| --rejected-log=FILE      | log reasons for URL rejection to FILE                     |

**Download:**

|                              |  |
|------------------------------|--|
| -t, --tries=NUMBER           | set number of retries to NUMBER (0 unlimits)                                 |
| --retry-connrefused          | retry even if connection is refused  |
| --retry-on-http-error=ERRORS | comma-separated list of HTTP errors to retry                                 |
| -O, --output-document=FILE   | write documents to FILE  |
| -nc, --no-clobber            | skip downloads that would download to<br>existing files (overwriting them)   |
| --no-netrc                   | don't try to obtain credentials from .netrc                                  |
| -c, --continue               | resume getting a partially-downloaded file                                   |
| --start-pos=OFFSET           | start downloading from zero-based position OFFSET                            |
| --progress=TYPE              | select progress gauge type   |
| --show-progress              | display the progress bar in any verbosity mode                               |
| -N, --timestamping           | don't re-retrieve files unless newer than<br>local                           |
| --no-if-modified-since       | don't use conditional if-modified-since get<br>requests in timestamping mode |
| --no-use-server-timestamps   | don't set the local file's timestamp by<br>the one on the server             |
| -S, --server-response        | print server response  |
| --spider                     | don't download anything  |
| -T, --timeout=SECONDS        | set all timeout values to SECONDS  |
| --dns-timeout=SECS           | set the DNS lookup timeout to SECS   |
| --connect-timeout=SECS       | set the connect timeout to SECS  |
| --read-timeout=SECS          | set the read timeout to SECS   |
| -w, --wait=SECONDS           | wait SECONDS between retrievals  |
| --waitretry=SECONDS          | wait 1..SECONDS between retries of a retrieval                               |
| --random-wait                | wait from 0.5*WAIT...1.5*WAIT secs between retrievals                        |
| --no-proxy                   | explicitly turn off proxy  |
| -Q, --quota=NUMBER           | set retrieval quota to NUMBER  |
| --bind-address=ADDRESS       | bind to ADDRESS (hostname or IP) on local host                               |
| --limit-rate=RATE            | limit download rate to RATE  |
| --no-dns-cache               | disable caching DNS lookups  |
| --restrict-file-names=OS     | restrict chars in file names to ones OS allows                               |
| --ignore-case                | ignore case when matching files/directories                                  |

|                         |  |
|-------------------------|--|
| -4, --inet4-only        | connect only to IPv4 addresses   |
| -6, --inet6-only        | connect only to IPv6 addresses   |
| --prefer-family=FAMILY  | connect first to addresses of specified family,<br>one of IPv6, IPv4, or none  |
| --user=USER             | set both ftp and http user to USER   |
| --password=PASS         | set both ftp and http password to PASS   |
| --ask-password          | prompt for passwords   |
| --use-askpass=COMMAND   | specify credential handler for requesting<br>username and password. If no COMMAND is<br>specified the WGET_ASKPASS or the SSH_ASKPASS<br>environment variable is used. |
| --no-iri                | turn off IRI support   |
| --local-encoding=ENC    | use ENC as the local encoding for IRIs   |
| --remote-encoding=ENC   | use ENC as the default remote encoding   |
| --unlink                | remove file before clobber   |
| --keep-badhash          | keep files with checksum mismatch (append .badhash)  |
| --metalink-index=NUMBER | Metalink application/metalink4+xml metaurl ordinal NUMBER  |
| --metalink-over-http    | use Metalink metadata from HTTP response headers   |
| --preferred-location    | preferred location for Metalink resources  |
| --xattr                 | turn on storage of metadata in extended file attributes  |

**Directories:**

|                               |   |
|-------------------------------|---|
| -nd, --no-directories         | don't create directories                  |
| -x, --force-directories       | force creation of directories             |
| -nH, --no-host-directories    | don't create host directories             |
| --protocol-directories        | use protocol name in directories          |
| -P, --directory-prefix=PREFIX | save files to PREFIX/..                   |
| --cut-dirs=NUMBER             | ignore NUMBER remote directory components |

**HTTP options:**

|                      |  |
|----------------------|--|
| --http-user=USER     | set http user to USER                  |
| --http-password=PASS | set http password to PASS              |
| --no-cache           | disallow server-cached data            |
| --default-page=NAME  | change the default page name (normally |

|  |   |
|--|---|
| -E, --adjust-extension<br>--ignore-length<br>--header=STRING<br>--compression=TYPE<br>--max-redirect<br>--proxy-user=USER<br>--proxy-password=PASS<br>--referer=URL<br>--save-headers<br><br>-U, --user-agent=AGENT<br>--no-http-keep-alive<br>--no-cookies<br>--load-cookies=FILE<br>--save-cookies=FILE<br>--keep-session-cookies<br>--post-data=STRING<br>--post-file=FILE<br>--method=HTTPMethod<br>--body-data=STRING<br>--body-file=FILE<br>--content-disposition<br><br>--content-on-error<br>--auth-no-challenge | this is 'index.html'.)<br>save HTML/CSS documents with proper extensions<br>ignore 'Content-Length' header field<br>insert STRING among the headers<br>choose compression, one of auto, gzip and none. (default: none)<br>maximum redirections allowed per page<br>set USER as proxy username<br>set PASS as proxy password<br>include 'Referer: URL' header in HTTP request<br>save the HTTP headers to file<br>identify as AGENT instead of Wget/VERSION<br>disable HTTP keep-alive (persistent connections)<br>don't use cookies<br>load cookies from FILE before session<br>save cookies to FILE after session<br>load and save session (non-permanent) cookies<br>use the POST method; send STRING as the data<br>use the POST method; send contents of FILE<br>use method "HTTPMethod" in the request<br>send STRING as data. --method MUST be set<br>send contents of FILE. --method MUST be set<br>honor the Content-Disposition header when<br>choosing local file names (EXPERIMENTAL)<br>output the received content on server errors<br>send Basic HTTP authentication information<br>without first waiting for the server's<br>challenge |
|--|---|

#### HTTPS (SSL/TLS) options:

|  |   |
|--|---|
| --secure-protocol=PR<br><br>--https-only<br>--no-check-certificate<br>--certificate=FILE | choose secure protocol, one of auto, SSLv2,<br>SSLv3, TLSv1, TLSv1_1, TLSv1_2 and PFS<br>only follow secure HTTPS links<br>don't validate the server's certificate<br>client certificate file |
|--|---|

|                            |  |
|----------------------------|--|
| --certificate-type=TYPE    | client certificate type, PEM or DER  |
| --private-key=FILE         | private key file   |
| --private-key-type=TYPE    | private key type, PEM or DER   |
| --ca-certificate=FILE      | file with the bundle of CAs  |
| --ca-directory=DIR         | directory where hash list of CAs is stored   |
| --crl-file=FILE            | file with bundle of CRLs   |
| --pinnedpubkey=FILE/HASHES | Public key (PEM/DER) file, or any number<br>of base64 encoded sha256 hashes preceded by<br>'sha256://' and separated by ';', to verify<br>peer against   |
| --ciphers=STR              | Set the priority string (GnuTLS) or cipher list string (OpenSSL) directly.<br>Use with care. This option overrides --secure-protocol.<br>The format and syntax of this string depend on the specific SSL/TLS engine. |
| <b>HSTS options:</b>       |  |
| --no-hsts                  | disable HSTS   |
| --hsts-file                | path of HSTS database (will override default)  |

|                        |   |
|------------------------|---|
| <b>FTP options:</b>    |   |
| --ftp-user=USER        | set ftp user to USER                          |
| --ftp-password=PASS    | set ftp password to PASS                      |
| --no-remove-listing    | don't remove '.listing' files                 |
| --no-glob              | turn off FTP file name globbing               |
| --no-passive-ftp       | disable the "passive" transfer mode           |
| --preserve-permissions | preserve remote file permissions              |
| --retr-symlinks        | when recursing, get linked-to files (not dir) |

|                              |  |
|------------------------------|--|
| <b>FTPS options:</b>         |  |
| --ftps-implicit              | use implicit FTPS (default port is 990)  |
| --ftps-resume-ssl            | resume the SSL/TLS session started in the control connection when<br>opening a data connection |
| --ftps-clear-data-connection | cipher the control channel only; all the data will be in plaintext                             |
| --ftps-fallback-to-ftp       | fall back to FTP if FTPS is not supported in the target server                                 |

**WARC options:**

|                          |  |
|--------------------------|--|
| --warc-file=FILENAME     | save request/response data to a .warc.gz file              |
| --warc-header=STRING     | insert STRING into the warcinfo record                     |
| --warc-max-size=NUMBER   | set maximum size of WARC files to NUMBER                   |
| --warc-cdx               | write CDX index files                                      |
| --warc-dedup=FILENAME    | do not store records listed in this CDX file               |
| --no-warc-compression    | do not compress WARC files with GZIP                       |
| --no-warc-digests        | do not calculate SHA1 digests                              |
| --no-warc-keep-log       | do not store the log file in a WARC record                 |
| --warc-tempdir=DIRECTORY | location for temporary files created by the<br>WARC writer |

#### Recursive download:

|                        |  |
|------------------------|--|
| -r, --recursive        | specify recursive download   |
| -l, --level=NUMBER     | maximum recursion depth (inf or 0 for infinite)                        |
| --delete-after         | delete files locally after downloading them                            |
| -k, --convert-links    | make links in downloaded HTML or CSS point to<br>local files           |
| --convert-file-only    | convert the file part of the URLs only (usually known as the basename) |
| --backups=N            | before writing file X, rotate up to N backup files                     |
| -K, --backup-converted | before converting file X, back up as X.orig                            |
| -m, --mirror           | shortcut for -N -r -l inf --no-remove-listing                          |
| -p, --page-requisites  | get all images, etc. needed to display HTML page                       |
| --strict-comments      | turn on strict (SGML) handling of HTML comments                        |

#### Recursive accept/reject:

|                        |   |
|------------------------|---|
| -A, --accept=LIST      | comma-separated list of accepted extensions |
| -R, --reject=LIST      | comma-separated list of rejected extensions |
| --accept-regex=REGEX   | regex matching accepted URLs                |
| --reject-regex=REGEX   | regex matching rejected URLs                |
| --regex-type=TYPE      | regex type (posix)                          |
| -D, --domains=LIST     | comma-separated list of accepted domains    |
| --exclude-domains=LIST | comma-separated list of rejected domains    |
| --follow-ftp           | follow FTP links from HTML documents        |
| --follow-tags=LIST     | comma-separated list of followed HTML tags  |

```
--ignore-tags=LIST           comma-separated list of ignored HTML tags
-H, --span-hosts             go to foreign hosts when recursive
-L, --relative               follow relative links only
-I, --include-directories=LIST list of allowed directories
--trust-server-names         use the name specified by the redirection
                             URL's last component
-X, --exclude-directories=LIST list of excluded directories
-np, --no-parent              don't ascend to the parent directory
```

Email bug reports, questions, discussions to <[bug-wget@gnu.org](mailto:bug-wget@gnu.org)>  
and/or open issues at <https://savannah.gnu.org/bugs/?func=additem&group=wget>.

### 3.3 - ftp



**Important** - Si la commande **ftp** n'est pas installée sous CentOS 8, installez-le à l'aide de la commande **dnf install ftp** en tant que root.

La commande **ftp** est utilisée pour le transfert de fichiers. Une fois connecté, il convient d'utiliser la commande **help** pour afficher la liste des commandes disponibles :

```
ftp> help
Commands may be abbreviated.  Commands are:

!      debug      mdir      sendport   site
$      dir        mget      put        size
account  disconnect  mkdir     pwd       status
append   exit       mls      quit      struct
ascii    form       mode     quote     system
bell     get        modtime   recv      sunique
binary   glob       mput     reget     tenex
bye     hash       newer    rstatus   tick
```

```
case      help      nmap      rhelp      trace
cd       idle      nlist      rename      type
cdup     image      ntrans      reset      user
chmod    lcd       open       restart      umask
close    ls        prompt      rmdir      verbose
cr      macdef    passive      runique      ?
delete   mdelete   proxy      send
ftp>
```

Le caractère ! permet d'exécuter une commande sur la machine cliente

```
ftp> !pwd
/root
```

Pour transférer un fichier vers le serveur, il convient d'utiliser la commande **put** :

```
ftp> put nom_fichier_local nom_fichier_distant
```

Vous pouvez également transférer plusieurs fichiers à la fois grâce à la commande **mput**. Dans ce cas précis, il convient de saisir la commande suivante:

```
ftp> mput nom*.*
```

Pour transférer un fichier du serveur, il convient d'utiliser la commande **get** :

```
ftp> get nom_fichier
```

Vous pouvez également transférer plusieurs fichiers à la fois grâce à la commande **mget** ( voir la commande **mput** ci-dessus ).

Pour supprimer un fichier sur le serveur, il convient d'utiliser la commande **del** :

```
ftp> del nom_fichier
```

Pour fermer la session, il convient d'utiliser la commande **quit** :

```
ftp> quit  
[root@centos7 ~]#
```

## 3.4 - SSH

### Présentation

La commande **ssh** est le successeur et la remplaçante de la commande **rlogin**. Il permet d'établir des connexions sécurisées avec une machine distante. SSH comporte cinq acteurs :

- Le **serveur SSH**
  - le démon sshd, qui s'occupe des authentifications et autorisations des clients,
- Le **client SSH**
  - ssh ou scp, qui assure la connexion et le dialogue avec le serveur,
- La **session** qui représente la connexion courante et qui commence juste après l'authentification réussie,
- Les **clefs**
  - **Couple de clef utilisateur asymétriques** et persistantes qui assurent l'identité d'un utilisateur et qui sont stockés sur disque dur,
  - **Clef hôte asymétrique et persistante** garantissant l'identité du serveur et qui est conservé sur disque dur
  - **Clef serveur asymétrique et temporaire** utilisée par le protocole SSH1 qui sert au chiffrement de la clé de session,
  - **Clef de session symétrique qui est générée aléatoirement** et qui permet le chiffrement de la communication entre le client et le serveur. Elle est détruite en fin de session. SSH-1 utilise une seule clef tandis que SSH-2 utilise une clef par direction de la communication,
- La **base de données des hôtes connus** qui stocke les clés des connexions précédentes.

SSH fonctionne de la manière suivante pour la mise en place d'un canal sécurisé:

- Le client contacte le serveur sur son port 22,
- Le client et le serveur échangent leur version de SSH. En cas de non-compatibilité de versions, l'un des deux met fin au processus,
- Le serveur SSH s'identifie auprès du client en lui fournissant :
  - Sa clé hôte,
  - Sa clé serveur,

- Une séquence aléatoire de huit octets à inclure dans les futures réponses du client,
- Une liste de méthodes de chiffrage, compression et authentification,
- Le client et le serveur produisent un identifiant identique, un haché MD5 long de 128 bits contenant la clé hôte, la clé serveur et la séquence aléatoire,
- Le client génère sa clé de session symétrique et la chiffre deux fois de suite, une fois avec la clé hôte du serveur et la deuxième fois avec la clé serveur. Le client envoie cette clé au serveur accompagnée de la séquence aléatoire et un choix d'algorithmes supportés,
- Le serveur déchiffre la clé de session,
- Le client et le serveur mettent en place le canal sécurisé.

## SSH-1

SSH-1 utilise une paire de clefs de type RSA1. Il assure l'intégrité des données par une  **Contrôle de Redondance Cyclique** (CRC) et est un bloc dit **monolithique**.

Afin de s'identifier, le client essaie chacune des six méthodes suivantes :

- **Kerberos**,
- **Rhosts**,
- **RhostsRSA**,
- Par **clef asymétrique**,
- **TIS**,
- Par **mot de passe**.

## SSH-2

SSH-2 utilise **DSA** ou **RSA**. Il assure l'intégrité des données par l'algorithme **HMAC**. SSH-2 est organisé en trois **couches** :

- **SSH-TRANS** – Transport Layer Protocol,
- **SSH-AUTH** – Authentification Protocol,
- **SSH-CONN** – Connection Protocol.

SSH-2 diffère de SSH-1 essentiellement dans la phase authentification.

Trois méthodes d'authentification :

- Par **clef asymétrique**,
  - Identique à SSH-1 sauf avec l'algorithme DSA,
- **RhostsRSA**,
- Par **mot de passe**.

### Options de la commande

Les options de cette commande sont :

```
[root@centos8 ~]# ssh --help
unknown option -- -
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
```

### Authentification par mot de passe

L'utilisateur fournit un mot de passe au client ssh. Le client ssh le transmet de façon sécurisée au serveur ssh puis le serveur vérifie le mot de passe et l'accepte ou non.

Avantage:

- Aucune configuration de clef asymétrique n'est nécessaire.

Inconvénients:

- L'utilisateur doit fournir à chaque connexion un identifiant et un mot de passe,
- Moins sécurisé qu'un système par clef asymétrique.

## Authentification par clef asymétrique

- Le **client** envoie au serveur une requête d'authentification par clé asymétrique qui contient le module de la clé à utiliser,
- Le **serveur** recherche une correspondance pour ce module dans le fichier des clés autorisés **~/.ssh/authorized\_keys**,
  - Dans le cas où une correspondance n'est pas trouvée, le serveur met fin à la communication,
  - Dans le cas contraire le serveur génère une chaîne aléatoire de 256 bits appelée un **challenge** et la chiffre avec la **clé publique du client**,
- Le **client** reçoit le challenge et le décrypte avec la partie privée de sa clé. Il combine le challenge avec l'identifiant de session et chiffre le résultat. Ensuite il envoie le résultat chiffré au serveur.
- Le **serveur** génère le même haché et le compare avec celui reçu du client. Si les deux hachés sont identiques, l'authentification est réussie.

## Configuration du Serveur

La configuration du serveur s'effectue dans le fichier **/etc/ssh/sshd\_config** :

```
[root@centos8 ~]# cat /etc/ssh/sshd_config
#       $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

# If you want to change the port on a SELinux system, you have to tell
```

```
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# This system is following system-wide crypto policy. The changes to
# crypto properties (Ciphers, MACs, ...) will not have any effect here.
# They will be overridden by command-line options passed to the server
# on command line.
# Please, check manual pages for update-crypto-policies(8) and sshd_config(5).

# Logging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

```
#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes
```

```
# GSSAPI options
GSSAPIAuthentication yes
GSSAPICleanupCredentials no
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
#GSSAPIEnablek5users no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
# WARNING: 'UsePAM no' is not supported in Fedora and may cause several
# problems.
UsePAM yes

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes

# It is recommended to use pam_motd in /etc/pam.d/sshd instead of PrintMotd,
# as it is more configurable and versatile than the built-in version.
PrintMotd no

#PrintLastLog yes
#TCPKeepAlive yes
```

```
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# Accept locale-related environment variables
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS

# override default of no subsystems
Subsystem      sftp      /usr/libexec/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#      X11Forwarding no
#      AllowTcpForwarding no
#      PermitTTY no
#      ForceCommand cvs server
```

Pour ôter les lignes de commentaires dans ce fichier, utilisez la commande suivante :

```
[root@centos8 ~]# cd /tmp ; grep -E -v '^(#|$)' /etc/ssh/sshd_config > sshd_config
[root@centos8 tmp]# cat sshd_config
```

```
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
SyslogFacility AUTHPRIV
PermitRootLogin yes
AuthorizedKeysFile      .ssh/authorized_keys
PasswordAuthentication yes
ChallengeResponseAuthentication no
GSSAPIAuthentication yes
GSSAPICleanupCredentials no
UsePAM yes
X11Forwarding yes
PrintMotd no
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS
Subsystem      sftp      /usr/libexec/openssh/sftp-server
```

Pour sécuriser le serveur ssh, ajoutez ou modifiez les directives suivantes :

```
AllowGroups adm
Banner /etc/issue.net
HostbasedAuthentication no
IgnoreRhosts yes
LoginGraceTime 60
LogLevel INFO
PermitEmptyPasswords no
PermitRootLogin no
PrintLastLog yes
Protocol 2
StrictModes yes
X11Forwarding no
```

Votre fichier ressemblera à celui-ci :

```
[root@centos8 tmp]# vi sshd_config
[root@centos8 tmp]# cat sshd_config
AllowGroups adm
Banner /etc/issue.net
HostbasedAuthentication no
IgnoreRhosts yes
LoginGraceTime 60
LogLevel INFO
PermitEmptyPasswords no
PermitRootLogin no
PrintLastLog yes
Protocol 2
StrictModes yes
X11Forwarding no
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
SyslogFacility AUTHPRIV
AuthorizedKeysFile      .ssh/authorized_keys
PasswordAuthentication yes
ChallengeResponseAuthentication no
GSSAPIAuthentication yes
GSSAPICleanupCredentials no
UsePAM yes
PrintMotd no
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS
Subsystem      sftp    /usr/libexec/openssh/sftp-server
```

Renommez le fichier **/etc/ssh/sshd\_config** en **/etc/ssh/sshd\_config.old** :

```
[root@centos8 tmp]# cp /etc/ssh/sshd_config /etc/ssh/sshd_config.old
```

Copiez le fichier **/tmp/sshd\_config** vers **/etc/ssh/**:

```
[root@centos8 tmp]# cp /tmp/sshd_config /etc/ssh  
cp: overwrite '/etc/ssh/sshd_config'? y
```

Redémarrez le service sshd :

```
[root@centos8 tmp]# systemctl restart sshd  
[root@centos8 tmp]# systemctl status sshd  
● sshd.service - OpenSSH server daemon  
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)  
  Active: active (running) since Mon 2021-08-30 02:17:00 EDT; 11s ago  
    Docs: man:sshd(8)  
          man:sshd_config(5)  
 Main PID: 1042039 (sshd)  
   Tasks: 1 (limit: 23535)  
  Memory: 1.1M  
 CGroup: /system.slice/sshd.service  
         └─1042039 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-  
ctr,aes256-cbc,aes128-gcm@openssh.com,a>  
  
Aug 30 02:17:00 centos8.ittraining.loc systemd[1]: Starting OpenSSH server daemon...  
Aug 30 02:17:00 centos8.ittraining.loc sshd[1042039]: Server listening on 0.0.0.0 port 22.  
Aug 30 02:17:00 centos8.ittraining.loc sshd[1042039]: Server listening on :: port 22.  
Aug 30 02:17:00 centos8.ittraining.loc systemd[1]: Started OpenSSH server daemon.  
[q]
```

Mettez l'utilisateur **trainee** dans le groupe **adm** :

```
[root@centos8 tmp]# groups trainee  
trainee : trainee
```

```
[root@centos8 tmp]# usermod -aG adm trainee
[root@centos8 tmp]# groups trainee
trainee : trainee adm
```

Pour générer les clefs du serveur, saisissez la commande suivante en tant que **root**. Notez que la passphrase doit être **vide**.

```
[root@centos8 tmp]# ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa): /etc/ssh/ssh_host_dsa_key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
SHA256:dywC6jKyIMaTxsaEamz1kmthEmuG18HxmS22qRIC0Yk root@centos8.ittraining.loc
The key's randomart image is:
+---[DSA 1024]---+
| |
| .
| .o . o.+
| E. o.*.. .
| +ooo.o +S o o
| X==++ o o o
| B/=+oo
| Ooo++
| . .o
+---[SHA256]---+
```

De la même façon, il est possible de générer les clefs au format **RSA**, **ECDSA** et **ED25519** :

```
[root@centos8 tmp]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /etc/ssh/ssh_host_rsa_key
Enter passphrase (empty for no passphrase):
```

Enter same passphrase again:

Your identification has been saved in /etc/ssh/ssh\_host\_rsa\_key.

Your public key has been saved in /etc/ssh/ssh\_host\_rsa\_key.pub.

The key fingerprint is:

SHA256:8jXFK50NnoJCz9E7aPKpFYSYCstCPfRsdmlLBTNUnKg root@centos8.ittraining.loc

The key's randomart image is:

+---[RSA 3072]---

```
|   . .==o.      |
|  o oo o+= .    |
| .. oo==+o . +  |
|oo .+E++.+ = *  |
|o.. +.S B * .   |
|.     B + =     |
|       =         |
|       o         |
|       .         |
+---[SHA256]---
```

[root@centos8 tmp]# ssh-keygen -t ecdsa

Generating public/private ecdsa key pair.

Enter file in which to save the key (/root/.ssh/id\_ecdsa): /etc/ssh/ssh\_host\_ecdsa\_key

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /etc/ssh/ssh\_host\_ecdsa\_key.

Your public key has been saved in /etc/ssh/ssh\_host\_ecdsa\_key.pub.

The key fingerprint is:

SHA256:AMqFUJKGqnUEPh/IYda0wnbW1kXK+lnprpHs0o4UMbI root@centos8.ittraining.loc

The key's randomart image is:

+---[ECDSA 256]---

```
|++*+= .o      |
|oX.=o+ o o    |
|o %.B + +    |
|...0.= o .    |
|..E.o . S o  |
|.   . o =    |
```

```
|   . * . |
|   . . . 0 |
|   ..000.. |
+---[SHA256]---+
[root@centos8 tmp]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519): /etc/ssh/ssh_host_ed25519_key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_ed25519_key.
Your public key has been saved in /etc/ssh/ssh_host_ed25519_key.pub.
The key fingerprint is:
SHA256:CtbcN9iXv00PfbHFGf2bEW7iRib0FwRctyqM5hmlhwE root@centos8.ittraining.loc
The key's randomart image is:
+--[ED25519 256]--+
|   E . . . . |
|   . . . . 0 |
|   . . . . +. |
|   o . oB ..o.=|
|   o o S*+=o* *+|
|   . . .o.*o*.+.B|
|   . o o +o++|
|   o =o |
|   . o |
+---[SHA256]---+
```

Les clefs publiques générées possèdent l'extension **.pub**. Les clefs privées n'ont pas d'extension :

```
[root@centos8 tmp]# ls /etc/ssh
moduli      ssh_config.d  sshd_config.old      ssh_host_ecdsa_key.pub  ssh_host_ed25519_key.pub
ssh_host_rsa_key.pub
ssh_config  sshd_config    ssh_host_ecdsa_key  ssh_host_ed25519_key    ssh_host_rsa_key
```

Re-démarrez ensuite le service sshd :

```
[root@centos8 tmp]# systemctl restart sshd.service
[root@centos8 tmp]# systemctl status sshd.service
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-08-30 02:24:57 EDT; 9s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1042204 (sshd)
    Tasks: 1 (limit: 23535)
   Memory: 1.1M
      CGroup: /system.slice/sshd.service
              └─1042204 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes256-cbc,aes128-gcm@openssh.com,a>

Aug 30 02:24:57 centos8.ittraining.loc systemd[1]: Starting OpenSSH server daemon...
Aug 30 02:24:57 centos8.ittraining.loc sshd[1042204]: Server listening on 0.0.0.0 port 22.
Aug 30 02:24:57 centos8.ittraining.loc sshd[1042204]: Server listening on :: port 22.
Aug 30 02:24:57 centos8.ittraining.loc systemd[1]: Started OpenSSH server daemon.
[q]
```

## Configuration du Client

Saisissez maintenant les commandes suivantes en tant que **trainee** :



**Important** - Lors de la génération des clefs, la passphrase doit être **vide**.

```
[root@centos8 tmp]# exit
logout
[trainee@centos8 ~]$ ssh-keygen -t dsa
```

```
Generating public/private dsa key pair.  
Enter file in which to save the key (/home/trainee/.ssh/id_dsa):  
Created directory '/home/trainee/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/trainee/.ssh/id_dsa.  
Your public key has been saved in /home/trainee/.ssh/id_dsa.pub.  
The key fingerprint is:  
SHA256:Qd17X1iR0jk5rL0QBbyVg1hNXkUdTeiFtEpn3rgPKc4 trainee@centos8.ittraining.loc  
The key's randomart image is:  
+---[DSA 1024]---+  
| =o+o.o+0B |  
| o +o=o oo= |  
| . +.+oB+ |  
| o o.&+o. |  
| S o o.*.o |  
| o o o. |  
| . + + |  
| + . o |  
| E . |  
+---[SHA256]---+  
[trainee@centos8 ~]$ ssh-keygen -t rsa  
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/trainee/.ssh/id_rsa): Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/trainee/.ssh/id_rsa.  
Your public key has been saved in /home/trainee/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:BgEjoWQrGCzdJfyZczVZYVoafiHsz9GK5PDWuywG/z0 trainee@centos8.ittraining.loc  
The key's randomart image is:  
+---[RSA 3072]---+  
|o+o++oo .oo*. |  
|=+o.oo . .=B . |
```

```
|=. .o o+... |
| . =.o o... . |
| oS= = o |
| .. = =
| +
| +...E |
| . o+... |
+---[SHA256]---+
[trainee@centos8 ~]$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_ecdsa.
Your public key has been saved in /home/trainee/.ssh/id_ecdsa.pub.
The key fingerprint is:
SHA256:mpBDgsCxP2DqRPkmGvXHpNnvm5B+C17MSiiZKfDjWLk trainee@centos8.ittraining.loc
The key's randomart image is:
+---[ECDSA 256]---+
|o..
|.oo
|.*o ..
|+.++ B
|+o =B + S
|=*oo.* =
|B.* o 0 .
|.= = = o..
|. E o oo+.
+---[SHA256]---+
[trainee@centos8 ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

```
Your identification has been saved in /home/trainee/.ssh/id_ed25519.
Your public key has been saved in /home/trainee/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:JfFxG2mg9feAvFGxoxZ8uSs0N3sXvtYQYYg5iVxzZS4 trainee@centos8.ittraining.loc
The key's randomart image is:
+--[ED25519 256]--+
| ..0*=++=. |
| 0==0+Boo |
| o ooE.0. |
| o 0.= |
| S + ... |
| . .0 |
| . + 0.0 |
| + +.00 |
|     0..0. |
+---[SHA256]---
```

Les clés générées seront placées dans le répertoire **~/.ssh/** :

```
[trainee@centos8 ~]$ ls .ssh
id_dsa  id_dsa.pub  id_ecdsa  id_ecdsa.pub  id_ed25519  id_ed25519.pub  id_rsa  id_rsa.pub
```

## Tunnels SSH

Le protocole SSH peut être utilisé pour sécuriser les protocoles tels telnet, pop3 etc.. En effet, on peut créer un *tunnel* SSH dans lequel passe les communications du protocole non-sécurisé.

La commande pour créer un tunnel ssh prend la forme suivante :

```
ssh -N -f compte@hôte -Lport-local:localhost:port_distant
```

Dans votre cas, vous allez créer un tunnel dans votre propre vm entre le port 15023 et le port 23 :

```
[root@centos8 ~]# ssh -N -f trainee@localhost -L15023:localhost:23
\S
Kernel \r on an \m
trainee@localhost's password: trainee
```

Installez maintenant le serveur telnet :

```
[root@centos8 ~]# dnf install telnet-server
```

Telnet n'est ni démarré ni activé. Il convient donc de le démarrer et de l'activer :

```
[root@centos8 ~]# systemctl status telnet.socket
● telnet.socket - Telnet Server Activation Socket
  Loaded: loaded (/usr/lib/systemd/system/telnet.socket; disabled; vendor preset: disabled)
  Active: inactive (dead)
    Docs: man:telnetd(8)
   Listen: [::]:23 (Stream)
  Accepted: 0; Connected: 0;
```

```
[root@centos8 ~]# systemctl start telnet.socket
```

```
[root@centos8 ~]# systemctl status telnet.socket
● telnet.socket - Telnet Server Activation Socket
  Loaded: loaded (/usr/lib/systemd/system/telnet.socket; disabled; vendor preset: disabled)
  Active: active (listening) since Mon 2021-08-30 02:44:01 EDT; 4s ago
    Docs: man:telnetd(8)
   Listen: [::]:23 (Stream)
  Accepted: 0; Connected: 0;
           CGroup: /system.slice/telnet.socket
```

Aug 30 02:44:01 centos8.ittraining.loc systemd[1]: Listening on Telnet Server Activation Socket.

```
[root@centos8 ~]# systemctl enable telnet.socket
```

```
Created symlink /etc/systemd/system/sockets.target.wants/telnet.socket → /usr/lib/systemd/system/telnet.socket.
```

Connectez-vous ensuite via telnet sur le port 15023, vous constaterez que votre connexion n'aboutit pas :

```
[root@centos8 ~]# telnet localhost 15023
Trying ::1...
Connected to localhost.
Escape character is '^]'.

Kernel 4.18.0-305.7.1.el8.i2tch.x86_64 on an x86_64
centos8 login: trainee
Password:
Last login: Mon Aug 30 02:37:00 from ::1
[trainee@centos8 ~]$ whoami
trainee
[trainee@centos8 ~]$ pwd
/home/trainee
```



**Important** - Notez bien que votre communication telnet passe par le tunnel SSH.

## 3.5 - SCP

### Présentation

La commande **scp** est le successeur et la remplaçante de la commande **rcp** de la famille des commandes **remote**. Il permet de faire des transferts sécurisés à partir d'une machine distante :

```
$ scp compte@numero_ip(nom_de_machine):/chemin_distant/fichier_distant /chemin_local/fichier_local
```

ou vers une machine distante :

```
$ scp /chemin_local/fichier_local compte@numero_ip(nom_de_machine):/chemin_distant/fichier_distant
```

## Utilisation

Nous allons maintenant utiliser **scp** pour chercher un fichier sur le «serveur» :

Créez le fichier **/home/trainee/scp\_test** :

```
[trainee@centos8 ~]$ touch scp-test
[trainee@centos8 ~]$ exit
logout
Connection closed by foreign host.
[root@centos8 ~]#
```

Récupérez le fichier **scp\_test** en utilisant scp :

```
[root@centos8 ~]# scp trainee@127.0.0.1:/home/trainee/scp-test .
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:Q7T/CP0SLiMbMAIgVzTuEHegYS/spPE5zzQchCHD5Vw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '127.0.0.1' (ECDSA) to the list of known hosts.
\S
Kernel \r on an \m
trainee@127.0.0.1's password: trainee
scp-test
100%    0      0.0KB/s   00:00

[root@centos8 ~]# ls -l
total 32
-rw-----. 1 root root 1358 Jun 16 06:40 anaconda-ks.cfg
drwxr-xr-x. 3 root root    21 Jun 16 06:39 home
```

```
-rw-r--r--. 1 root root 1749 Aug 24 11:20 I2TCH.asc
-rw-r--r--. 1 root root 1853 Jun 16 06:54 initial-setup-ks.cfg
-rw-r--r--. 1 root root    31 Aug 24 11:22 message.txt
-rw-r--r--. 1 root root  561 Aug 24 11:32 message.txt.asc
-rw-r--r--. 1 root root  367 Aug 24 11:30 message.txt.gpg
-rw-r--r--. 1 root root  329 Aug 24 11:23 message.txt.sig
-rw-r--r--. 1 root root     0 Aug 30 03:55 scp-test
-rw-r--r--. 1 root root   46 Aug 29 06:22 wget_file.txt
```

### 3.6 - Mise en Place des Clefs Asymétriques

Il convient maintenant de se connecter sur le «serveur» en utilisant ssh et vérifiez la présence du répertoire `~/.ssh` :

```
[root@centos8 ~]# ssh -l trainee 127.0.0.1
\S
Kernel \r on an \m
trainee@127.0.0.1's password: trainee
Activate the web console with: systemctl enable --now cockpit.socket
```

```
[trainee@centos8 ~]$ ls -la | grep .ssh
drwx-----. 2 trainee trainee    4096 Aug 30 02:26 .ssh
```



**Important** - Si le dossier distant `.ssh` n'existe pas dans le répertoire personnel de l'utilisateur connecté, il faut le créer avec des permissions de 700. Dans votre cas, puisque votre machine joue le rôle de serveur **et** du client, le dossier `/home/trainee/.ssh` existe déjà.

Ensuite, il convient de transférer le fichier local `.ssh/id_ecdsa.pub` du «client» vers le «serveur» en le renommant en **authorized\_keys** :

```
[trainee@centos8 ~]$ exit
```

```
logout
Connection to 127.0.0.1 closed.
```

```
[root@centos8 ~]# exit
logout
```

```
[trainee@centos8 ~]$ scp .ssh/id_ecdsa.pub trainee@127.0.0.1:/home/trainee/.ssh/authorized_keys
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:Q7T/CP0SLiMbMAIgVzTuEHegYS/spPE5zzQchCHD5Vw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '127.0.0.1' (ECDSA) to the list of known hosts.
\S
Kernel \r on an \m
trainee@127.0.0.1's password: trainee
id_ecdsa.pub
100% 192 497.6KB/s 00:00
```

Connectez-vous via telnet :

```
[trainee@centos8 ~]$ ssh -l trainee localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:Q7T/CP0SLiMbMAIgVzTuEHegYS/spPE5zzQchCHD5Vw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
\S
Kernel \r on an \m
Activate the web console with: systemctl enable --now cockpit.socket
```

```
Last login: Mon Aug 30 03:57:14 2021 from 127.0.0.1
[trainee@centos8 ~]$
```



**Important** - Lors de la connexion au serveur, l'authentification utilise le couple de clefs



asymétrique au format ecdsa et aucun mot de passe n'est requis.

Insérez maintenant les clefs publiques restantes dans le fichier .ssh/authorized\_keys :

```
[trainee@centos8 ~]$ cd .ssh
[trainee@centos8 .ssh]$ ls
authorized_keys  id_dsa  id_dsa.pub  id_ecdsa  id_ecdsa.pub  id_ed25519  id_ed25519.pub  id_rsa  id_rsa.pub
known_hosts
[trainee@centos8 .ssh]$ cat authorized_keys
ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmldHAyNTYAAABBBHDrzSXP+Ecx/sQ18VwCRNm7rrSrrsaJmuIw/RgTH5puKF5E+Yy15cvAAKBX
pJPxUmr0a0yhab84PevV7XSHcI= trainee@centos8.ittraining.loc

[trainee@centos8 .ssh]$ cat id_rsa.pub >> authorized_keys
[trainee@centos8 .ssh]$ cat id_dsa.pub >> authorized_keys
[trainee@centos8 .ssh]$ cat id_ed25519.pub >> authorized_keys

[trainee@centos8 .ssh]$ cat authorized_keys
ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmldHAyNTYAAABBBHDrzSXP+Ecx/sQ18VwCRNm7rrSrrsaJmuIw/RgTH5puKF5E+Yy15cvAAKBX
pJPxUmr0a0yhab84PevV7XSHcI= trainee@centos8.ittraining.loc
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQgQD3ZSMn/GIAHtaDFc6ZNnKJam9hzq8TxqMN5IopUr8Qhw0DyPadbB+FgH4r50qTux4ubwr1BlymgIdqRVWy3
2mE15M8tdtKc3j8DNMpUwPGEh+s/PT7GW+3E3shoyPvpLc1kKaKXKG1/JwfCK/8IYsubk2BmiiJYkzLECotPlaaxm4w1K0AtlnZQuLHt1HK3/rHCh
xo2o2w1t59/QwNcMLiKve1Z+zQ1P0Ko8VJ/DDrf90y2QWC28ejUs/ZjP6f6C4bn5Jjol4TbHls3ArMsbU6C1Ev5jqbzZ0kmognQ2CnRjeNM51YdFo
6nRsLoPKpeLRMBpT/87HK1bPU0BVCyhXxSkqkVhMlgg8tgcD/MRlBaUVFoZ1wQ0L26Fe+q7c20ykj54pdXCAK2ZTpCXGfh/FxrfqGw7cSeKLGX4
QUzHMjMk2oHFC9h30BUk1gGN21KJPTTh7/S10ZVrnCc3GUi5fXPvEpral0IU0sws+j03dj0sWm5ICQFKRkmZN11HCyT0=
trainee@centos8.ittraining.loc
ssh-dss
AAAAB3NzaC1kc3MAAACBALIdwEEqHrMWSDzARm9ldsZK9ebbtZShtmwgdjph0k77fxymK0y6wV7QEmLL25L0cLb12uZ1F0LtRt/t2oqgrwqk3vUS
pCPLr09AXpcD/nxL9kc+rUxHyl6u1mHtyfCVLCPsavCMR8TaA8egVMk3EwGrfHTiuDOKi7Iwus7gXPAAAQDHEQPGVRI7gVYKzCT6nrjDsQ06jw
AAAIEAhHH7fEjdldASXY0qTwkCvcs3cfK9/Ff315zByn47002y9Vdo3QG5n0r10o8fc2xEkIBNmFr8Rr2g60cpvEev5hy4XZ1ghxnQ53iwKuiS72
ZATwhD6bZBrsiH0k1Et25gRcj5KCvDe/jHhbxCxsCuHUH2qvWsQNVwztE7hD0sxkAACAQ8Dkpy8zXj7jW8o1txxf2W6J4r2+1lPldynA45ywZokN
```

4SCwvXlpPAuyBt0/Hiu0R2PI9aq0AMosCLcy9WmnSwLQ2Z7QcD2i3XlAih2+1q9NJP22sPT3jSK9UZcdRjoZ/eNiz84sXZucNape32tFxjvcV4txo  
bH/vD53q8g63fA= trainee@centos8.ittraining.loc  
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOffQULLU8IZyKiSU63D2Zz6yGLqyHcBHnCRdSR9JSmc trainee@centos8.ittraining.loc

Copyright © 2024 Hugh Norris.

From:  
<https://ittraining.team/> - **www.ittraining.team**

Permanent link:  
<https://ittraining.team/doku.php?id=elearning:workbooks:centos:8:avance:l113>

Last update: **2024/10/10 14:21**

