Rappels sur les réseaux TCP/IP

1. Éléments d'un réseau

Un réseau comprend :

- **Périphériques** : ordinateurs, routeurs, serveurs…
- **Messages** : données échangées (web, mails…)
- **Protocoles** : règles de communication (ex. TCP/IP)
- **Supports** : câbles, fibres, Wi-Fi, etc.

2. Périphériques

- **Utilisateurs** : ordinateurs, smartphones…
- **Infrastructure** : routeurs, commutateurs
- **Serveurs** : fournissent des services (web, mail…)

3. Messages

Exemples : HTTP (web), SMTP (mails), VoIP (Skype), messagerie instantanée.

4. Supports de transmission

- Câbles Ethernet, fibres optiques, ondes radio (Wi-Fi, 4G…).

5. Protocoles

- **TCP/IP** : fondement d'Internet
- **HTTP/HTTPS** : web
- **SMTP/IMAP/POP3** : mails
- **DNS** : noms de domaine → IP
- **FTP** : transferts de fichiers

6. Modèles de communication

- **Modèle OSI** (7 couches)
- **Modèle TCP/IP** (4 couches : Application, Transport, Réseau, Liaison)

7. Encapsulation

Chaque couche ajoute ses infos :

- Application (HTTP, DNS…)
- Transport (TCP/UDP)
- Réseau (IP)
- Liaison (Ethernet…)

8. Protocoles de la couche Application

- DNS, HTTP, SMTP, Telnet, FTP…

9. Client/Serveur

- Client fait une requête, serveur répond.
- Exemple : téléchargement de fichier.
- **Telnet** : protocole d'accès distant via port 23.

10. Peer-to-Peer (P2P)

- Pas de serveur central.
- Chaque hôte est client et serveur.
- Exemple : partage de fichiers BitTorrent, imprimante partagée.

11. HTTP

- Affichage des pages web.
- Port 80 (443 pour HTTPS).

12. FTP

- Transfert de fichiers.
- Port 21 (contrôle) & 20 (transfert).
- Commandes : `get`, `put`.

13. DNS

- Nom → IP.
- Port 53 TCP/UDP.

14. DHCP

- Attribution automatique d'adresse IP.
- Port UDP 67.
- Étapes : Discover, Offer, Request, Ack.

15. Exemple de dépannage réseau

- Si le site est accessible par IP mais pas par nom : problème DNS.

Remarque :

Le **hub** diffuse les données à tous les appareils, le **switch** les envoie uniquement au destinataire, et le **routeur** connecte différents réseaux entre eux.

Couche Transport

1 Modèle TCP/IP

Le modèle TCP/IP comporte 4 couches : Application, Transport, Internet, Accès réseau.
La **couche transport** permet de transmettre les données entre les applications des différents périphériques.

2 Objectifs de la couche transport

Elle permet :

- Le suivi de plusieurs conversations entre applications
- La segmentation et le réassemblage des données
- L'identification via les ports
- La fiabilité, le contrôle de flux, et la livraison ordonnée

3 Couche transport du modèle TCP/IP

TCP est utilisé pour les services fiables (web, e-mail), UDP pour ceux rapides et tolérants à la perte (VoIP, streaming).
Elle connecte les clients et serveurs selon leurs besoins.

4 Différences entre TCP et UDP

- **TCP** : fiable, avec connexion, ordonné, plus lent
- **UDP** : rapide, sans connexion, pas de garantie d'ordre ou de livraison

5 Numéros de port TCP et UDP

Les ports identifient les applications :

- 0-1023 : ports réservés (HTTP, DNS…)
- 1024-49151 : ports inscrits (MSN, VoIP…)
- 49152-65535 : ports dynamiques

6 Protocole TCP

Protocole fiable : garantit la réception complète et dans l'ordre des données. Utilise les numéros de séquence, les accusés de réception (ACK) et contrôle d'erreur.

7️⃣ Segmentation des données par TCP

Les données sont divisées en **segments**, chacun avec son **numéro de séquence**, ACK, et champ de contrôle d'erreur pour assurer la fiabilité.

8️⃣ En-tête du protocole TCP

Contient :

- Ports source/destination
- Numéros de séquence et d'ACK
- Taille de fenêtre (contrôle de flux)
- Options, checksum, etc.

9️⃣ Fonctionnement du protocole TCP

Chaque service utilise un port. Le client utilise un port source temporaire et un port destination fixe. Le serveur répond en inversant les rôles des ports.

🔟 TCP est un protocole avec connexion

TCP établit une connexion **avant** d'envoyer les données (SYN, SYN-ACK, ACK), et la **termine proprement** (FIN, ACK).

1️⃣1️⃣ Établissement d'une session TCP

Réalisé via le **Three-Way Handshake** :

1. Client envoie SYN
2. Serveur répond avec SYN-ACK
3. Client répond avec ACK

1️⃣2️⃣ Terminaison d'une session TCP

Fermeture via le **Four-Way Handshake** :

1. Client envoie FIN
2. Serveur répond avec ACK
3. Serveur envoie FIN

4. Client répond avec ACK final

13 Autre terminaison

TCP peut aussi être fermé de manière **brutale** avec un segment **RST**, souvent en cas d'erreur ou de rejet de connexion.

14 Ordonnancement des segments TCP

TCP remet les segments dans l'ordre même s'ils arrivent en désordre grâce au **numéro de séquence** et les stocke dans une mémoire tampon.

15 TCP est un protocole fiable

- Il gère les **pertes de segments** en les **retransmettant**.
- Chaque segment non acquitté est conservé puis renvoyé si besoin.

16 TCP permet le contrôle de flux

TCP utilise une **fenêtre d'envoi** pour limiter la quantité de données transmises avant d'avoir un ACK, s'adaptant ainsi à la capacité du récepteur.

17 Protocole UDP

UDP est un protocole **léger et rapide**, sans connexion ni garantie de livraison.
Utilisé pour DNS, VoIP, streaming, jeux en ligne.

18 UDP n'effectue pas d'ordonnancement

Les datagrammes peuvent arriver dans le désordre. UDP ne réorganise rien et ne retransmet pas en cas de perte.

19 Fonctionnement du protocole UDP

Chaque application utilise un port. Les clients envoient des requêtes depuis un port source vers un port destination fixe du serveur, sans établir de connexion.

20 Conclusion

- **TCP** : utilisé si la **fiabilité** est essentielle (web, mail)
- **UDP** : privilégié si la **rapidité** prime (VoIP, streaming) La **couche transport** est donc vitale pour relier correctement les applications via le réseau.

Adressage IPv4

1️⃣ Architecture du réseau

- **LANs** : pour le siège et les succursales (juridique, RH, commerciale).
- **WANs** : pour relier les différents sites (liens point-à-point entre routeurs).

2️⃣ Calcul du nombre total d'hôtes

- Besoin global : **800 hôtes** (interfaces incluses + marge).
- Bloc choisi : **172.16.0.0/22** ➤ 1022 hôtes disponibles (bon compromis).

3️⃣ Détermination des sous-réseaux

- Siège : 58 hôtes ➤ **/26**
- RH : 26 hôtes ➤ **/27**
- Juridique et Commercial : 10 hôtes ➤ **/28** chacun

4️⃣ Attribution des adresses

- Utilisation de **masques variables** (VLSM) pour optimiser.
- Exemple :
    - Siège : 172.16.0.0/26 (172.16.0.1 à 0.62)
    - RH : 172.16.0.64/27 (172.16.0.65 à 0.94)
    - WANs : chaque lien ➤ /30 (2 hôtes par lien)

5️⃣ Inconvénients du schéma standard

- Taille fixe des sous-réseaux ➤ gaspillage d'adresses
- Peu flexible ➤ nécessite VLSM

6️⃣ VLSM (Variable Length Subnet Masking)

- Permet de **varier les tailles de sous-réseaux** selon les besoins.
- Optimise l'espace IP en évitant les pertes.

7️⃣ Procédure de VLSM

1. Attribuer le plus grand besoin en premier.
2. Réserver un sous-réseau adapté.
3. Répéter pour les besoins restants, du plus grand au plus petit.

8️⃣ Exemple complet de VLSM

- 4 LANs + 3 WANs planifiés précisément.
- Utilisation efficace de **172.16.0.0/22**
- Plages libres prévues pour **extensions futures**

9️⃣ Tableau récapitulatif

- Montre chaque sous-réseau avec :
    - Plage IP
    - Masque
    - Adresses utilisables
- Exemple : WAN 1 ➤ 172.16.0.128/30 ➤ 2 hôtes utilisables

🔟 Pourquoi cette solution ?

✅ Aucune perte d'adresses
✅ Évolutif (WAN 4 ou autres LANs possibles)
✅ Masques adaptés à chaque besoin

1️⃣1️⃣ Dépannage de la couche réseau

🔧 Protocole ICMP

- Utilisé pour signaler les erreurs réseau et tester la connectivité.

🔧 Commande Ping

- Envoie un **Echo Request**
- Attend un **Echo Reply**

- Permet de savoir si un hôte est joignable et de mesurer la latence.

🔧 **Commande Traceroute (Tracert)**

- Affiche chaque **étape (routeur)** traversée jusqu'à la destination.
- Utile pour **localiser les coupures** ou les lenteurs réseau.