LES BASES DU TRAVAIL EN RESEAU

SOMMAIRE

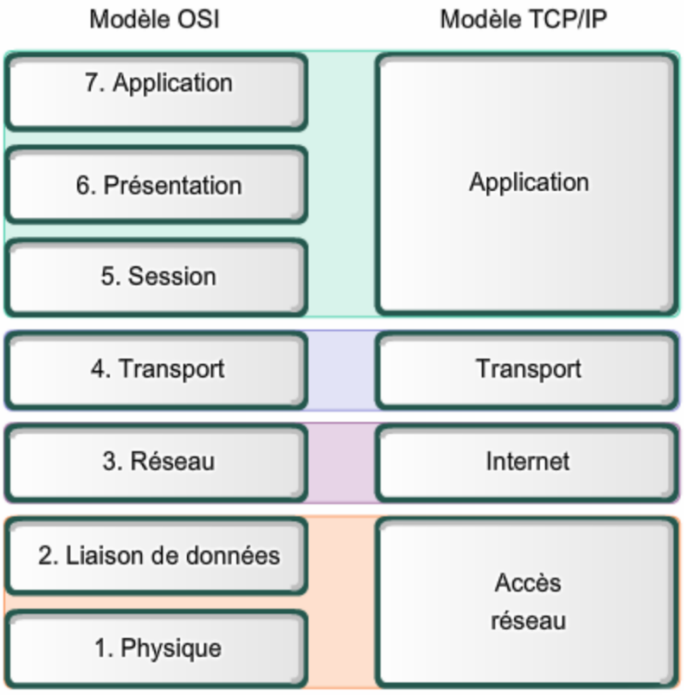
Table des matières

LES BASES DU TRAVAIL EN RESEAU

Introduction au travail en réseau

Sommaire

- Les ordinateurs, les appareils mobiles et les serveurs sont tous reliés entre eux via un vaste réseau.
- Le réseau fonctionne avec une organisation en couches. 3 couches pour faire simple :
  - ⇒ Couche réseau
  - ⇒ Couche TCP/IP
  - ⇒ Couche applicative (web)
- Les couches utilisent des protocoles avec lesquels on interagit en tant que développeur Web.
- Modèle OSI à 7 couches et modèle TCP / IP équivalent :



Les couches basses (1, 2)

- Le réseau est construit comme une série de couches.

  ⇒ **1 : la couche physique :** elle définit les protocoles d'échange d'informations de très bas niveau (les normes d'encodage et de signalisation des données).

  ⇒ **2 : la couche de liaison de données :** elle contient des protocoles tels que Ethernet et Wi-Fi, et définit davantage de protocoles qui fournissent le cadrage des données, l'adressage (comment nous identifions un périphérique dans le réseau), la détection des erreurs, etc.

- Ces tâches de couches doivent s'assurer que la connexion se produit, et elles offrent également une fiabilité élevée et gèrent la redondance et le basculement. Le réseau Internet a été initialement conçu comme un outil militaire, conçu pour être fiable même si plusieurs nœuds du réseau pouvaient tomber en panne ou être détruits.

Les couches intermédiaires (3, 4) : TCP / IP , UDP

- À partir de là, les niveaux supérieurs se superposent pour fournir des fonctionnalités de niveau supérieur.
- Le Web (les couche hautes) repose sur une pile de protocoles communément appelés TCP / IP .
- Il s'agit d'un terme générique qui identifie plusieurs protocoles différents qui fonctionnent à différents niveaux.

  ⇒ **3 : la couche réseau :** elle définit les protocole IP.

  ⇒ **4 : la couche transport :** on y trouve le protocole TCP (Transmission Control Protocol) et aussi UDP (User Datagram Protocol).

Les couches hautes (5, 6, 7) : HTTP, FTP

- En plus des protocoles TCP et UDP, on trouve les protocoles de couche Application :

  ⇒ HTTP (Hyper Text Transfer Protocol)

  ⇒ Sa version sécurisée HTTPS

  ⇒ FTP

  ⇒ Websockets

  ⇒ etc.

RFC

Présentation

- RFC = Request for Comments = documents décrivant les aspects et spécifications techniques d'Internet.
- https://fr.wikipedia.org/wiki/Request_for_comments
- Tous les protocoles dont nous parlerons sont définis dans les RFC.

Liste des RFC

- https://fr.wikipedia.org/wiki/Liste_de_RFC
- Les plus connus :
- RFC 791: IP : https://fr.wikipedia.org/wiki/Internet_Protocol
- RFC 793: TCP : https://fr.wikipedia.org/wiki/Transmission_Control_Protocol
- RFC 1034: DNS : https://tools.ietf.org/html/rfc1034
- RFC 4291: IPv6 : https://tools.ietf.org/html/rfc4291
- RFC 6749: OAuth 2.0 : https://tools.ietf.org/html/rfc6749

Couche IP et adresse IP

Couche IP

- IP : Internet Protocol
- La couche IP correspond à la couche 3 : c'est la couche réseau.
- Chaque nœud du réseau est identifié par une adresse, que nous appelons l'adresse IP.
- L'objectif est d'envoyer des données d'un ordinateur à un autre, dans un réseau d'ordinateur.
- Les données sont coupées par l'expéditeur en de nombreux petits fragments IP appelé paquet IP. Leur numérotation permet au récepteur de reconstruire le message d'origine.
- Les paquets IP sont envoyés par l'expéditeur et ils peuvent emprunter différentes routes pour atteindre la destination. Cela ne dépend ni de l'expéditeur, ni du récepteur. Le réseau Internet fonctionne de telle sorte que si un itinéraire échoue, d'autres itinéraires peuvent être utilisés pour atteindre la destination.

Adresse IP

- Dans un réseau TCP / IP, une adresse IP identifie de manière unique un nœud du réseau.
- Les adresses IPv4 sont contiennent 32 bits, les adresses IPv6 128.
- Une adresse IPv4 est composée de 4 parties de 8 bits. Chacune de ces parties est identifiée par un nombre de 0 à 255, chacun séparé par un point. Par exemple :

  ⇒ 212.21.4.28

  ⇒ 8.8.8.8

  ⇒ 121.2.1.2
- Avec 32 bits, il existe 2 puissance 32 = 4 milliards d'identifiants : un nombre qui s'avère trop petit !
- À l'origine, chaque ordinateur avait sa propre adresse IP unique.
- Aujourd'hui, chaque fois que vous vous connectez à un réseau avec votre ordinateur ou votre téléphone, vous obtenez une adresse IP aléatoire. Cette adresse IP est locale, ce qui signifie que dans la plupart des cas, vous ne pouvez pas être atteint de l'extérieur du réseau construit par votre routeur domestique ou professionnel.
- Certains fournisseurs proposent des adresses IP statiques, qui ne changent jamais, qui peuvent être utilisées pour atteindre votre réseau de l'extérieur, mais il s'agit généralement d'un service premium.
- Il existe des plages spéciales d'adresses IP réservées à l'utilisation d'un LAN. Le plus courant est 192.168.xxx.xxx.
- Depuis n'importe quel ordinateur, vous pouvez accéder à lui-même en utilisant l'adresse spéciale 127.0.0.1. ou localhost (c'est le nom du domaine)

DNS : système de nom de domaine

- On n'essaye généralement pas d'accéder à un site Web en utilisant son adresse IP.
- On peut mais ce n’est pas pratique.
- On utilise généralement un nom de domaine comme google.com ou bliaudet.free.fr
- Le DNS, Domain Name System, est le système qui mappe (qui fait correspondre) les noms de domaine aux adresses IP. Ce système est un serveur : un logiciel qui répond à des demandes.
- Votre fournisseur aura son propre DNS : votre routeur est déjà préconfiguré pour l'utiliser.
- Ces serveurs DNS recevront les requêtes de votre ordinateur, et à leur tour demanderont à leur propre serveur DNS de référence.
- L’organisation des serveurs DNS est un arbre. Il y a un serveur DNS tout en haut, appelé serveur DNS racine. En première approximation, on peut se dire que le serveur DNS racine connaît l'adresse IP des serveurs DNS qui gèrent chaque extension de domaine, comme com, net, orget ainsi de suite et va donc pouvoir leur demander de mapper le nom de domaine fourni à une adresse IP.

Le protocole TCP

- La couche TCP correspond à la couche 4 : c'est la couche transport.
- TCP signifie Transmission Control Protocol.
- C'est la base du Web : le transport de données d'un ordinateur à un autre.
- C'est aussi la base et d'autres applications comme le courrier électronique.
- Défini dans la RFC 793 en 1981, TCP est l'un des plus anciens piliers de l'Internet.
- TCP, contrairement à IP et UDP, est orienté « connexion » ce qui veut dire que : avant que la transmission puisse avoir lieu via TCP, une connexion doit être établie. Ensuite les données sont envoyées, sous forme de petits paquets, et lorsque la communication se termine, la connexion est fermée.
- Une connexion c'est un « handshake » : une poignée de main entre l'émetteur et le récepteur ce qui permet de toujours savoir si un paquet envoyé par l'expéditeur a été correctement reçu par le destinataire. Ainsi, si un paquet est perdu, le protocole est capable de le gérer et le paquet est renvoyé.
- Avec le protocole IP, les connexions se font d'un ordinateur à l'autre. Avec TCP, une connexion utilise le concept de ports. Un port, associé à une adresse IP, permet d'identifier de manière unique un processus sur un ordinateur. Par exemple :

```
localhost:8080
google.com:1234
```

- Chaque protocole d'application a un port par défaut :
  - HTTP 80
  - HTTPS 443
  - FTP 21
- C'est pourquoi on généralement pas besoin de spécifier le port dans le navigateur.
- Les programmes ne sont pas obligés d'utiliser la valeur par défaut.
- Les numéros de port vont de 1 à 65535 (codé sur 2 octets).

Le protocole UDP

- UDP, User Datagram Protocol, est une alternative à TCP.
- Sa principale différence avec TCP est qu'il est sans connexion.
- Avantage : il est plus rapide car chaque paquet envoyé est plus léger, car il ne contient pas toutes les informations nécessaires à la validation.
- Inconvénient : il n’est pas fiable comme TCP. Avec TCP, si un paquet est perdu, le protocole est capable de le gérer et le paquet est renvoyé. Avec UDP, ce n'est pas intégré au protocole.
- Certains des protocoles d'application les plus notables qui reposent sur la couche UDP sont
- DNS, DHCP, et la couche de base de HTTP / 3 (prochaine version de HTTP) utilise l’UDP.
- UDP peut utiliser des ports pour permettre la communication entre les processus, comme avec TCP.

Le protocole HTTP

Présentation

- HTTP (Hyper Text Transfer Protocol) est un protocole d'application très populaire (couches 5, 6 et 7) qui se place au-dessus du TCP/IP.
- HTTP est ce qui fait fonctionner le World Wide Web, donnant aux navigateurs un langage pour communiquer avec les serveurs distants qui hébergent des pages Web.
- HTTP a été normalisé pour la première fois en 1991, à la suite du travail effectué par Tim Berners-Lee au CERN.
- L'objectif était de permettre aux chercheurs d'échanger et de relier facilement leurs articles. Il s'agissait d'un moyen pour la communauté scientifique de mieux travailler.
- À l'époque, les principales applications Internet consistaient essentiellement en FTP (le protocole de transfert de fichiers), Email et Usenet (groupes de discussion, aujourd'hui presque abandonnés).
- En 1993, Mosaic, le premier navigateur Web graphique, est sorti, et les choses ont explosé à partir de là.
- De là, le Web est devenu l'application tueur d'Internet.
- Au fil du temps, le Web a considérablement évolué, mais les bases demeurent. Un exemple d'évolution : HTTP alimente désormais, en plus des pages Web, les API REST, un moyen courant d'accéder par programmation à un service sur Internet.
- Historique
  - HTTP 1991
  - HTTP-1.1 1997 (révision mineure)
  - HTTP-2 2015
  - HTTP-3 en cours de normalisation
- HTTP et HTTP / 2 fonctionnent sur TCP , tandis que HTTP / 3 fonctionne sur QUIC, un protocole construit sur UDP .
- Le protocole HTTP est non sécurisé, comme tout les protocoles (SMTP, FTP ..) non servis par une connexion cryptée.
- HTTPS est la version cryptée de HTTP, servi par TLS.

Documents HTML

- HTTP est la façon dont les navigateurs Web tels que Chrome, Firefox, Edge et bien d'autres (également appelés clients à partir de maintenant) communiquent avec les serveurs Web.
- Le nom Hyper Text Transfer Protocol dérive de la nécessité de transférer non seulement des fichiers, comme dans FTP (F-TP : File-Transfer Protocol), mais des hypertextes écrits en HTML (HT-TP : Hyper Text-Transfer Protocol).
- Les liens ont été le moteur de l'adoption, ainsi que la facilité de création de nouvelles pages Web.
- HTTP est ce qui transfère ces fichiers hypertextes sur le réseau (et aussi des images et d'autres types de fichiers).

Hyperliens

- Dans un navigateur Web, un document peut pointer vers un autre document à l'aide de liens.
- Un lien est composé de 3 parties
  - le protocole
  - l'adresse du serveur, soit via un nom de domaine, soit via une adresse IP
  - le document demandé avec son nom et son adresse dans le domaine.
- Par exemple http://bliaudet.free.fr/article.php3?id_article=216
  - http est le protocole.
  - bliaudet.free.fr est le nom de domaine qui pointe vers mon serveur

  ⇒ article.php3?id_article=216 est l'URL du document dans son domaine : ici le fichier article.php3 auquel est ajouté un couple clé-valeur.

Requête HTTP

- Quand son saisis une adresse et qu'on appuie sur Entrée dans le navigateur, ce qui est envoyé sur le réseau, c'est la requête

```
GET /a-page HTTP/1.1
```

  - GET est la méthode HTTP utilisée (on dit aussi verbe)
  - /une-page est l'URL demandée.
  - HTTP/1.1 est le protocole utilisé

- Au début, HTTP définissait 3 verbes :
  - GET
  - POST
  - HEAD
- HTTP-1.1 a introduit :
  - PUT
  - DELETE
  - OPTIONS
  - TRACE
- Il existe une troisième partie dans la requête :

```
GET /a-page HTTP/1.1 Host: bliaudet.free.fr
```

  - Host est ce qu'on appelle une en-tête HTTP. C'est la seul qui est obligatoire
- Liste des en-têtes HTTP, toutes facultatives sauf Host :

Les méthodes HTTP

GET

- GET est la méthode la plus utilisée.
- C'est celle qui est utilisée à quand on tape une URL dans la barre d'adresse du navigateur ou lorsque vous cliquez sur un lien.
- GET demande au serveur d'envoyer la ressource demandée en tant que réponse.

HEAD

- HEAD est comme GET, mais dit au serveur de ne renvoyer que les en-têtes et pas le corps de la réponse.

POST

- POST est utilisé pour envoyer des données au serveur. Il est généralement utilisé dans les formulaires, mais également lors de l'interaction avec une API REST.

PUT

- PUT est utilisé pour créer une ressource (autrement dit pour modifier le serveur, d'une façon ou d'une autre) avec les paramètres passés dans le corps de la requête. Principalement utilisé dans les API REST

DELETE

- DELETE est utilisé pour supprimer une ressource. Principalement utilisé dans les API REST

OPTIONS

- Quand OPTIONS est utilisé, le serveur doit renvoyer la liste des méthodes HTTP autorisées pour l'URL demandée.

TRACE

- Quand TRACE est utilisé, le serveur doit renvoyer la demande qui a été reçue à des fins de débogage ou de diagnostic.

Communication client / serveur HTTP

- HTTP, comme la plupart des protocoles appartenant à la suite TCP / IP, est un protocole sans état, c'est-à-dire que le serveur n'a aucune idée de l'état actuel du client. Le serveur reçoit une demande, il doit y répondre : point final !
- Toute demande préalable n'a pas de sens dans ce contexte, ce qui permet et cela permt à un serveur Web d'être très rapide.
- D'autres protocoles, comme impliquent beaucoup de handshaking et de confirmations aux extrémités de réception, ce qui les ralentit.
- Les navigateurs graphiques utilisent le protocole HTTP.
- Un message est composé d'une première ligne, qui commence par la méthode HTTP, puis contient le chemin relatif de la ressource et la version du protocole, puis les lignes des en-têtes HTTP. La seule obligatoire est Host:

```
GET /exo1.json HTTP/1.1
Host: bliaudet.free.fr
```

Tester avec telnet

Installation sur mac

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install
          )"

brew install telnet
```

Tester avec telnet

On démarre telnet en lui passant le nom de domaine
Ensuite, on passe la requête GET, sur 2 lignes, et on tape entréee.
Ici on demande des fichiers json et txt : ils s'affichent.

```
telnet bliaudet.free.fr 80

GET /exo1.json HTTP/1.1
Host: bliaudet.free.fr

telnet bliaudet.free.fr 80

GET /anthropocene/latour/gaia-en-une-heure.txt HTTP/1.1
Host: bliaudet.free.fr
```

- Le web fonctionne comme ça : lorsqu'une page HTML est récupérée par le navigateur, elle est interprétée par le navigateur et, en général, il l'affiche. Le navigateur peut récupérer du HTML, des fichiers texte (txt, json, php, etc.), des images, du son, des vidéos.

Alternative

- On peut aussi utiliser curl à la place de telnet.

Le protocole HTTPS

- HTTP n'est pas sécurisé. Quand on demande une page à un serveur Web, des données circulent du navigateur au serveur puis du serveur au navigateur, en passant par différents ordinateurs du réseau, jusqu'à atteindre l'ordinateur hôte du serveur Web.
- En fonction de la demande, il peut y avoir d'autres connexion pour récupérer des fichiers CSS, des fichiers JavaScript, les images, etc.
- Il est très facile pour quelqu'un d'écouter simplement les paquets HTTP transmis sur un réseau Wi-Fi public et non chiffré.
- Au cours d'une de ces connexions, les données qui circulent peuvent être observées et manipulées : des publicités peuvent être injectées, des virus peuvent être injectés, des données sensibles peuvent être récupérées.
- HTTPS résout le problème à la racine : toute la communication entre votre navigateur et le serveur Web est cryptée.
- Aujourd'hui, HTTPS est une exigence sur n'importe quel site. Plus de 50% de l'ensemble du Web l'utilise maintenant. Google Chrome a récemment commencé à marquer les sites HTTP comme non sécurisés, pour inciter à l'utilisation de HTTPS.
- Port du serveur par défaut (pas besoin donc de l'ajouter explicitement) :
  - HTTP : 80
  - HTTPS : 443
- HTTPS est parfois appelé HTTP sur SSL ou HTTP sur TLS (TLS est le successeur de SSL).
- Lors de l'utilisation de HTTPS, la seule chose qui n'est pas chiffrée est le domaine du serveur Web et le port du serveur.
- HTTPS permet l'utilisation du dernier protocole HTTP / 2 qui est beaucoup plus rapide que http /1.1. Ainsi HTTPS, malgré la surcharge de chiffrement, est plus rapide que HTTP, si les choses sont correctement configurées.

Les protocole HTTP / 2 et HTTP / 3

- HTTP / 2, sorti en 2015, est la dernière version normalisée du protocole HTTP.
- HTTP / 2 est bien plus performant que HTTP / 1.1.
- HTTP / 2 est 100% rétro compatible avec HTTP / 1.1 : les sites Web et applications Web fonctionnent maintenant beaucoup plus rapidement automatiquement.
- HTTP / 3 est en cours de développement et sera basé sur le protocole QUIC.
- QUIC est un protocole basé sur UDP (plutôt que TCP) au niveau de la couche de transport, ce qui signifie que HTTP / 3 sera complètement différente de HTTP / 2 et HTTP.

Websockets

- Les WebSockets sont une alternative à la communication HTTP dans les applications Web.
- Ils offrent un canal de communication bidirectionnel de longue durée entre le client et le serveur.
- Une fois établi, le canal reste ouvert, offrant une connexion très rapide.

  ⇒ Les WebSockets sont adaptés pour les communications en temps réel et de longue durée et si le serveur veut communiquer avec le client de sa propre initiative,

  ⇒ Le HTTP est adapté pour les échanges de données occasionnels et les interactions initiées par le client.