

Architecture matérielle et réseau

Le réseau

Jean-Christophe FORTON

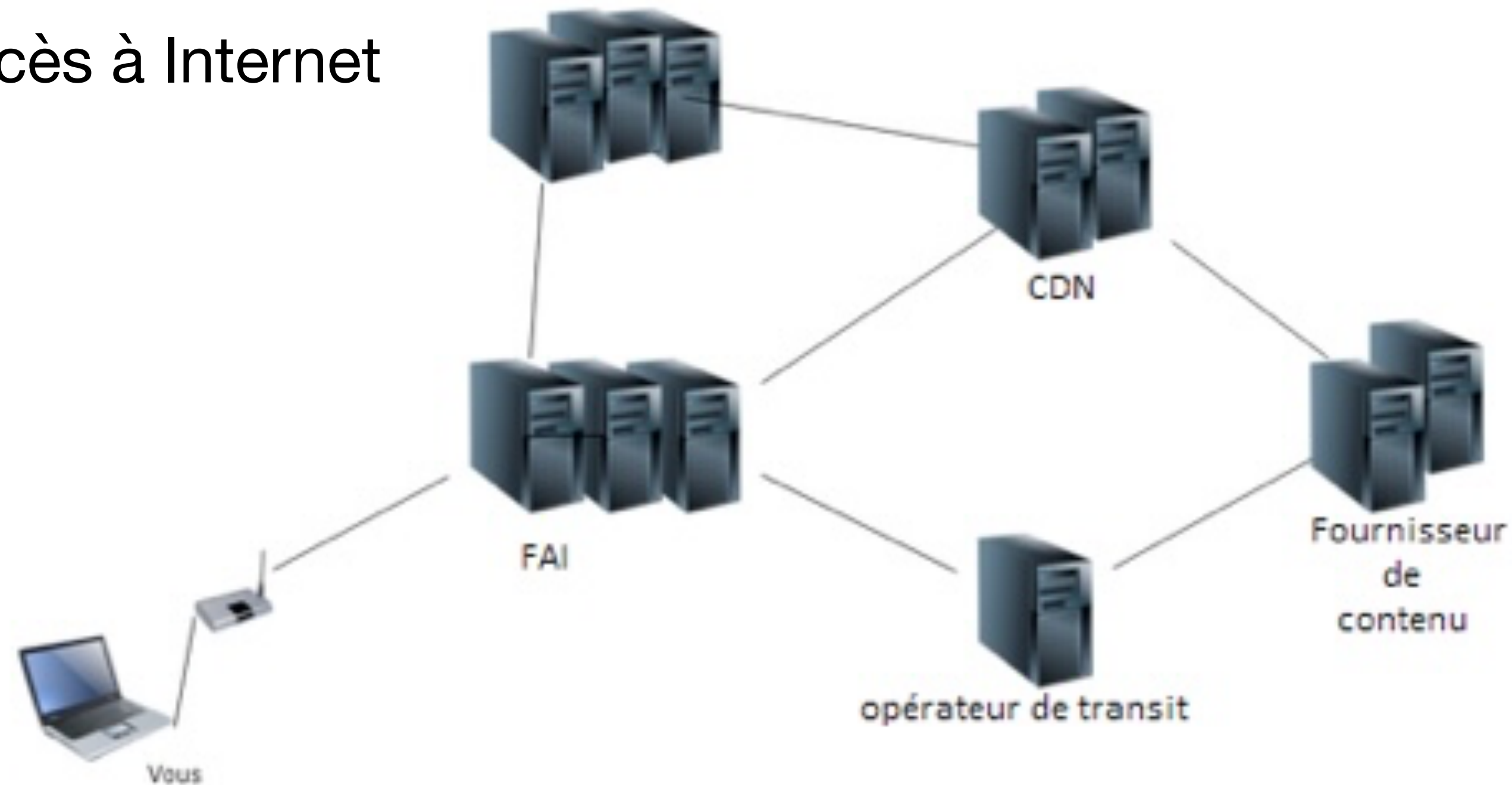
1. Fonctionnement global d'Internet
2. Principe d'une communication
3. Les adresses IP
4. Aperçu du protocole HTTP
5. Le serveur Apache

Fonctionnement global d'Internet

Architecture réseau

Fonctionnement global d'Internet

- C'est un réseau mondial
- FAI : Fournisseur d'Accès à Internet



- CDN (Content Delivery Network) : copie et mise à disposition du contenu, permet d'accélérer les temps de chargement
- Les opérateurs de transit font le lien entre les différents opérateurs
- Les fournisseurs de contenu : Google, Youtube, Twitter, etc.

Architecture réseau

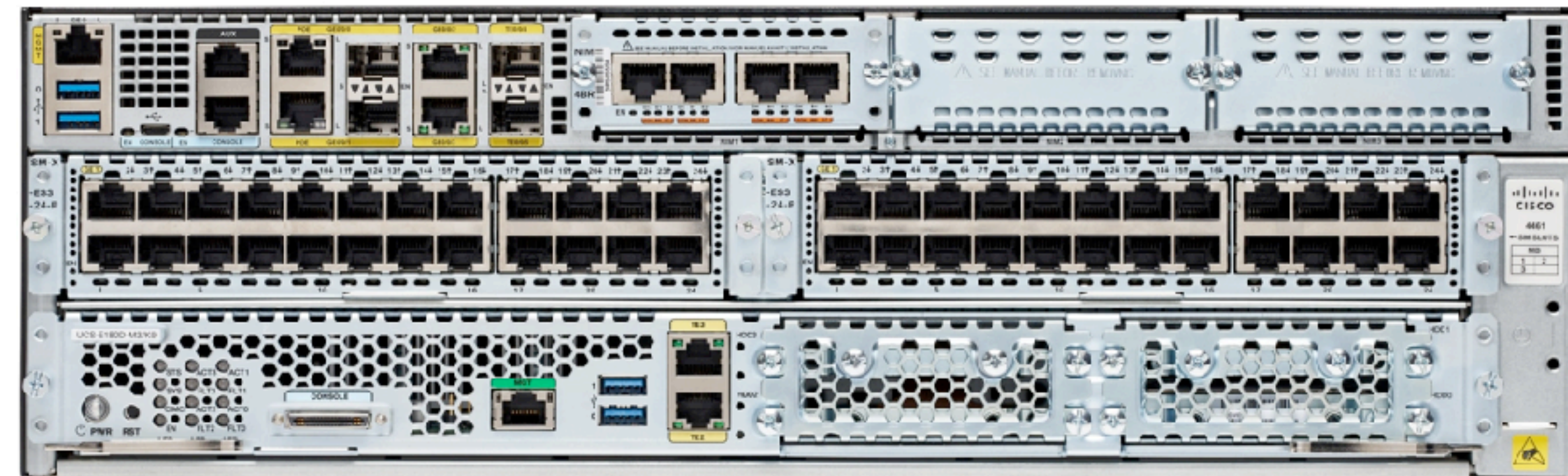
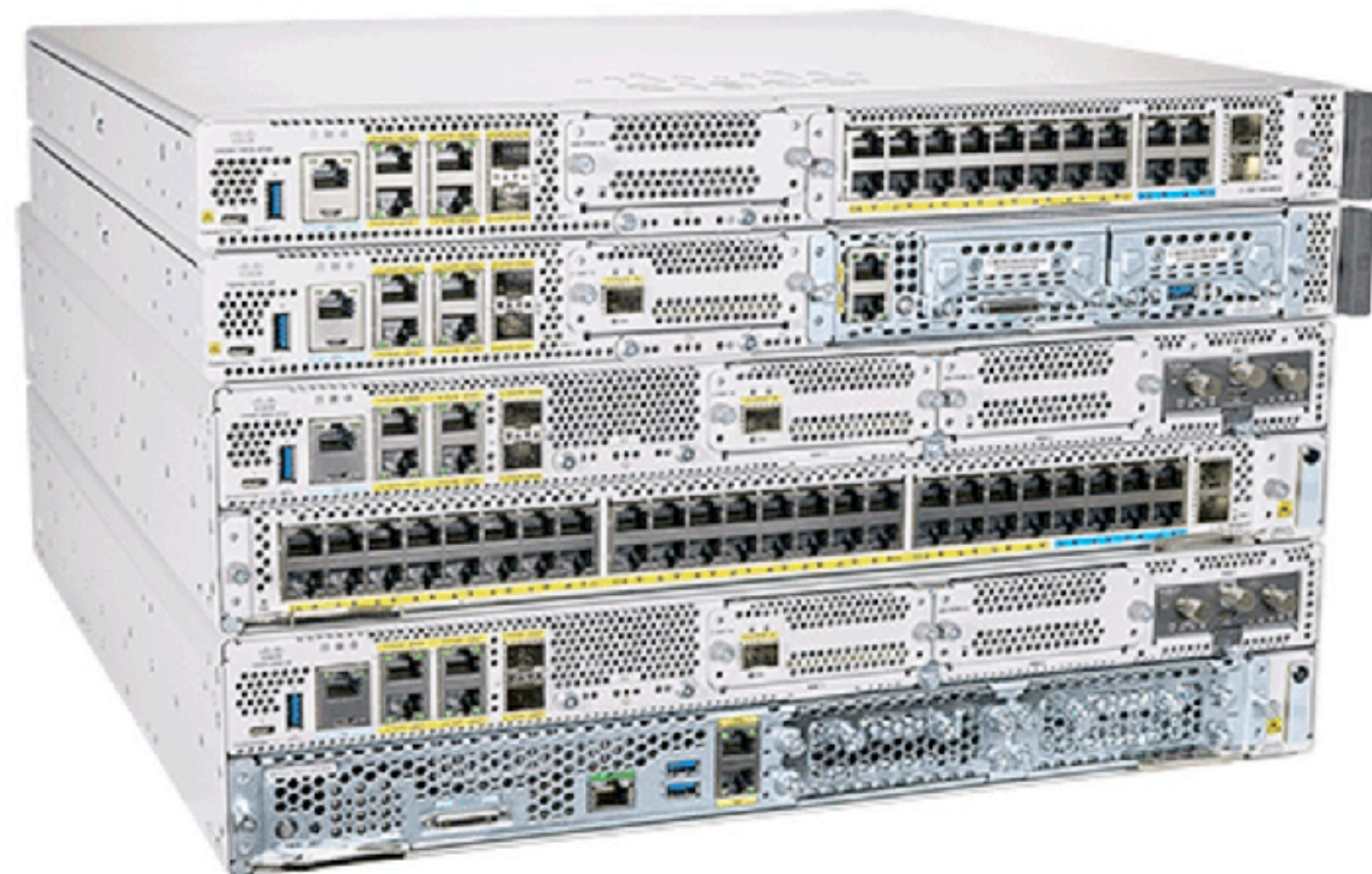
Fonctionnement global d'Internet

Il existe plusieurs chemins pour joindre une destination, cela permet à l'Internet d'être plus robuste.

Les machines qui calculent les itinéraires s'appellent des routeurs.

Les routeurs sont focalisés sur l'opération de routage et ne filtrent pas le contenu.

Ils sont en capacité d'adapter le chemin emprunté par les données en fonction de la congestion du réseau.



Architecture réseau

Fonctionnement global d'Internet

Certains itinéraires possèdent des machines qui bloquent une partie du contenu.

Ces machines, les proxys, sont des équipements de sécurité qui permettent de bloquer du contenu indésirable (filtrage d'URL).

Il peuvent également stocker temporairement une partie du contenu pour accélérer les échanges.

Souvent, le contenu statique est stocké au plus proche de la demande pour être resservi instantanément.



Architecture réseau

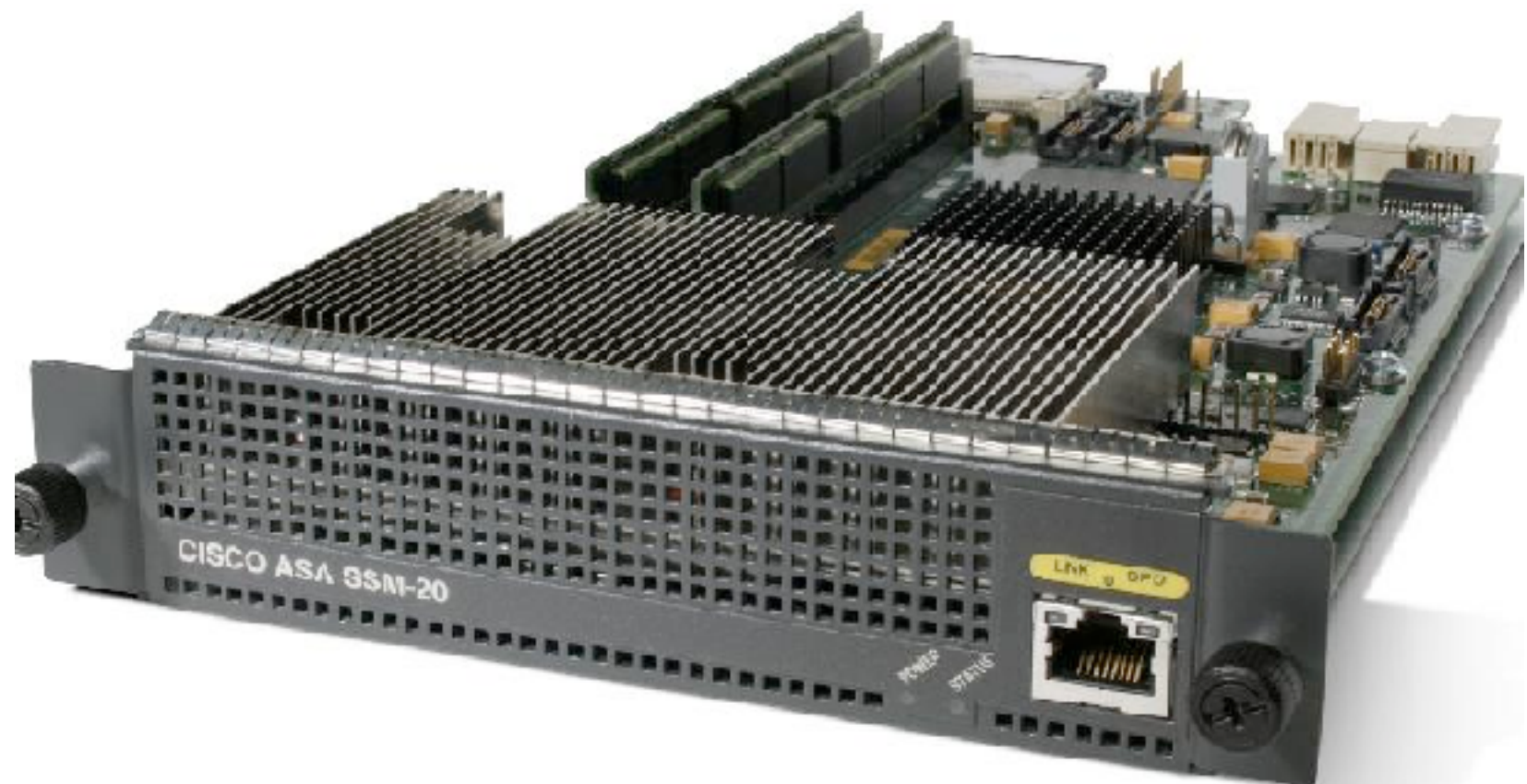
Fonctionnement global d'Internet

Il existe des machines qui sont capables de bloquer des échanges en fonction de la nature ou de la fréquence.

Ces équipements de sécurité, les pare-feux, permettent aux serveurs fournissant le contenu de rester joignables.

Ils sont responsables de la sécurité des échanges sur Internet grâce à l'application d'une politique de filtrage.

Ils existent sous deux formes : matérielle et logicielle.



Architecture réseau

Fonctionnement global d'Internet

Les machines fournissant le contenu s'appellent des serveurs.

Ces machines sont généralement très puissantes et sont raccordées à des réseaux qui permettent d'exploiter pleinement cette puissance de calcul (data center).

Il existe deux types de serveurs : physique et virtuel.

Les serveurs physiques sont devenus tellement puissants que pour exploiter cette puissance, on est obligé de les « découper » en plusieurs serveurs virtuels



Principe d'une communication

Architecture réseau

Principe d'une communication

Pour établir une communication avec un destinataire, il faut connaître son adresse.

Cette adresse doit être unique pour éviter les confusions au niveau des routeurs.

Il existe deux types d'adresse IP : publique et privée.

Les adresses IP privées servent à la communication sur un réseau local (LAN) —> eg. 192.168.0.1

Les adresses IP publiques servent à la communication des machines sur Internet —> eg. 1.2.3.4

Pour assurer l'unicité, plusieurs organismes s'occupent de la distribution des adresses (IANA).

Architecture réseau

Principe d'une communication

Les adresses IP ne sont pas simples à retenir, c'est pourquoi on peut également utiliser des noms pour joindre une machine.

Ces noms permettent de désigner des domaines et se louent pour une certaine durée.

La translation entre nom de domaine et adresse IP est assurée par un certain type de serveur appelé DNS (Domain Name Server).

Les machines et programmes n'utilisent que les adresses IP pour communiquer, ce qui montre l'importance des DNS dans le fonctionnement de l'Internet.

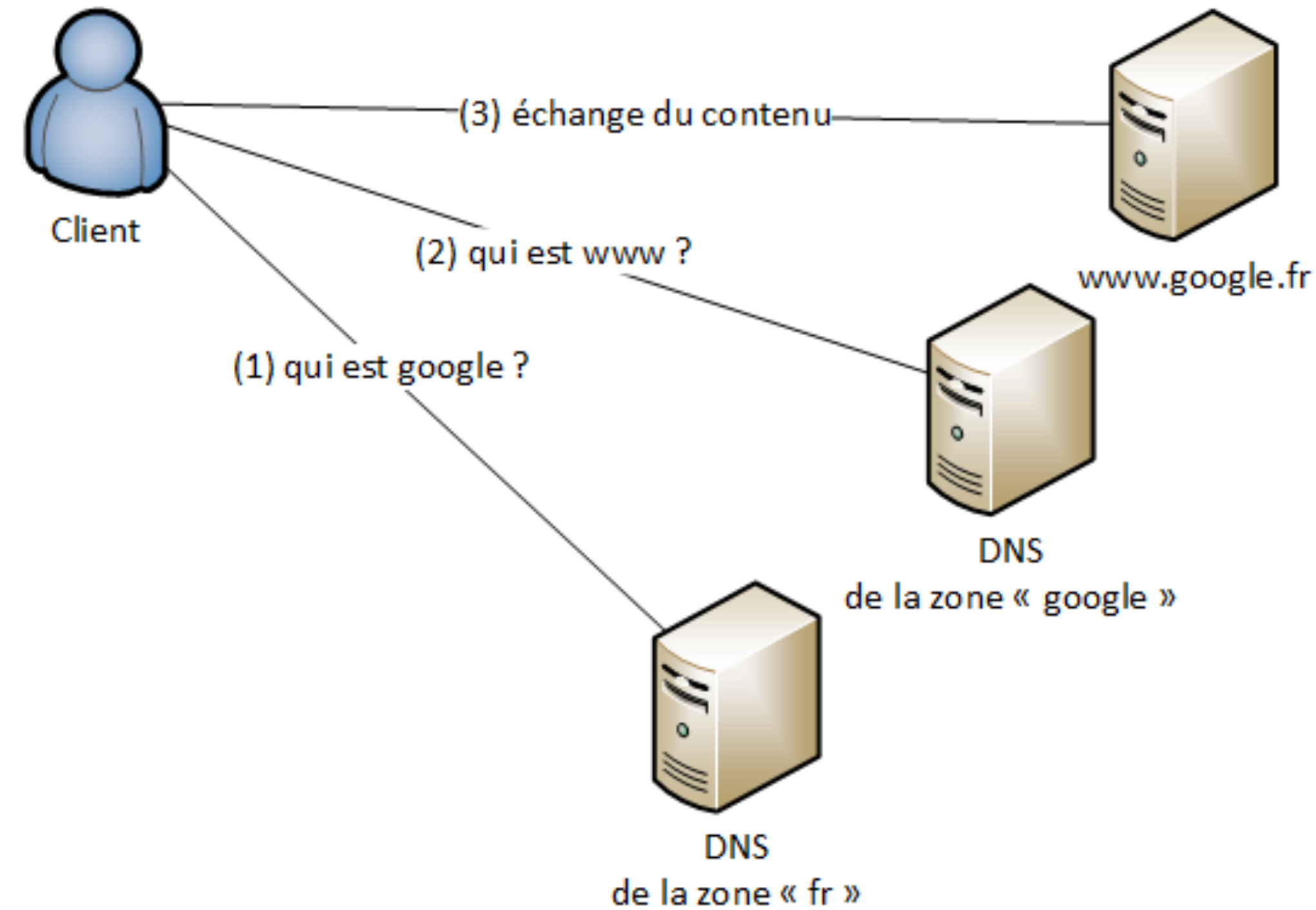
Pour assurer l'unicité des noms de domaines, plusieurs organismes s'occupent de la location des noms de domaine.

Architecture réseau

Principe d'une communication

Comme pour le contenu, l'association entre IP et nom de domaine (résolution de nom) peut être mise dans un cache pour accélérer les échanges.

En fonction de l'organisme qui gère le nom de domaine, ce cache peut mettre jusqu'à 24h pour se rafraîchir.

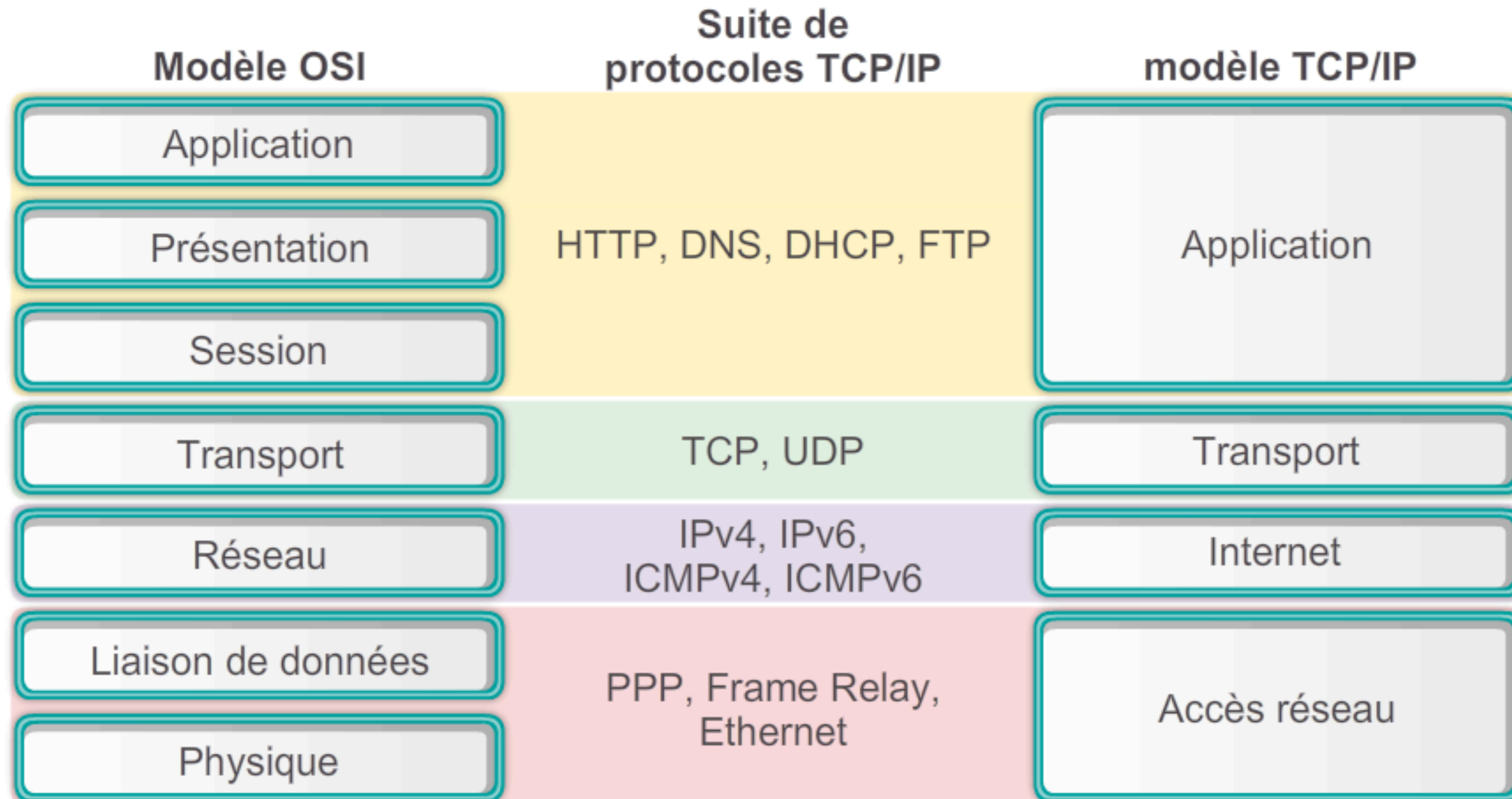


En cas de lenteur, souvent les DNS sont responsables !

Architecture réseau

Principe d'une communication

Pour que la communication fonctionne, les différentes machines d'Internet utilisent toutes un modèle : le modèle OSI.



Architecture réseau

Principe d'une communication

Il existe des équipements pour implémenter le bas niveau du modèle OSI :

- Le switch ou commutateur permet la gestion du niveau 2 OSI en utilisant les adresses Ethernet pour envoyer les messages aux machines d'un **même réseau**.



- Le routeur permet la gestion du niveau 3 OSI en utilisant les adresses IP pour envoyer des messages à travers **des réseaux différents**.

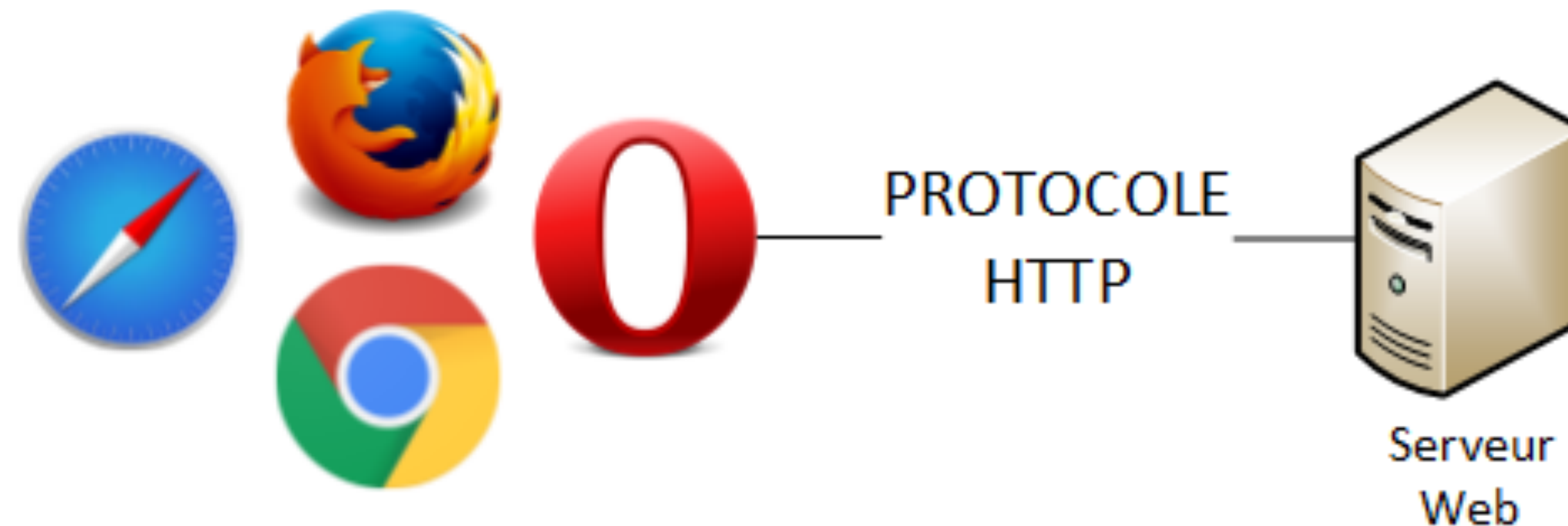
Architecture réseau

Principe d'une communication

Lorsque l'on surfe sur le web, la communication entre les programmes est régie grâce à un protocole : HTTP(S).

En informatique, un protocole est la mise en application d'une norme ou RFC (Request For Comment).

La communication se fait entre un programme client (le navigateur ou butineur) et un programme serveur (Apache, Nginx, Tomcat, ...).



Les adresses IP

Architecture réseau

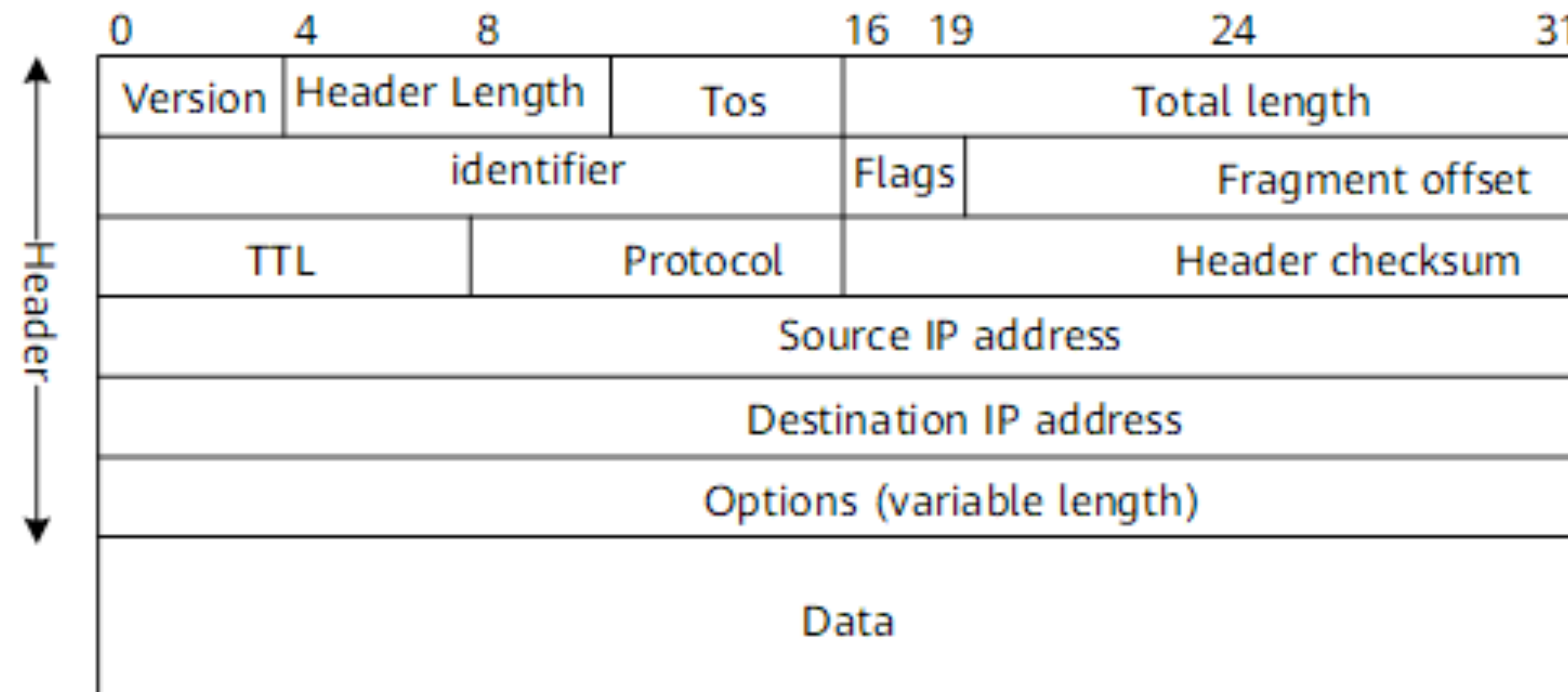
Les adresses IP

Chaque interface avec le réseau peut posséder zéro, une ou plusieurs adresses IP.

Elles sont attribuées soit :

- manuellement par un administrateur réseau ;
- automatiquement par DHCP ;

Les adresses IP sont utilisées dans les entêtes des paquets transmis par le protocole du même nom.



Architecture réseau

Les adresses IP

Au début d'Internet, l'assignation des adresses aux réseaux se faisait grâce au premier octet de l'adresse. Problèmes :

- uniquement 256 réseaux
- 16 millions d'adresses par réseau

La RFC 790 (1981) prévoit le découpage d'une adresse en deux parties:

- *netid* → qui permet d'identifier le réseau ;
- *hostid* → qui permet d'identifier un poste sur le réseau.

Chaque classe possède un *netid* et *hostid* différents:

- Les adresses de classe A ont un net id d'un seul octet (0.0.0.0).
- Les adresses de classe B ont un net id de deux octets (128.0.0.0).
- Les adresses de classe C ont un net id de trois octets (192.0.0.0).
- Les adresses de classe D et E sont réservées (240.0.0.0).

Il est possible de faire une subdivision logique d'un réseau en sous-réseaux.

Le masque de sous-réseau permet de distinguer la partie de l'adresse utilisée pour le routage et celle utilisable pour numéroté des machines.

Avantages:

- Limite la propagation des messages (coûteux en bande passante) ;
- Utilisation de routeurs pour la communication inter-réseaux.

Architecture réseau

Les adresses IP

L'adresse du sous-réseau est obtenue en faisant un ET logique entre l'adresse IP et le masque de sous-réseau.

L'adresse de l'hôte à l'intérieur du sous-réseau est obtenue en faisant un ET logique entre l'adresse IP et le complément à un du masque.

Un masque de sous-réseau possède 4 octets.

Il est constitué (sous sa forme binaire) d'une suite de 1 suivie d'une suite de 0.

Il y a donc 32 masques réseau possibles.

Architecture réseau

Les adresses IP

Un octet permet de représenter un nombre décimal (base 10) sous forme binaire (base 2).

Un octet est constitué de 8 bits.

Le bit de poids fort est situé à gauche et celui de poids faible à droite.

A chaque bit est attribuée une valeur.

Bit	7	6	5	4	3	2	1	0
Valeur décimale	128	64	32	16	8	4	2	1

Architecture réseau

Les adresses IP

Modifier la valeur de ces bits permet de faire varier le résultat de l'addition qui permet d'obtenir la valeur décimale d'un octet.

Par exemple, pour faire 30 en binaire on fait :

Bit	7	6	5	4	3	2	1	0
Valeur décimale	128	64	32	16	8	4	2	1
Valeur binaire	0	0	0	1	1	1	1	0

Le complément à 1 est obtenu en inversant toutes les valeurs d'un octet.

Valeur	1	1	0	0	0	0	1	0
Complément à 1	0	0	1	1	1	1	0	1

L'opération ET logique possède la table de vérité suivante :

a	b	a ET b
0	0	0
0	1	0
1	0	0
1	1	1

Architecture réseau

Les adresses IP

Exemple:

Calculons le net id et host id pour l'adresse 192.168.0.30 et le masque 255.255.255.0

192	168	0	30
11000000	10101000	00000000	00011110

255	255	255	0
11111111	11111111	11111111	00000000

Architecture réseau

Les adresses IP

Exemple:

Calculons le net id et host id pour l'adresse 192.168.0.30 et le masque 255.255.255.0:

- net id

	11000000	10101000	00000000	00011110
&	11111111	11111111	11111111	00000000
=	11000000	10101000	00000000	00000000

- host id

	11000000	10101000	00000000	00011110
&	00000000	00000000	00000000	11111111
=	00000000	00000000	00000000	00011110

La notation CIDR a été introduite en 1993 par l'IETF (RFC 1338).

Elle correspond au nombre de bits à 1 du masque de sous-réseau.

Elle permet l'agrégation d'adresses et ainsi l'allégement de la charge de travail des routeurs d'Internet.

Elle rend obsolète la notion de classe → un masque ne peut plus être déduit d'une adresse IP;

Une adresse IP doit obligatoirement être accompagnée d'un masque de sous-réseau !

Exemple:

Pour l'adresse 192.168.3.10 et le masque 255.255.255.248

Le masque en binaire est :

11111111.11111111.11111111.11111000

Si on compte le nombre de bits à "1" on obtient un CIDR de 29.

Donc, pour l'hôte 192.168.3.10/29 on est maintenant capable de dire que son masque est 255.255.255.248

Aperçu du protocole HTTP

Architecture réseau

Aperçu du protocole HTTP

Un message HTTP (HyperText Transfert Protocol) se décompose en deux parties : en-tête et corps.

Les en-têtes servent à décrire le contenu et permettent de spécifier la langue, l'encodage des caractères, la longueur du message, ...

Le corps permet de véhiculer le contenu et peut être au format texte ou binaire (image, musique, ...)

HTTP se place au dessus de TCP et fonctionne selon un principe de requêtes/réponses :

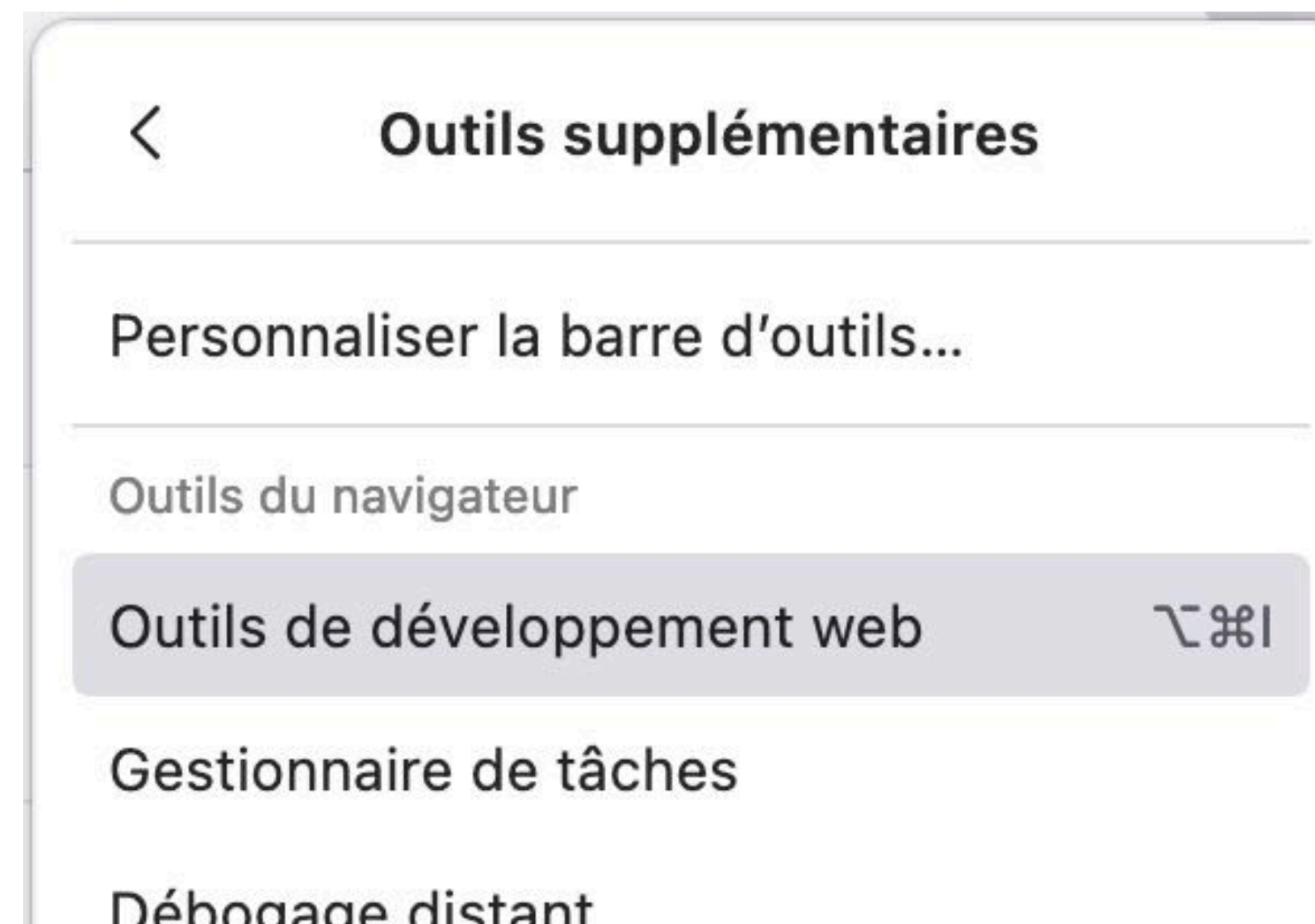
- le client transmet une requête comportant des informations sur le document demandé
- le serveur renvoie le document si disponible ou, le cas échéant, un message d'erreur

Architecture réseau

Aperçu du protocole HTTP

Vous pouvez observer les messages HTTP grâce aux outils de développement web de votre navigateur.

Exemple sur Firefox :



Architecture réseau

Aperçu du protocole HTTP

L'onglet réseau permet de visualiser tous les messages HTTP échangés ainsi que leurs contenus :

The screenshot displays the Chrome DevTools Network tab. The top toolbar includes icons for Inspecteur, Console, Débugueur, Réseau (active), Éditeur de style, Performances, Mémoire, Stockage, Accessibilité, and Applications. Below the toolbar, there's a filter bar with 'Filtrer les URL' and tabs for 'Tout', 'HTML', 'CSS', 'JS', 'XHR', 'Polices', 'Images', 'Médias', 'WS', and 'Autre'. A checkbox for 'Désactiver le cache' and a dropdown for 'Aucune limitation de la bande passante' are also present.

The main table lists network requests with columns: État, Méthode, Domaine, Fichier, Initiateur, Type, Transfert, Taille, and Du. The first request is a GET to reseau-cd.360learning.com for allGroups, which is selected. Other requests include POST to data.eu.pendo.io (blocked by uBlock Origin), GET for various bundle.min.js files, and GET for fonts.gstatic.com.

The right sidebar shows the details for the selected GET request. It includes the URL, status (200 OK), version (HTTP/2), transfer size (1,30 Ko), and request priority (Highest). Below this, the 'En-têtes de la réponse' (783 o) are listed, including access-control-request-method, access-control-request-private-network, cache-control, content-encoding, content-security-policy, content-type, date, set-cookie, strict-transport-security, and X-Firefox-Sndv.

État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	Du
200	GET	reseau-cd.360learning.com	allGroups	document	html	1,30 Ko	1,44 Ko	90
Blocked	POST	data.eu.pendo.io	d5074087-b626-46cf-73b1-75c9f895c...	pendo.js:7 (beac...		Bloquée par uBlock Origin		0 m
200	GET	reseau-cd.360learning.com	initialLoading.gif	img	gif	63,30 Ko	62,88 Ko	57
200	GET	reseau-cd.360learning.com	commons.638bd64f.bundle.min.js	script	js	959,75 Ko	3,29 Mo	38
200	GET	reseau-cd.360learning.com	iterator.dd8a2002.bundle.min.js	script	js	1,45 Ko	2,18 Ko	22
200	GET	reseau-cd.360learning.com	index.886063ad.bundle.min.js	script	js	504,90 Ko	2,98 Mo	38
200	GET	reseau-cd.360learning.com	init	commons.638bd...	json	2,50 Ko	4,42 Ko	2,8
200	GET	reseau-cd.360learning.com	favicon.png	FaviconLoader.js...	png	mis en cache	2,76 Ko	0 m
200	GET	reseau-cd.360learning.com	9556.b431406e.bundle.min.js	index.886063ad....	js	105,25 Ko	356,27 Ko	50
Blocked	POST	api.amplitude.com	/	commons.638bd...		Bloquée par uBlock Origin		0 m
200	GET	reseau-cd.360learning.com	sequences	commons.638bd...	json	764 o	172 o	66
200	GET	reseau-cd.360learning.com	onboardingProgress	commons.638bd...	json	591 o	2 o	39
200	GET	reseau-cd.360learning.com	mine	commons.638bd...	json	826 o	1,04 Ko	102
200	GET	reseau-cd.360learning.com	platform	commons.638bd...	json	1,41 Ko	2,35 Ko	53
200	GET	fonts.gstatic.com	memtYaGs126MiZpBA-UFUIcVXSCEkx2c	commons.638bd...	woff2	mis en cache	0 o	0 m
200	GET	fonts.gstatic.com	memtYaGs126MiZpBA-UFUIcVXSCEkx2c	commons.638bd...	woff2	mis en cache	0 o	0 m

91 requêtes | 9,52 Mo / 1,67 Mo transférés | Terminé en : 14,71 s | DOMContentLoaded: 706 ms | load: 710 ms

Architecture réseau

Aperçu du protocole HTTP

Le premier volet permet de visualiser l'ensemble des requêtes :

État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	Du
200	GET	reseau-cd.360learning.com	allGroups	document	html	1,30 Ko	1,44 Ko	90
	POST	data.eu.pendo.io	d5074087-b626-46cf-73b1-75c9f895c0	pendo.js:7 (beac...		Bloquée par uBlock Origin		0 m
200	GET	reseau-cd.360learning.com	initialLoading.gif	img	gif	63,30 Ko	62,88 Ko	57
200	GET	reseau-cd.360learning.com	commons.638bd64f.bundle.min.js	script	js	959,75 Ko	3,29 Mo	38
200	GET	reseau-cd.360learning.com	iterator.dd8a2002.bundle.min.js	script	js	1,45 Ko	2,18 Ko	22
200	GET	reseau-cd.360learning.com	index.886063ad.bundle.min.js	script	js	504,90 Ko	2,98 Mo	38
200	GET	reseau-cd.360learning.com	init	commons.638bd...	json	2,50 Ko	4,42 Ko	2,8
200	GET	reseau-cd.360learning.com	favicon.png	FaviconLoader.js...	png	mis en cache	2,76 Ko	0 m
200	GET	reseau-cd.360learning.com	9556.b431406e.bundle.min.js	index.886063ad....	js	105,25 Ko	356,27 Ko	50

Architecture réseau

Le deuxième volet permet de visualiser le contenu des messages :

N'hésitez pas à faire un tour dans ces onglets pour comprendre ce qui ne fonctionne pas dans votre code !

Aperçu du protocole HTTP

The screenshot shows the 'En-têtes' (Headers) tab of a web browser's developer tools. The request is a GET to `https://reseau-cd.360learning.com/dashboard/group/allGroups`. The status is **200 OK**. The response headers are as follows:

- `access-control-request-method`: GET,POST,PUT,OPTIONS
- `access-control-request-private-network`: true
- `cache-control`: no-cache, no-store, private
- `content-encoding`: gzip
- `content-security-policy`: frame-ancestors 'self' `https://teams.microsoft.com`
- `content-type`: text/html; charset=utf-8
- `date`: Mon, 26 Sep 2022 09:59:43 GMT
- `set-cookie`: fromSSO=; Path=/; Expires=Thu, 01 Jan 1970 00:00:00 GMT
- `set-cookie`: target=; Path=/; Expires=Thu, 01 Jan 1970 00:00:00 GMT
- `set-cookie`: fromSSO=SAML; Path=/; Expires=Mon, 26 Sep 2022 10:04:43 GMT
- `set-cookie`: externalPlatformInformation=; Path=/; Expires=Thu, 01 Jan 1970 00:00:00 GMT
- `strict-transport-security`: max-age=31536000; includeSubDomains; preload; always;

Le format général d'une URL HTTP est le suivant :

`http://<hôte>:<port>/<chemin>?<requête>#<fragment>`

- `<hôte>` → nom d'hôte ou adresse IP
- `<port>` → généralement 80 ou 443
- `<chemin>` → permet de désigner le fichier désiré
- `<requête>` → suite de tuples (clés/valeurs) séparés par des '&'
- `<fragment>` → permet d'indiquer une position dans la page

Encodage d'URL :

Les caractères ne pouvant être représentés dans une URL comme ';' '/' '?' '&', ...) doivent être échappés.

Afin de rendre "escaped" un caractère (par exemple '&'), il faut remplacer ce caractère par son code ASCII codé en hexadécimal (26 pour '&') et le faire précéder par le caractère '% '.

Le caractère '&' devient donc %26, un espace ' ' devient %20, ...

Exemples :

`http://www.exemple.com`

`http://www.exemple.com:80/news/search?cle=valeur`

`http://www.exemple.com/texte%20avec%20espace/exemple.html`

Voici les méthodes définies dans le protocole et les plus utilisées sont GET, POST, PUT et DELETE :

Méthodes	1.0	1.1	Description
Get			Permet de demander un document
Post			Permet de transmettre des données (d'un formulaire par exemple) à l'URL spécifiée dans la requête. L'URL désigne en général un script Perl, PHP...
Head			Permet de ne recevoir que les lignes d'en-tête de la réponse, sans le corps du document
Options			Permet au client de connaître les options du serveur utilisables pour obtenir une ressource
Put			Permet de transmettre au serveur un document à enregistrer à l'URL spécifiée dans la requête
Delete			Permet d'effacer la ressource spécifiée
Trace			Permet de signaler au serveur qu'il doit renvoyer la requête telle qu'il la reçoive
Connect			Permet de se connecter à un <i>proxy</i> ayant la possibilité d'effectuer du <i>tunneling</i>

Un message HTTP est composé :

- d'en-têtes

```
? Accept: application/json, text/plain, */*
? Accept-Encoding: gzip, deflate, br
? Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3
? Connection: keep-alive
? Content-Type: application/json; charset=utf-8
? Cookie: amp_18810d=PSmfML_6e3BTWEjwn47brn...1gdkh
token=evlhGciOiJLLlZlLXMiLnR5cCI6IkpXVCJ9.eyJ1c2Vz
```

- d'un contenu (optionnel)



The screenshot shows the 'Réponse' tab in a web browser's developer tools. The JSON response is as follows:

```
{
  "name": "homepage",
  "isStepCounterHidden": true,
  "steps": [
    {
      "name": "courseCreation"
    },
    {
      "name": "programCreation"
    }
  ]
},
{
  "name": "newHomepage",
  "isStepCounterHidden": true,
  "steps": []
}
```

Architecture réseau

Aperçu du protocole HTTP

Lorsque le serveur renvoie un document, il lui associe un code de statut renseignant le client sur le résultat de la requête (requête invalide, document non trouvé...).

Les principales valeurs des codes de statut HTTP sont détaillées dans le tableau ci-après.

Code	Nom	Description
Information 1xx		
100	Continue	Utiliser dans le cas où la requête possède un corps.
101	Switching protocol	Réponse à une requête
Succès 2xx		
200	OK	Le document a été trouvé et son contenu suit
201	Created	Le document a été créé en réponse à un PUT
202	Accepted	Requête acceptée, mais traitement non terminé
204	No response	Le serveur n'a aucune information à renvoyer
206	Partial content	Une partie du document suit
Redirection 3xx		
301	Moved	Le document a changé d'adresse de façon permanente
302	Found	Le document a changé d'adresse temporairement
304	Not modified	Le document demandé n'a pas été modifié
Erreurs du client 4xx		
400	Bad request	La syntaxe de la requête est incorrecte
401	Unauthorized	Le client n'a pas les privilèges d'accès au document
403	Forbidden	L'accès au document est interdit
404	Not found	Le document demandé n'a pu être trouvé
405	Method not allowed	La méthode de la requête n'est pas autorisée
Erreurs du serveur 5xx		
500	Internal error	Une erreur inattendue est survenue au niveau du serveur
501	Not implemented	La méthode utilisée n'est pas implémentée
502	Bad gateway	Erreur de serveur distant lors d'une requête <i>proxy</i>

Le serveur Apache

Architecture réseau

Le serveur Apache

Le rôle du serveur web est de servir des ressources au travers du protocole HTTP.

Lorsque l'on parle d'un serveur web, on désigne généralement le processus qui s'exécute sur une machine et qui répond aux requêtes des clients.

Un tel processus s'appelle un démon.

```
root@devmach:~  
[root@devmach ~]# ps -ef | grep httpd  
root      1782      1  0 Oct11 ?           00:00:21 /usr/sbin/httpd  
apache    11694    1782  0 Oct12 ?           00:00:00 /usr/sbin/httpd  
apache    11695    1782  0 Oct12 ?           00:00:00 /usr/sbin/httpd  
apache    11696    1782  0 Oct12 ?           00:00:00 /usr/sbin/httpd  
apache    11697    1782  0 Oct12 ?           00:00:00 /usr/sbin/httpd  
apache    11698    1782  0 Oct12 ?           00:00:00 /usr/sbin/httpd  
apache    11699    1782  0 Oct12 ?           00:00:00 /usr/sbin/httpd  
apache    11700    1782  0 Oct12 ?           00:00:00 /usr/sbin/httpd  
apache    11701    1782  0 Oct12 ?           00:00:00 /usr/sbin/httpd  
root      16916   16899  0 14:17 pts/0       00:00:00 grep httpd  
[root@devmach ~]#
```

Architecture réseau

Le serveur Apache

Comme la plupart des programmes, son comportement peut être modifié grâce à un fichier de configuration.

Dans le cas d'Apache, ce fichier se nomme « httpd.conf » et il est possible de modifier beaucoup de paramètres, notamment :

- Le port d'écoute
- Le répertoire de travail
- Les fichiers de logs
- Les sites hébergés
- Les URLs de redirection
- ...

```
root@devmach:~  
# NOTE!  If you intend to place this on an NFS (or otherwise network)  
# mounted filesystem then please read the LockFile documentation  
# (available at <URL:http://httpd.apache.org/docs/2.2/mod/mpm_common.html#lockfile>);  
# you will save yourself a lot of trouble.  
#  
# Do NOT add a slash at the end of the directory path.  
#  
ServerRoot "/etc/httpd"  
  
#  
# PidFile: The file in which the server should record its process  
# identification number when it starts.  Note the PIDFILE variable in  
# /etc/sysconfig/httpd must be set appropriately if this location is  
# changed.  
#  
PidFile run/httpd.pid  
  
#  
# Timeout: The number of seconds before receives and sends time out.  
#  
Timeout 60  
  
#  
# KeepAlive: Whether or not to allow persistent connections (more than  
# one request per connection).  Set to "Off" to deactivate.  
#  
KeepAlive Off
```


Architecture réseau

Le serveur Apache

Le processus répond aux requêtes en servant des fichiers disponibles dans un répertoire particulier appelé répertoire de travail.

Ce répertoire est généralement 'www' et il y a une correspondance qui est faite entre les URLs et l'arborescence sous le répertoire de travail.

`http://monserver/index.php` → `c:\wamp\www\index.php`

ou

`http://monserver/index.php` → `/var/www/html/index.php`

Architecture réseau

Le serveur Apache

Le processus serveur note absolument toute son activité dans des fichiers. Cette pratique s'appelle la journalisation ou « logging ».

Il existe, de base, deux fichiers de logs : *access_log* et *error_log*

access_log permet de journaliser toutes les requêtes

```
[root@devmach ~]# cat /var/log/httpd/access_log
192.168.186.1 - - [13/Oct/2018:14:50:18 +0200] "GET / HTTP/1.1" 403 4961 "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0"
192.168.186.1 - - [13/Oct/2018:14:50:19 +0200] "GET /icons/apache_pb.gif HTTP/1.1" 200 2326 "http://192.168.186.201/" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0"
192.168.186.1 - - [13/Oct/2018:14:50:19 +0200] "GET /icons/poweredby.png HTTP/1.1" 200 3956 "http://192.168.186.201/" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0"
192.168.186.1 - - [13/Oct/2018:14:50:19 +0200] "GET /favicon.ico HTTP/1.1" 404 290 "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0"
192.168.186.1 - - [13/Oct/2018:14:50:19 +0200] "GET /favicon.ico HTTP/1.1" 404 290 "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0"
[root@devmach ~]#
```


Architecture réseau

Le serveur Apache

error_log permet de journaliser toutes les erreurs ou événements serveur

```
[root@devmach ~]# cat /var/log/httpd/error_log
[Fri Oct 12 17:20:02 2018] [notice] Digest: generating secret for digest authentication ...
[Fri Oct 12 17:20:02 2018] [notice] Digest: done
[Fri Oct 12 17:20:02 2018] [notice] Apache/2.2.15 (Unix) DAV/2 PHP/7.0.25 configured -- resuming normal operations
[Sat Oct 13 14:50:18 2018] [error] [client 192.168.186.1] Directory index forbidden by Options directive: /var/www/html/
[Sat Oct 13 14:50:19 2018] [error] [client 192.168.186.1] File does not exist: /var/www/html/favicon.ico
[Sat Oct 13 14:50:19 2018] [error] [client 192.168.186.1] File does not exist: /var/www/html/favicon.ico
```

En fonction du langage de programmation utilisé, un autre fichier peut être créé comme *www-error.log* ou *php_error.log*

```
[27-Jun-2018 21:54:01 UTC] PHP Notice: Undefined variable: json in C:\wamp64\www\MusicSender\src\classes\shell\Youtube.class.php on line 162
[27-Jun-2018 22:04:10 UTC] PHP Notice: Undefined offset: 1 in C:\wamp64\www\MusicSender\src\classes\shell\Youtube.class.php on line 179
[27-Jun-2018 22:06:28 UTC] PHP Notice: Undefined offset: 1 in C:\wamp64\www\MusicSender\src\classes\shell\Youtube.class.php on line 175
[20-Jul-2018 11:35:53 Europe/Paris] PHP Fatal error: Uncaught Error: Call to undefined method Logger::info() in C:\wamp64\www\DomoTech\src\index.php:10
Stack trace:
#0 {main}
  thrown in C:\wamp64\www\DomoTech\src\index.php on line 10

[20-Jul-2018 11:49:22 Europe/Paris] PHP Notice: Use of undefined constant MSG_EOF - assumed 'MSG_EOF' in C:\wamp64\www\DomoTech\src\classes\controller\Modul

[20-Jul-2018 11:49:22 Europe/Paris] PHP Stack trace:
```


Architecture réseau

Le serveur Apache

Le serveur peut étendre ses capacités (fonctionnalités) grâce à l'ajout de modules.

Ces modules sont paramétrables grâce à des fichiers de configuration se trouvant dans le répertoire « conf.d ».

```
[root@proxy ~]# ll /etc/httpd/conf.d/
total 48
-rw-r--r-- 1 root root 2926 25 janv. 2022 autoindex.conf
-rw-r--r-- 1 root root 370 7 déc. 2021 ct-proxy.conf
-rw-r--r-- 1 root root 245 4 août 19:54 domotech.conf
-rw-r--r-- 1 root root 196 5 déc. 2021 md_ssl.conf
drwxr-xr-x 2 root root 4096 20 avril 07:47 old
-rw-r--r-- 1 root root 400 25 janv. 2022 README
-rw-r--r-- 1 root root 1076 5 déc. 2021 ssl.conf
-rw-r--r-- 1 root root 1077 7 nov. 2021 ssl.conf.rpmsave
drwxr-xr-x 2 root root 4096 20 avril 07:48 tala
-rw-r--r-- 1 root root 251 15 nov. 2021 torrent-proxy.conf
-rw-r--r-- 1 root root 1252 25 janv. 2022 userdir.conf
-rw-r--r-- 1 root root 574 25 janv. 2022 welcome.conf
```

```
[root@proxy ~]# cat /etc/httpd/conf.d/welcome.conf | grep -v '#'
<LocationMatch "^/+>"
    Options -Indexes
    ErrorDocument 403 /.noindex.html
</LocationMatch>

<Directory /usr/share/httpd/noindex>
    AllowOverride None
    Require all granted
</Directory>

Alias /.noindex.html /usr/share/httpd/noindex/index.html
Alias /poweredby.png /usr/share/httpd/icons/apache_pb3.png
```


Architecture réseau

Le serveur Apache

On peut terminer en regardant les tendances en matière de serveur web.

