Approche technique de l'espace numérique

Pablo Rauzy
pr@up8.edu
pablo.rauzy.name/teaching/aten

UNIVERSITÉ PARIS 8 VINCENNES-SAINT-DENIS
Institut Français de Géopolitique / M2 Cyberstratégie & Datascience

Séance 0

Fonctionnement (et gouvernance) d'Internet
Chargement d'une page web

- Sur Wikipédia, a propos de la géopolitique :

  « *la géopolitique a pour objet l'étude des interactions entre l'espace géographique et les rivalités de pouvoirs qui en découlent. [...] elle est le terrain de manœuvre de la puissance locale, régionale ou mondiale.* », Alexandre Defay.
- Le réseau Internet est appuyé sur une architecture physique au même titre que les réseaux d'eau, d'électricité, de gaz, de transport, etc.
- La gestion du réseau et son fonctionnement sont hautement politiques.
- C'est donc un objet d'étude de géopolitique qui semble complètement naturel.

Fonctionnement (et gouvernance) d'Internet

- Internet est organisé hiérarchiquement :
  - au niveau de son infrastructure matérielle, et
  - au niveau des protocoles qui régissent le trafic sur le réseau.
- Pourtant, son fonctionnement reste très décentralisé.

Internet Protocol

- Chaque machine sur le réseau est identifiée par une *adresse IP*.
- Les données qui transitent sur le réseau sont découpées en *paquets*.
  - Chaque paquet a une IP *source* et une IP *destination*.
- La *route* de chaque paquet de machine en machine est indépendante.
  - Elle est décidée au fur et à mesure de manière complètement décentralisée.

Réseau local

- Sur un réseau local, il y a essentiellement deux types de machines :
  - les machines hôtes, et
  - les routeurs.

- Sur une machine *hôte*, le trafic est dirigé en fonction d'une *table de routage*.
- Cette table définit à qui envoyer un paquet en fonction de sa destination :
  - soit à elle même si elle est la destination,
  - soit l'adresser directement à la machine destination si elle est sur le réseau local.
  - soit l'envoyer à une adresse par défaut (le *routeur*).

- Un *routeur* est simplement une machine connectée à plusieurs réseaux.
  - Elle dispose aussi d'une table de routage qui permet de faire passer les paquets d'un réseau à l'autre quand c'est nécessaire.
- Quand les deux réseaux sont de nature différente, on l'appelle *passerelle*.
  - Votre modem-routeur ("box") est typiquement une passerelle entre votre réseau local et Internet. En pratique, entre votre réseau local et votre fournisseur d'accès.

Systèmes autonomes (AS)

- Un *système autonome* est un ensemble de réseaux dont le routage interne est cohérent.
  - Exemple typique d'AS : un fournisseur d'accès à Internet (FAI).
- Un protocole de routage interne calcule tous les plus courts chemins dans l'AS.
- Les paquets qui ont une destination externe à l'AS sont :
  - soit envoyés à une passerelle en "bordure" de l'AS (*peering*),
  - soit remontés vers une route par défaut à un FAI de niveau supérieur (*transit*).

- Un AS peut avoir des accords d'*appairage* avec d'autres AS :
  - Il peut s'agir d'un lien physique avec un autre AS (peering privé), ou
  - de la participation à un *point d'échange Internet* (peering public).
- Le protocole BGP permet d'échanger des informations de routage entre AS :
  - par exemple "je sais aller très rapidement à toutes mes destinations internes",
  - ou encore "j'ai un accord de peering me permettant d'aller à telle destination à telle vitesse",

- Un AS peut avoir des accords d'*appairage* avec d'autres AS :
  - Il peut s'agir d'un lien physique avec un autre AS (peering privé), ou
  - de la participation à un *point d'échange Internet* (peering public).
- Le protocole BGP permet d'échanger des informations de routage entre AS :
  - par exemple "je sais aller très rapidement à toutes mes destinations internes",
  - ou encore "j'ai un accord de peering me permettant d'aller à telle destination à telle vitesse",
  - ou même "Je suis YouTube promis juré !"...
  - BGP est capable de s'auto-réparer en propageant les informations concernants les routes cassées, mais ça peut prendre du temps.

- Un *point d'échange Internet* relie plusieurs systèmes autonomes.
- Cela permet à ces AS de s'échanger du trafic directement, sans passer par un FAI :
  - optimisation du coût,
  - optimisation de la vitesse du trafic.

Trois niveaux d'opérateurs Internet

- Opérateurs *tier 3* : pas d'accord de peering, dépendent entièrement d'une offre de transit.
- Opérateurs *tier 2* : ont des accords de peering, nécessitent une offre de transit.
- Opérateurs *tier 1* : voient tout le réseau par peering, n'ont pas besoin de transit.
  - La *default-free zone* est l'ensemble des routeurs qui n'utilisent pas de route par défaut.

Gouvernance

- Concernant le routage, la gouvernance d'Internet est assez décentralisée :
  - une table de routage à n'importe quel niveau peut être configurée manuellement ;
  - bien sûr, il reste un aspect hiérarchique fort dans la structure du réseau.
- En revanche, même si "code is law", la gouvernance de certaines ressources est relativement centralisée.

- L'*Internet Corporation for Assigned Names and Numbers* est l'autorité suprême :
  - elle gère les racines du système des noms de domaines,
  - elle attribue les numéros d'AS et les adresses IP (déléguées depuis le début des années 90 aux *registres Internet régionaux*).
- L'ICANN n'est indépendante du gouvernement américain que depuis 2016…
  - Cela a évidemment fait l'objet de nombreuses controverses (privatisation au profit des GAFAM ?).
- La gouvernance de l'ICANN elle même est assez complexe, je vous laisse découvrir ça :
  `https://www.icann.org/resources/pages/newcomers-2015-04-01-fr`

Chargement d'une page web

Taper une adresse dans son navigateur

- Que se passe-t-il quand on demande à son navigateur de charger une page web ?

Taper une adresse dans son navigateur

- Que se passe-t-il quand on demande à son navigateur de charger une page web ?
  - requête DNS (*Domain Name System*),

Taper une adresse dans son navigateur

- Que se passe-t-il quand on demande à son navigateur de charger une page web ?
  - requête DNS (*Domain Name System*),
  - requête HTTP (*HyperText Transfer Protocol*).

Taper une adresse dans son navigateur

- Que se passe-t-il quand on demande à son navigateur de charger une page web ?
  - requête DNS (*Domain Name System*),
  - requête HTTP (*HyperText Transfer Protocol*).
- On va prendre l'exemple de `www.univ-paris8.fr`.

Requête DNS

- Le but de la requête DNS est de trouver l'adresse IP associée au nom de domaine du site web qu'on veut consulter.
- Cette requête va être adressée à un *serveur de nom*, sous forme d'un *paquet*.
- Un paquet est constitué d'entêtes et d'un corps.

- Votre système crée donc un paquet DNS.
- Les entêtes de ce paquet disent (entre autres) :
  - je suis une requête (et pas une réponse),
  - si le message arrive en une seule fois ou si il est tronqué,
  - j'autorise (ou non) les requêtes récursives.
- Le corps de ce paquet dit :
  - pour le nom de domaine `www.univ-paris8.fr`,
  - je veux l'adresse IPv4 (`A`),
  - sur internet (`IN`).

Entêtes :

Field	Description	Length (bits)
QR	Indicates if the message is a query (0) or a reply (1)	1
OPCODE	The type can be QUERY (standard query, 0), IQUERY (inverse query, 1), or STATUS (server status request, 2)	4
AA	Authoritative Answer, in a response, indicates if the DNS server is authoritative for the queried hostname	1
TC	TrunCation, indicates that this message was truncated due to excessive length	1
RD	Recursion Desired, indicates if the client means a recursive query	1
RA	Recursion Available, in a response, indicates if the replying DNS server supports recursion	1
Z	Zero, reserved for future use	3
RCODE	Response code, can be NOERROR (0), FORMERR(1, Format error), SERVFAIL (2), NXDOMAIN (3, Non existent domain), etc.[31]	4

Anatomie d'un paquet DNS

Requête :

Field	Description	Length (octets)
NAME	Name of the requested resource	Variable
TYPE	Type of RR (A, AAAA, MX, TXT, etc.)	2
CLASS	Class code	2

Réponse :

Field	Description	Length (octets)
NAME	Name of the node to which this record pertains	Variable
TYPE	Type of RR in numeric form (e.g., 15 for MX RRs)	2
CLASS	Class code	2
TTL	Count of seconds that the RR stays valid (The maximum is $2^{31}-1$, which is about 68 years)	4
RDLENGTH	Length of RDATA field (specified in octets)	2
RDATA	Additional RR-specific data	Variable, as per RDLENGTH

Encapsulation dans un datagramme UDP

- La requête ne part pas encore de votre système, avant ça elle est *encapsulée* dans un paquet UDP.
- UDP signifie *User Datagram Protocol*.
- Le rôle de UDP est celui d'une enveloppe de *transport* des données (ici notre paquet DNS) d'un programme source à un programme destination.

Encapsulation dans un datagramme UDP

- La requête ne part pas encore de votre système, avant ça elle est *encapsulée* dans un paquet UDP.
- UDP signifie *User Datagram Protocol*.
- Le rôle de UDP est celui d'une enveloppe de *transport* des données (ici notre paquet DNS) d'un programme source à un programme destination.

- Le système crée donc un paquet UDP.
- Les entêtes de ce paquet UDP contiennent (entre autres) :
  - le port source (par exemple `43234`),
  - le port destination (`53`, le port assigné aux serveurs de noms).
- Le corps de ce paquet contient le paquet DNS.

Anatomie d'un datagramme UDP

Port Source (16 bits)	Port Destination (16 bits)
Longueur (16 bits)	Somme de contrôle (16 bits)
Données (longueur variable)	

Encapsulation dans un paquet IP

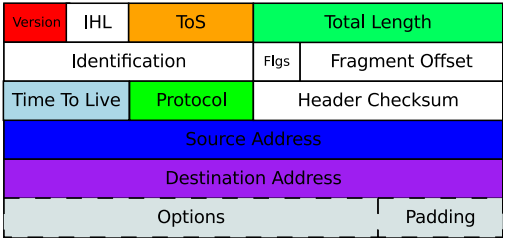
- La requête ne part toujours pas de votre système !
  Avant ça, elle est encapsulée dans un paquet IP.
- IP signifie *Internet Protocol*.
- Le rôle de IP est de servir de transporteur sur le *réseau*.

- La requête ne part toujours pas de votre système !
  Avant ça, elle est encapsulée dans un paquet IP.
- IP signifie *Internet Protocol*.
- Le rôle de IP est de servir de transporteur sur le *réseau*.

- Le système crée donc un paquet IP.
- Les entêtes de ce paquet IP contiennent (entre autres) :
  - la version du protocole utilisé (`4` ou `6`),
  - le protocole utilisé au niveau supérieur (pour nous `17` pour UDP),
  - l'adresse source (par exemple `192.168.1.51`),
  - l'adresse destination (par exemple `192.168.1.254`).

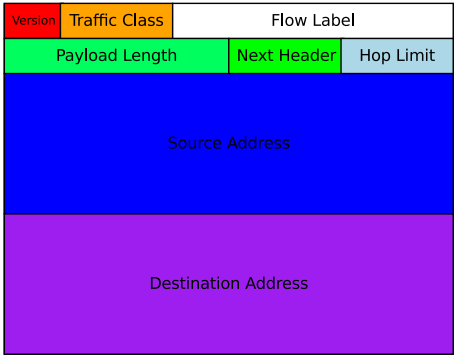
Anatomie d'un paquet IP

IPv4 :



Anatomie d'un paquet IP

IPv6 :

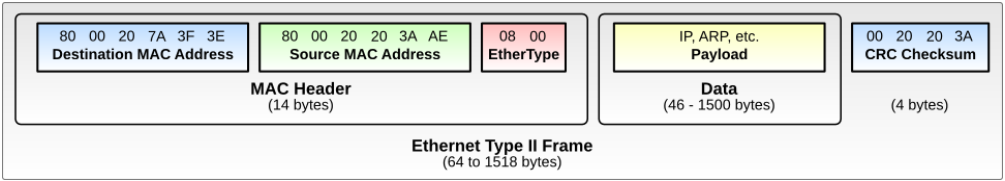


- La requête ne part toujours pas de votre système !
  Avant ça, elle est encapsulée dans un paquet Ethernet.
- Dans notre analogie Ethernet est en quelques sortes la sacoche du postier qui récupère le courrier dans la boîte au lettre et l'amène au centre de tri.

Encapsulation dans une trame Ethernet

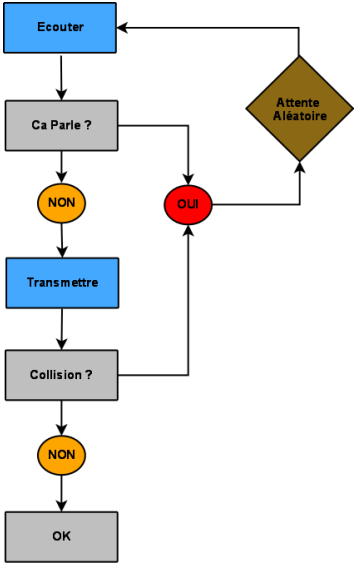
- La requête ne part toujours pas de votre système !
  Avant ça, elle est encapsulée dans un paquet Ethernet.
- Dans notre analogie Ethernet est en quelques sortes la sacoche du postier qui récupère le courrier dans la boîte au lettre et l'amène au centre de tri.
- Le système crée donc un paquet Ethernet.
- Les entêtes de ce paquet Ethernet contiennent (entre autres) :
  - le protocole utilisé au niveau supérieur (pour nous `0×0800` pour IPv4),
  - l'adresse mac source (par exemple `3c:a9:f4:78:78:78`),
  - l'adresse mac destination (par exemple `e8:f1:b0:78:78:78`).

Anatomie d'une trame Ethernet



Ethernet et la politesse

- Le principe d'Ethernet est le même que celui des ondes radio : on communique sur un canal commun et chacun possède une clef unique (adresse MAC).
- L'orchestration se fait avec le protocole CSMA/CD (Carrier Sense Multiple Access with Collision Detection) expliqué ci-contre.



- Arrive enfin le déplacement “physique” des données.
- Dans mon cas, elles sont transmises suivant le protocole 802.11n (WiFi) jusqu'à ma box ADSL.
- Dans notre analogie, il s'agit de la mobylette du postier.

Réponse DNS

- La réponse à ma requête DNS est transmise par ma box ADSL à mon ordinateur de la même manière.
- Si la réponse n'était pas en cache au niveau de la box, celle-ci fait alors récursivement la requête à un autre serveur de noms (celui de mon fournisseur d'accès) :
  - Cette fois-ci, le déplacement physique s'est fait en ADSL (au moins au début),
  - la liaison avec le protocole ATM (par exemple, sur la partie ADSL).

- Une fois la réponse DNS reçue, on connaît l'adresse IP de la machine qui héberge `www.univ-paris8.fr` : `193.54.174.19`.
- On peut donc lui faire notre requête HTTP.

Requête HTTP

- Le but de la requête HTTP est de récupérer le contenu de la page web qu'on veut visiter.
- Cette requête va être adressée à un *serveur web*.

Création de la requête HTTP

- Votre système crée la requête HTTP.
- HTTP est un protocole haut niveau (application).
- La requête commence par un verbe (`GET`),
  suivie du chemin de la ressource demandée (`/`),
  suivie de la version du protocole (`HTTP/1.1`).
- Viennent ensuite les entêtes, dont :
  - le serveur (`www.univ-paris8.fr`),
  - l'agent utilisateur (`Mozilla (…)`),
  - potentiellement des cookies, etc.
- Vient ensuite le corps de la requête (mais celui-ci est vide pour une requête `GET`).

Encapsulation dans des paquets TCP

- Comme pour la requête DNS, on a besoin d'une enveloppe.
- Cette fois-ci cependant :
  - la requête peut être trop grande pour rentrer dans une seule enveloppe,
  - on veut être sûr d'avoir transmis correctement toutes les enveloppes,
  - et aussi être capable de les remettre dans l'ordre.

Encapsulation dans des paquets TCP

- Comme pour la requête DNS, on a besoin d'une enveloppe.
- Cette fois-ci cependant :
  - la requête peut être trop grande pour rentrer dans une seule enveloppe,
  - on veut être sûr d'avoir transmis correctement toutes les enveloppes,
  - et aussi être capable de les remettre dans l'ordre.
- On utilise donc TCP (*Transmission Control Protocol*) à la place de UDP.
- TCP permet des transmissions par *sessions* :
  - on commence par établir une connexion,
  - on transfert les données,
  - puis on ferme la connexion.
- Chaque paquet TCP contient en entête (entre autres) :
  - le port source (par exemple `57743`),
  - le port destination (`80`, le port assigné aux serveurs web).
  - un numéro de séquence,
  - un numéro d'acquittement.
- Le corps de chaque paquet comprend un morceau de la requête HTTP.

Anatomie d'un paquet TCP

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																Urgent pointer (if URG set)															
20 …	160 …	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.) …																															

Couches basses

- Comme les paquets UDP, chaque paquet TCP est encapsulé dans un paquet IP.
- Qui sont eux-mêmes encapsulés dans des paquets Ethernet (dans un premier temps).
- Qui eux transitent au niveau de la couche physique (WiFi, puis ADSL, etc.)

- Le *routage* des paquets est assuré par le protocole IP.
- Le paquet est transmis de réseaux en réseaux via des *routeurs*, jusqu'à arriver à destination.
- Un routeurs est un dispositif relié à au moins deux réseaux, avec une table de routage qui lui indique quoi faire en fonction de l'adresse de destination d'un paquet.
- Dans chaque *routeur* sur le trajet il se passe ceci :
  - Si l'adresse de destination est directement accessible, y envoyer le paquet.
  - Si il y a une entrée qui correspond à l'adresse de destination, y router le paquet.
  - Si il existe une route par défaut, y envoyer le paquet.
  - Sinon renvoyer un message d'erreur.

- Il existe différents protocoles de routage, en charge aussi du maintient à jour des routes existantes.
- À l'intérieur d'un système autonomes, c'est le protocole OSPF (*Open Shortest Path First*) qui est le plus courant.
- Entre systèmes autonomes, c'est le protocole BGP (*Border Gateway Protocol*) qui est utilisé.

Réponse HTTP

- La réponse HTTP subit le même traitement que la requête, mais depuis le serveur web.
- Une fois les différents paquets TCP arrivés et remis dans l'ordre, la réponse est recomposée et le contenu de la page web récupéré.
- Viennent ensuite les requêtes des autres composants de la page web (images, feuilles de styles, scripts, etc.).

Couches réseaux : le modèle OSI