Réseaux

Introduction à TCP/IP

Introduction à TCP/IP

Ce chapitre est une brève introduction à la pile des protocoles TCP/IP.

1. Protocoles Internet

Un protocole de communication est un ensemble de règles qui rendent les communications possibles car les intervenants sont censés les respecter.

Les protocoles définissent un sorte de langage commun que les intervenants utilisent pour se trouver, se connecter l'un à l'autre et y transporter des informations.

Les protocoles peuvent définir :

- des paramètres physiques comme des modulations, de type de supports physiques, des connecteurs, …
- le comportement d'un certain type de matériel
- des commandes
- des machines à état
- des types de messages
- des en-têtes qui comportent des informations utiles au transport

Ceux-ci sont discutés et élaborés par des organismes de standardisation.

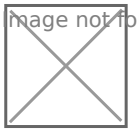
Les protocoles TCP/IP sont formalisé par l'IETF dans des documents publics qui prennent le nom de RFC ("requests for comments"). On désigne ces documents par de numéro de référence. Tous les RFCs ne sont pas nécessairement des standards … pour un peu plus de détails sur les RFCs.

Les protocoles LAN / WAN / PAN sont formalisés par l'IEEE (IEEE 802), par l'ITU, l'ANSI, …

On distinguera ces organismes de standardisation de consortium commerciaux comme la WiFi Alliance ou des organismes étatiques nationaux et internationaux de régulation comme le FCC, l'ETSI, l'IBPT, etc.

1.1. Objectif de TCP/IP

- **Communiquer**
  - à l'échelle du globe
  - de manière libérale (ouverte)
- **quel que soit**
  - le contenu
  - le support
  - les hôtes



Objectif de TCP/IP

1.2. L'Internet

L'Internet est **l'interconnexion de réseaux à l'échelle du globe**. En IPv4, l'Internet a atteint sa taille limite.

1.3. Quatre couches

Le modèle de communication TCP/IP compte quatre couches.

- **Couche Application**
  - Elle est la couche de communication qui s'interface avec les utilisateurs.
  - Exemples de protocoles applicatifs : HTTP, DNS, DHCP, FTP, …
  - S'exécute sur les machines hôtes.
- **Couche Transport : TCP**
  - Elle est responsable du dialogue entre les hôtes terminaux d'une communication.
  - Les applications utiliseront TCP pour un transport fiable et UDP sans ce service.
  - Les routeurs NAT et les pare-feu opèrent un filtrage au niveau de la couche transport.
- **Couche Internet : IP**
  - Elle permet de déterminer les meilleurs chemins à travers les réseau en fonction des adresses IPv4 ou IPv6 à portée globale.
  - Les routeurs transfèrent le trafic IP qui ne leur est pas destiné.
- **Couche Accès au réseau**
  - TCP/IP ne s'occupe pas de la couche Accès Réseau
  - Elle organise le flux binaire et identifie physiquement les hôtes
  - Elle place le flux binaire sur les support physique
  - Les commutateurs, cartes réseau, connecteurs, câbles, etc. font partie de cette couche

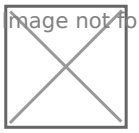
Comparatif des modèles OSI et TCP/IP

Modèle TCP/IP

Plus on monte dans les couches, plus on quitte les aspects matériels, plus on se rapproche de problématiques logicielles.

1.4. Encapsulation

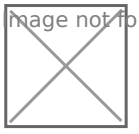
- Pour transmettre du contenu d'un ordinateur à un autre, l'utilisateur va utiliser un programme qui construit un message enveloppé par un en-tête applicatif, SMTP par exemple. Le message subit une première encapsulation.
- Le logiciel va utiliser un protocole de couche transport correspondant pour établir la communication avec l'hôte distant en ajoutant un en-tête TCP ou UDP.
- Ensuite, l'ordinateur va ajouter un en-tête de couche Internet, IPv4 ou IPv6 qui servira à la livraison des informations auprès de l'hôte destinataire. L'en-tête contient les adresses d'origine et de destination des hôtes.
- Enfin, ces informations seront encapsulées au niveau de la couche Accès qui s'occupera de livrer physiquement le message.



Encapsulation

A la réception, l'hôte récepteur réalise l'opération inverse en vérifiant les en-têtes de chaque protocole correspondant à une des couches décrites. Ce processus s'appelle la désencapsulation.

Chaque couche ajoute une information fonctionnelle au message original. A la réception, l'hôte examine chaque couche et prend une décision quant à ce trafic.



Désencapsulation

Modèle TCP/IP détaillé

Modèle TCP/IP détaillé

2. Adressage et matériel

2.1. Adressage et identifiants

Les machines et leurs interfaces disposent d'identifiants au niveau de chaque couche :

- Couche Application : Nom de domaine, par exemple : **linux.goffinet.org**
- Couche Transport : Port TCP ou UDP, par exemple : **TCP80**
- Couche Internet : Adresse IPv4 et/ou IPv6, par exemple : **192.168.150.252/24** ou **2001:db8::1/64**
- Couche Accès : adresse physique (MAC), par exemple une adresse MAC 802 : **70:56:81:bf:7c:37**

2.2. Rôles des périphériques

IP voit deux rôles :

1. Les **hôtes terminaux** : nos ordinateurs au bout du réseau
2. Les **routeurs chargés** de transférer les **paquets** en fonction de l'**adresse L3 IP (logique, hiérarchique)** de destination. Il permettent d'interconnecter les hôtes d'extrémité.

Aussi au sein du réseau local (LAN), le **commutateur** (switch) est chargé de transférer rapidement les **trames** Ethernet selon leur **adresse L2 MAC (physique)** de destination.

3. Routage IP

3.1. Domaines IP

- Deux noeuds (hôtes, interfaces, cartes réseau, PC, smartphone, etc.) doivent appartenir au même réseau, au même domaine IP pour communiquer directement entre eux.

- Quand les noeuds sont distants, ils ont besoin de livrer leur trafic à une passerelle, soit un routeur.
- D'une extrémité à l'autre, les adresses IP ne sont pas censées être modifiées (sauf NAT) par les routeurs. Par contre, le paquet est dés-encapsulé /ré-encapsulé différemment au niveau de la couche Accès au passage de chaque routeur.

Domaines IP et routage

3.2. Type d'adresses IP

Les adresses IP permettent d'identifier de manière unique les hôtes d'origine et de destination. Les routeurs se chargent d'acheminer les paquets à travers les liaisons intermédiaires.

Il existe plusieurs types d'adresses qui correspondent à plusieurs usages.

On trouve au moins trois grandes catégories :

- les adresses *unicast* : à destination d'un seul hôte
- les adresses *broadcast* (IPv4) : à destination de tous les hôtes du réseau
- les adresses *multicast* (IPv4 et IPv6) : à destination de certains hôtes du réseau.

Parmi les adresses **unicast** on distinguera :

- les adresses *non-routées* : locales ou de loopback jamais transférées par les routeurs.
- les adresses *publiques* ou *globales* : pour lesquelles les routeurs publics acheminent les paquets
- les adresses *privées* ou *unique locale* : pour lesquelles seuls les routeurs privés transfèrent le trafic (les routeurs publics ne connaissent pas de chemin pour des destinations privées).

3.3. Nécessité du NAT en IPv4

A cause de la consommation galopante d'adresses IPv4 publiques, les autorités de l'Internet on décidé de proposer des solutions :

- L'adoption d'un **protocole nouveau IPv6** corrigeant les problèmes d'IPv4 avec un conseil de transition en double pile IPv4/IPV6.

- Entre autres solution d’offrir une connectivité IPv4 avec une seule adresse publique qui cache un réseau adressé avec des blocs privés. En vue d’offrir une connectivité globale (publique) à des hôtes privés, une des solutions est la traduction du trafic (NAT). Une autre est la mise en place d’un proxy applicatif. Ces opérations tronquent le trafic IP d’origine, génère de la charge sur les ressources nécessaires à la traduction à la réécriture du trafic dans un sens et dans un autre, ce qui n’est pas sans poser problème et génère un certain coût.

3.4. Transition IPv6

Il n’est plus envisagé de manière crédible traduire le nouveau protocole dans l’ancien, IPv6 dans IPv4.

Par contre, l’inverse, soit l’ancien dans le nouveau, IPv4 dans IPv6 annonce la prochaine étape de transition. Les solutions NAT64/DNS64 offrent la possibilité de déploiement “IPv6 Only”.

3.5. NAT et pare-feu

On a tendance à confondre les fonctionnalités NAT avec celles du pare-feu.

Même si il est probable que le même logiciel prenne en charge les deux fonctions, ces procédures sont distinctes. Alors que le NAT permet de traduire le trafic, il ne protège en rien.

Cette fonction est prise en charge par le pare-feu à état qui arrête les connexions non sollicitées sur le réseau à protéger. Le proxy (mandataire) quant à lui aura d’autres fonction que la traduction telle que le contrôle du trafic qui lui est livré, avec des mécanismes de cache, d’authentification et de transformation du trafic dont aussi la traduction.

Dans tous les cas, c’est la fonction pare-feu qui protège des tentatives de connexions externes.

3.6. Adressage IPv4

Une adresse IPv4 est un identifiant de 32 bits représentés par 4 octets (8 bits) codés en décimales séparées par des points.

tableau adresses IPv4

Types d’adresses IPv4	Plages	Remarques

Adresses Unicast	`0.0.0.0` à `223.255.255.255`	Pour adresser les routeurs des réseaux d'extrémité et les ordinateurs accessible à travers l'Internet public
Adresses Unicast privées	`10.0.0.0/8` (Classe A), `172.16.0.0/12` (Classe B), `192.168.0.0/16` (Classe C)	L'objectif de départ des adresses IPv4 privées est d'identifier des connexions privées (sur des réseaux privés). Mais en situation de carence d'adresses IPv4 publiques, les adresses privées permettent aussi d'adresser des réseaux "clients" d'extrémité. Ceux-ci arrivent à placer du trafic sur l'Internet public grâce à des mécanismes de traduction d'adresses (NAT/PAT). Par définition, les adresses privées ne trouvent pas de destination sur l'Internet public.
Adresses Multicast	`224.0.0.0` à `239.255.255.255`	Ces adresses identifient plusieurs interfaces en général sur un réseau contrôlé.

Le masque de réseau lui aussi noté en décimal pointé indique avec les bits à 1 la partie réseau partagée par toutes les adresses d'un bloc. Les bits à 0 dans le masque indiquent la partie unique qui identifie les interfaces sur la liaison. Un masque de réseau est un suite homogène de bits 1 et puis de bits seulement à zéro. Il s'agit donc d'un masque de découpage de blocs homogènes d'adresses IP contigües. Ces blocs communiquent entre eux grâce à des "routeurs", sortent d'intermédiaire d'interconnexion, dont le rôle est justement de transférer du trafic qui ne leur est pas destiné. Les routeurs identifient un point géographique, lieux d'interconnexion des liaisons. Les routeurs IPv4 remplissent souvent des fonctions de traduction.

Par exemple, `192.168.1.25 255.255.255.0` indique un numéro de réseau (première adresse) `192.168.1.0` et un numéro de Broadcast `192.168.1.255`. Toutes les adresses comprises entre ces valeurs peuvent être utilisée par les interfaces attachées à une même liaison (un même switch).

Le masque peut aussi respecter la notation CIDR qui consiste à écrire le nombre de bits à 1 dans le masque. Soit dans l'exemple précédent une écriture de type `/24` pour les 24 bits à 1 (`255.255.255.0`). Le masque CIDR s'impose en IPv6 (imaginez des masques de 128 bits en hexadécimal). Cette notation est certainement plus intuitive et plus simple à encoder ou à lire.

A cause du manque d'espace IPv4 disponible, on trouve souvent des masques qui chevauchent les octets, comme par exemple `/27` (`255.255.255.224`) qui offre 32 adresses (5 bits à zéro dans le masque) ou `/30` (`255.255.255.252`) qui offre 4 adresses (2 bits à zéro dans le masque), sans oublier la première (numéro du réseau) et la dernière (l'adresse de diffusion, *broadcast*) que l'on ne peut attribuer aux interfaces. On constate que si le bits à 1 dans le masque indiquent le découpage de cet espace codé sur 32 bits, les bits à zéro restants donnent l'étendue du bloc.

L'utilitaire `ipcalc` calcule pour nous les adresses IP :

```
ipcalc --help
```

```
Usage: ipcalc [OPTION...]
  -c, --check         Validate IP address for specified address family
  -4, --ipv4          IPv4 address family (default)
  -6, --ipv6          IPv6 address family
  -b, --Broadcast     Display calculated Broadcast address
  -h, --hostname      Show hostname determined via DNS
  -m, --netmask       Display default netmask for IP (class A, B, or C)
  -n, --network       Display network address
  -p, --prefix        Display network prefix
  -s, --silent        Don't ever display error messages

Help options:
  -?, --help          Show this help message
  --usage             Display brief usage message
```

Par exemple :

```
ipcalc -n -b -m 192.167.87.65/26
```

```
NETMASK=255.255.255.192
BROADCAST=192.167.87.127
NETWORK=192.167.87.64
```

On trouvera un utilitaire plus explicite avec `sipcalc`, il est en dépôt chez Debian/Ubuntu. Sous Centos/RHEL, il sera nécessaire de le compiler sois-même.

```
sipcalc -I ens33
```

```
-[int-ipv4 : ens33] - 0

[CIDR]
Host address            - 172.16.98.241
Host address (decimal)  - 2886755057
Host address (hex)      - AC1062F1
Network address         - 172.16.98.0
Network mask            - 255.255.255.0
Network mask (bits)     - 24
Network mask (hex)      - FFFFFF00
```

```
Broadcast address - 172.16.98.255
Cisco wildcard - 0.0.0.255
Addresses in network - 256
Network range - 172.16.98.0 - 172.16.98.255
Usable range - 172.16.98.1 - 172.16.98.254
```

3.7. Adressage IPv6

Les adresses IPv6 sont des identifiants uniques d'interfaces codés sur 128 bits et notés en hexadécimal en 8 mots de 16 bits (4 hexas) séparés par des ":".

Par exemple, pour l'adresse `2001:0db8:00f4:0845:ea82:0627:e202:24fe/64` dans son écriture extensive :

```
2001:0db8:00f4:0845:ea82:0627:e202:24fe
---- ---- ---- ---- ---- ---- ---- ----
16b  16b  16b  16b  16b  16b  16b  16b
------------------- -------------------
   Préfixe          Interface ID
```

Ecriture

Voici l'écriture résumée :

```
2001:0db8:00f4:0845:ea82:0627:e202:24fe
2001:-db8:--f4:-845:ea82:-627:e202:24fe
2001:db8:f4:845:ea82:627:e202:24fe
```

Ou encore l'adresse `fe80:0000:0000:0000:bb38:9f98:0241:8a95` peut être résumée en `fe80::bb38:9f98:241:8a95`.

Configuration

Ces adresses peuvent être configurées :

- De manière automatique (autoconfiguration), sans état
- De manière dynamique avec serveur DHCPv6, avec état
- De manière automatique et de manière dynamique
- De manière statique

Une interface IPv6 peut accepter plusieurs adresses et dans des préfixes distincts. L'idée est d'améliorer les politiques de routage et de filtrage en fonction de ces adresses.

Adresses Unicas Globale et Link-Localt

Un premier exemple avec qui illustre des adresses Unicast :

```
# ip -6 a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 2001:dc8:f4:845:ea82:627:e202:24fe/64 scope global noprefixroute dynamic
       valid_lft 1209585sec preferred_lft 604785sec
    inet6 fe80::bb38:9f98:241:8a95/64 scope link
       valid_lft forever preferred_lft forever
```

On y trouve deux interfaces :

- **`lo`** qui prend la seconde adresse de l'espace IPv6. `::1/128` ne joint qu'elle-même directement comme adresse de "Loopback".
- **`eth0`** : qui prend deux adresses IPv6 :
- L'une toujours présente se reconnait par son préfixe `fe80::/10`. Ces adresses sont uniques sur chaque interface et communiquent uniquement avec d'autres interfaces sur le lien local (des *voisins*) d'où leur "Link-Local Address".
- L'autre adresse avec le préfixe `2001:dc8:f4:845::/64` indique une adresse publique, soit routable, joignable par Internet et permettant de placer du trafic sur Internet. On l'appelle une adresse GUA "Gobal Unicast Address". Elles s'identifient dans le bloc `2000::/3`.

On notera que :

- Les adresses GUA et link-local disposent d'un masque `/64`. C'est le masque par défaut de toute interface IPv6 d'un point terminal (endpoint). On propose aux connexions d'entreprise le routage d'un bloc `/48` qu'elle pourrait découper sur les 16 bits suivant pour obtenir 65536 blocs `/64` à adresser sur ses réseaux.
- Les adresses GUA et link-local disposent de deux valeurs de durée de vie : `valid_lft` et `preferred` qui déterminent la durée de leur validité.

Adresses Mulitcast

Les groupes Multicast dans lesquels les interfaces sont inscrites.

```
# ip -6 maddress
1: lo
    inet6 ff02::1
    inet6 ff01::1
2: eth0
    inet6 ff02::1:ff02:24fe
    inet6 ff02::1:ff41:8a95
    inet6 ff02::1
    inet6 ff01::1
```

Cela signifie que les interfaces acceptent le trafic dont l'adresse de destination est une de ces adresses qui sont partagées par plusieurs interfaces sur plusieurs hôtes IPv6. A priori, le Multicast n'est pas transféré par les routeurs ; les commutateurs le traitent comme du *Broadcast* (Diffusion).

Ceci précisé, le Multicast IPv6 désigne finement le trafic sur base de la portée et d'une destination. `ff02::1` signifie "tous les hôtes sur le réseau local"; `ff01::1` signifie "tous les hôtes sur le noeud local". Les adresses `ff02::1:ff02:24fe` `ff02::1:ff41:8a95` servent de destination au trafic Neighbor Discovery (voir plus bas) qui demande l'adresse physique de livraison inconnue d'une adresse IPv6 à joindre, donc connue d'avance. On reconnait après le préfixe `ff02::1:ff/104` les 24 derniers bits `02:24fe` et `ff41:8a95` de chacune des deux adresses `2001:db8:f4:845:ea82:627:e202:24fe` et `fe80::bb38:9f98:241:8a95`. Il y aura une adresse Multicast différente pour chaque adresse GUA, ULA (Unique Local) ou link-local aux 24 derniers bits différents.

La table de routage indique que le trafic pour l'Internet (`default`) est livré à adresse link-local du routeur `fe80::22e5:2aff:fe1b:656a`.

```
# ip -6 route
2001:db8:f4:845::/64 via fe80::22e5:2aff:fe1b:656a dev ens33  proto ra  metric 100
fe80::22e5:2aff:fe1b:656a dev ens33  proto static  metric 100
fe80::/64 dev ens33  proto kernel  metric 256
default via fe80::22e5:2aff:fe1b:656a dev ens33  proto static  metric 100
```

Adresses Unique Local (Unicast)

Un second exemple révèle l'existence d'adresses privées IPv6 dans le préfixe `fd00::/8`, ici sur l'interface `eth0` :

```
# ip -6 a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 fd00:101::1a8/128 scope global dynamic
       valid_lft 3021sec preferred_lft 3021sec
    inet6 fe80::5054:ff:fe53:c52c/64 scope link
       valid_lft forever preferred_lft forever
```

Le préfixe `fd00::/8` nous renseigne un bloc d'adresses privées dont les 40 bits suivants sont censés être générés de manière aléatoire, ce qui est une obligation mais qui selon moi reste impossible à vérifier. Le préfixe suggéré offre alors un bloc `/48` dont on peut compenser 16 bits pour offrir 65536 blocs `/64`.

Ce type d'adresse privée à préfixe unique répond au problème des blocs IPv4 privés qui sont partagés par l'ensemble des postes de travail ou des périphériques mobiles dans le monde. Cet état de fait en IPv4 rend les connexions de bout en bout complexes à établir à travers un Internet public (notamment à travers des réseaux VPN, pour du partage peer-to-peer pour de la téléphonie Internet, etc.). Ce ne sont pourtant pas les solutions qui manquent mais celle-ci ne font que renforcer la mise en place de bricolages qui dégradent la connectivité.

Transition IPv6

L'Internet des Objets et les services en nuage annoncent une croissance exponentielle des besoins en connectivité auxquels seul IPv6 pourra répondre. L'Internet est entré en juin 2012 dans une très longue phase de transition duale d'IPv4 à IPv6. On pourrait encore parler d'IPv4 d'ici 2025 alors qu'IPv6 dominera les communications TCP/IP. Plusieurs explications apportent de l'eau au moulin de cette longue phase : une répartition des coûts sur tous les acteurs; malgré son aspect logique une migration d'infrastructure est nécessaire à l'échelle globale, sans compter les serveurs publics qui ne seront jamais migrés en IPv6 et qu'il faudra maintenir en IPv4 ...

Synthèse sur les adresses IPv6

En résumé pour IPv6, on peut retenir ce schéma :

Adresses IPv6

Etant donné la multiplication des sources et des destinations potentielles qu'offrent le protocole IPv6, on sera attentif aux configurations des pare-feux.

4. Protocoles de résolution d’adresses et de découverte des hôtes

- Afin d’encapsuler un paquet IP dans une trame, l’hôte d’origine a besoin de connaître l’adresse physique (MAC) de la destination.
- En IPv4, c’est le protocole ARP (Address Resolution Protocol) qui remplit cette fonction. Les hôtes IPv4 maintiennent une table appelée cache ARP.
- En IPv6, c’est le protocole ND (Neighbor Discovery), sous-protocole IPv6, qui reprend cette fonction. Les hôtes IPv6 maintiennent une table appelée table de voisinage.

4.1. Commandes utiles

- Table ARP sous Windows et Linux : `arp -a`
- Table ARP sous Linux : `ip -4 neigh`
- Table de voisinage sous Linux : `ip -6 neigh`
- Table de voisinage sous Windows : `netsh interface ipv6 show neighbors`

4.2. ARP (Address Resolution Protocol)

Au moment de l’encapsulation d’un paquet IPv4 dans une trame Ethernet ou Wi-fi, l’hôte émetteur connaît d’avance l’adresse IP de destination. Mais comment peut-il connaître son adresse physique correspondante (l’adresse MAC de destination par exemple) afin de placer le trafic sur le support ?

Un hôte TCP/IP ne peut connaître l’adresse de destination sans qu’elle ne s’annonce elle-même de *manière gratuite* ou de *manière sollicitée*.

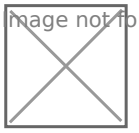
Dans le but de maintenir une correspondance entre des adresses IP à joindre et leur adresses physiques de destination, les hôtes TCP/IP entretiennent une “**table ARP**” pour les adresses IPv4 et une “**table de voisinage**” pour les adresses IPv6.

ARP est un protocole indépendant d’IPv4 qui offre ce service de résolution d’adresses.

En IPv6, ce sont des paquets ICMPv6 appelés Neighbor Discovery (ND) qui sont utilisés selon un mode sensiblement différent. En IPv6, les fonctions d’informations et de contrôle (ICMPv6) ont été améliorées et renforcées.

- La requête ARP émane en Broadcast et la réponse est envoyée en unicast. ND (IPv6) aura un fonctionnement similaire en utilisant une adresse Multicast spéciale en lieu et place du

Broadcast.

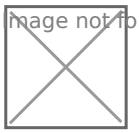


ARP (Address Resolution Protocol)

- Capture : https://www.cloudshark.org/captures/e64eaac12704?filter=arp

4.3. ND (Neighbor Discovery)

- Découverte de voisin sollicitée



ND (Neighbor Discovery)

- Capture : https://www.cloudshark.org/captures/85556fc52d28

5. Protocole de résolution de noms

- Au niveau protocolaire, seuls les adresses IP sont utilisées pour déterminer les partenaires d'une communication.
- Mais dans l'usage courant d'Internet, on utilise des noms pour joindre des machines sur le réseau : c'est plus facile à manipuler que des adresses IP.
- Le protocole et le système DNS permet de résoudre des noms en adresses IP.
- DNS est une sorte de service mondial de correspondance entre des noms et des adresses IP. DNS utilise le port UDP 53.

Recherche DNS

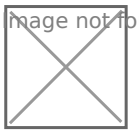
Source : https://visual.ly/help-understanding-dns-lookups

6. Protocoles d’attribution d’adresses

- Avant de pouvoir émettre du trafic TCP/IP, une interface doit disposer au minimum d’une adresse IP et de son masque et, éventuellement d’autres paramètres (passerelle par défaut, résolveur DNS, etc.).
- En IPv4, c’est DHCP qui permet d’attribuer ces paramètres à une interface qui le demande. DHCP maintient un état des adresses attribuées par un mécanisme de bail (à durée déterminée).
- En IPv6, le comportement par défaut est l’autoconfiguration des interfaces mais la version actualisée de DHCPv6 fournit un service géré des adresses.

6.1. DHCP (IPv4)

- La procédure d’attribution d’adresses en DHCP (IPv4) consiste en l’échange de 4 messages sur les ports UDP 67 et 68.
- Le premier message DHCP émane du client en Broadcast.



Echange entre client et serveur DHCP

Capture de trafic DHCP : https://www.cloudshark.org/captures/c109b95db0af

Dans un session typique, le client diffuse (Broadcast) un message DHCPDISCOVER sur son segment local. Le client peut suggérer son adresse IP et la durée du bail (lease). Si le serveur est sur le même segment, il peut répondre avec un message DHCPOFFER qui inclut une adresse IP valide et d’autres paramètres comme le masque de sous-réseau. Une fois que le client reçoit ce message, il répond avec un DHCPREQUEST qui inclut une valeur identifiant le serveur (pour le cas où[1] il y en aurait plusieurs). Cette valeur l’identifie de manière certaine et décline implicitement les offres des autres serveurs. Une fois le DHCPREQUEST reçu, le serveur répond avec les paramètres définitifs de configuration par un message DHCPACK (si le serveur a déjà assigné l’adresse IP, il envoie un DHCPNACK).

Si le client détecte que l’adresse IP est déjà utilisée sur le segment, il envoie un DHCPDECLINE au serveur et le processus recommence.

Si le client reçoit un message DHCPNACK du serveur après un DHCPREQUEST, le processus recommence également.

Si le client plus besoin d’une adresse IP, il envoie un DHCPRELEASE au serveur.

Si le client veut étendre la durée du bail qui lui est allouée, il envoie un DHCPREQUEST au serveur dans lequel le champ ‘ciaddr’ correspondra à son adresse IP actuelle. Le serveur répondra avec un DHCPACK comprenant la nouvelle durée du bail.

6.2. DHCPv6

Cette matière est abordée dans le chapitre Services d’infrastructure.

7. Interaction des protocoles

- Avant qu’une interface puisse envoyer du trafic faut-il :
- qu’elle ait obtenu une adresse IP statique ou dynamique (DHCP en IPv4 ou autoconfiguration/DHCPv6 en IPv6);
- qu’elle ait résolu le nom de l’hôte destinataire en adresse IP (DNS sur IPv4 ou sur IPv6) ;
- qu’elle ait obtenu l’adresse du livraison physique de la destination locale ou de la passerelle par défaut si la destination n’est pas locale (ARP en IPv4 ou ND en IPv6).

8. Autres protocoles de gestion

D’autres protocoles de gestion importants se rencontreront dans les réseaux TCP/IP, à titre d’exemples :

- ICMP qui permet en IPv4 et en IPv6 d’obtenir du diagnostic IP (`ping` et `traceroute`).
- NTP qui permet de synchroniser les horloges des hôtes sur le réseau.
- SNMP qui permet de collecter des informations sur le matériel à travers le réseau.
- SSH qui permet de monter une console distante à travers TCP/IP.
- Le routage IP met en oeuvre du NAT sur les passerelles des réseaux privés pour offrir une connectivité à l’Internet.
- …