

Internet et Web — Protocoles et architecture

1^{re}

Programme de première générale, enseignement de spécialité — BO spécial n°1 du 22 janvier 2019

Un cadenas dans la barre d'adresse ne dit pas que le site est honnête : il dit que la communication est chiffrée. Un site frauduleux peut parfaitement en afficher un.

1 L'architecture en couches

RÈGLE

Le réseau est organisé en **couches**, et chaque couche rend un service **précis et un seul**, en s'appuyant sur celle du dessous. La couche **réseau** — **IP** — assure l'**adressage** et le **routage** de paquets, sans garantie d'arrivée ni d'ordre. La couche **transport** — **TCP** — ajoute la **fiabilité** : numérotation, remise en ordre, retransmission des paquets perdus. La couche **application** — **HTTP** — transporte le **sens** de la requête.

IP : où · TCP : complet et dans l'ordre · HTTP : quoi

PROPRIÉTÉ

Cette séparation a une conséquence pratique décisive : chaque couche peut être **remplacée** sans toucher aux autres. On peut changer le support physique — fibre, radio, cuivre — sans modifier **IP** ; on peut remplacer **TCP** par UDP, qui renonce à la fiabilité pour gagner en latence, sans changer **IP** ni la couche physique. C'est cette indépendance qui a permis au réseau d'évoluer pendant des décennies sans être refondu.

UDP illustre bien que la fiabilité n'est pas toujours souhaitable. Pour un appel vidéo, retransmettre une image perdue n'a aucun intérêt : elle arriverait trop tard. Mieux vaut une image manquante qu'une image en retard. **TCP** convient au transfert de fichiers, UDP au temps réel : le choix dépend de ce que l'application peut **tolérer**.

2 Adresser et router

DÉFINITION

Une adresse **IP** identifie une interface sur le réseau. En IPv4, elle tient sur **4 octets**, notés en décimal séparés par des points. La notation **CIDR** — une barre oblique suivie d'un nombre — indique combien de **bits de poids fort** désignent le **réseau**, le reste identifiant la machine à l'intérieur de ce réseau. C'est cette séparation qui rend le **routage** possible : un routeur n'a pas besoin de connaître chaque machine, seulement chaque réseau.

PROPRIÉTÉ

Le **routage** consiste, pour chaque routeur, à lire l'adresse de destination et à transmettre le paquet au **voisin** qui le rapproche du but. Aucun routeur ne connaît l'**itinéraire complet** : chacun ne décide que de l'**étape suivante**, à partir d'une table. D'où la robustesse du réseau — si un lien disparaît, les tables se mettent à jour et les paquets passent ailleurs.

DÉFINITION

Le **DNS** est l'annuaire qui traduit un **nom de domaine** lisible en adresse **IP**. Il est **hiérarchique** et **distribué** : aucun serveur ne détient la totalité de l'annuaire, chacun sait à qui s'adresser pour la suite. Cette architecture le rend robuste, mais il reste un point de **contrôle** : supprimer une entrée rend un site inaccessible **par son nom** sans que le serveur ait cessé de fonctionner.

3 Le Web

PROPRIÉTÉ

Trois langages aux rôles **distincts** construisent une page. **HTML** décrit la **structure** et le contenu : titres, paragraphes, listes, liens. **CSS** décrit la **présentation** : couleurs, tailles, positions. **JavaScript** décrit le **comportement** : ce qui se passe quand on clique, ce qui se met à jour sans recharger. Mélanger les trois rôles — mettre en forme en HTML, structurer en CSS — produit un code que personne ne peut modifier sans tout casser.

RÈGLE

Une requête **HTTP** emploie une **méthode**. **GET** demande une ressource : ses paramètres figurent dans l'**URL**, elle est destinée à être **sans effet** sur le serveur, et elle peut être mise en cache ou rejouée sans risque. **POST** envoie des données destinées à **modifier** un état : ses données sont dans le **corps** de la requête, et la rejouer peut produire l'action deux fois. Employer **GET** pour une action qui modifie quelque chose est donc une faute de conception, pas un détail.

GET : lire, sans effet. POST : modifier, avec effet

4 HTTPS et la confiance

LE PIÈGE QUI COÛTE LE PLUS DE POINTS

Conclure du cadenas d'une connexion sécurisée que le site consulté est légitime et digne de confiance. — **HTTPS** garantit trois choses et **trois seulement** : que la communication est **chiffrée**, qu'elle n'a pas été **modifiée** en route, et que le serveur possède bien le **certificat** du nom de domaine affiché. Il ne dit **rien** sur les intentions de celui qui tient ce domaine : obtenir un certificat pour un nom qu'on contrôle est gratuit et immédiat. Un site frauduleux affiche donc un cadenas parfaitement valide.

Le réflexe : Le cadenas certifie le **tuyau**, jamais l'**interlocuteur** — lire le **nom de domaine**.

Ce qu'il faut vérifier, c'est le **nom de domaine** lui-même, en le lisant **de droite à gauche** à partir de la dernière partie : c'est là que se trouve le domaine réellement contrôlé. Un nom construit pour ressembler à celui d'une banque connue peut contenir ce nom au milieu sans lui appartenir — et le cadenas sera vert, puisqu'il certifie ce nom-là.

SUIVRE LE TRAJET D'UNE REQUÊTE WEB

- 1 Partir du **nom de domaine** saisi et le faire traduire en adresse **IP** par le **DNS**.
- 2 Établir la connexion **TCP** avec le serveur, puis la sécuriser si l'on est en **HTTPS**.
- 3 Envoyer la requête **HTTP** avec sa **méthode** — **GET** pour lire, **POST** pour modifier.
- 4 Suivre les paquets de routeur en routeur, chacun ne décidant que de l'**étape suivante**.
Aucun routeur ne connaît l'itinéraire complet.
- 5 Décirer la réponse : le navigateur reçoit du **HTML**, puis charge les **CSS** et les scripts **JavaScript** qu'il référence.

À VÉRIFIER

- Ai-je attribué l'**adressage** à **IP** et la **fiabilité** à **TCP** ?
- Ai-je dit qu'un routeur ne connaît que l'**étape suivante** ?
- Ai-je expliqué le rôle du **DNS** — et sa fonction de point de contrôle ?
- Ai-je distingué les rôles de **HTML**, **CSS** et **JavaScript** ?
- Ai-je réservé **GET** à la lecture et **POST** à la modification ?
- Ai-je dit ce que **HTTPS** garantit — et ce qu'il ne garantit pas ?

À RETENIR

- Chaque **couche** rend un service **précis et un seul**.
- **IP** : adressage et **routage**. **TCP** : fiabilité. **HTTP** : le sens de la requête.
- UDP renonce à la fiabilité pour gagner en **latence** : utile en temps réel.
- La notation **CIDR** sépare la part **réseau** de la part machine.
- Un routeur ne connaît que l'**étape suivante**, jamais l'itinéraire complet.
- Le **DNS** traduit un nom en **IP** ; il est distribué et point de contrôle.
- **HTML** structure, **CSS** présente, **JavaScript** anime.
- **GET** lit **sans effet** ; **POST** modifie. D'autres logiciels s'y fient.
- **HTTPS** garantit chiffrement, intégrité, et **certificat du nom affiché**.
- Il ne garantit **rien** sur les intentions de celui qui tient le domaine.
- Vérifier le **nom de domaine**, lu **de droite à gauche**.