1. QU'EST-CE QU'UNE ADRESSE IP

Internet Protocol, RFC 791

Il s'agit d'une adresse permettant l'**identification** d'une **machine** sur un **réseau**. Il convient de préciser qu'une telle adresse ne désigne pas un hôte mais une **carte réseau**. *Un hôte se trouvant sur 2 réseaux devra avoir 2 adresses IP. Les routeurs peuvent avoir plusieurs interfaces, donc plusieurs adresses IP.*

Ce **protocole** est utilisé en **couche 3** d'une architecture réseau. **La couche 3 dite couche réseau est la couche permettant d'interconnecter les réseaux entre eux.**

Un protocole est un langage permettant aux machines de dialoguer entre elles.

Actuellement, on utilise le protocole **IPv4** pour définir une adresse IP sur **32 bits** (4 octets). Il n'y aura bientôt plus d'adresses disponibles dans ce format et nous devrons bientôt utiliser le protocole **IPv6** qui définit ses adresses sur **128 bits**. On utilise la **notation décimale pointée** pour écrire une adresse IP. *Dans ce format, chacun des 4 octets est représente par un nombre décimal compris entre 0 et 255 et chaque octet est séparé du suivant par un point.*

Une partie de l'adresse IP représente **l'adresse du réseau**, l'autre partie **l'adresse de la machine**. La partie réseau a la même valeur pour tous les hôtes d'un même réseau. Un réseau correspond donc à un bloc contigu d'espace d'adressage IP. C'est ce qu'on appelle un **préfixe**.

2. QU'EST-CE QU'UN MASQUE DE SOUS-RÉSEAU

Le masque de **sous-réseau** est **indissociable** de l'adresse **IP**. Le **masque** indique quelle est la **partie réseau** d'une IP et quelle est la **partie machine**. Les **bits** a **1** dans le masque représentent la **partie réseau** de l'IP, et les **bits** a **0** correspondent a la **partie machine**.

```
Masque :    255.255.0.0              11111111.11111111.00000000.00000000
IP :        192.168.0.1              11000000.10101000.00000000.00000001
            Partie reseau                                  Partie machine
```

L'exercice est facile quand la coupure entre les 2 parties se fait entre 2 octets, c'est plus compliqué quand ça se passe au milieu d'un octet (et c'est souvent le cas).

```
Masque :    255.255.240.0            11111111.11111111.11110000.00000000
IP :        192.168.0.1              11000000.10101000.00000000.00000001
```

Les valeurs prises par les octets dans 1 masque sont spécifiques : **contigüité des bits**. Les **1** sont à **gauche** et les **0** à **droite**, on ne peut pas les mélanger. On retrouvera donc toujours les mêmes valeurs pour les octets d'un masque.

```
00000000    0
10000000    128
11000000    192
11100000    224
11110000    240
11111000    248
11111100    252
11111110    254
11111111    255
```

Toutes les **machines** d'un **même réseau** ont la particularité d'avoir tous les **bits** de leur **partie réseau identiques**.

3. QU'EST-CE QUE LE SOUS-RÉSEAU D'UNE IP AVEC UN MASQUE DE SOUS-RÉSEAU

```
Table logique & (ET binaire)
0 & 0       0
0 & 1       0
1 & 0       0
1 & 1       1
```

Un **sous-réseau** est une **subdivision logique** d'un **réseau** de taille plus importante. Le masque de sous-réseau permet de distinguer la partie de l'adresse commune à tous les appareils du sous-réseau et celle qui varie d'un appareil a l'autre. Cela correspond à un réseau local sous-jacent.

```
NETWORK PREFIX      HOSTNUMBER
NETWORK PREFIX      SUBNET HOST
                    NUMBER NUMBER
```

La subdivision en sous-réseau permet de **limiter** la **propagation** des ***broadcasts***, dont la gestion est *couteuse* en *bande passante* et en ressource au niveau des *commutateurs* réseau.

On obtient l'**adresse** du **réseau** en appliquant l'operateur **ET** bit à bit **&** sur l'adresse **IPv4** et le **masque**.
L'**adresse** de l'**hôte** (machine) à l'intérieur du sous-réseau est obtenue en appliquant **&** entre l'adresse **IPv4** et le **complément à un** du **masque**.

```
Adresse du reseau
192.168.1.2 & 255.255.255.0 == 192.168.1.0
11000000.10101000.00000001.00000010 &
11111111.11111111.11111111.00000000 ==
11000000.10101000.00000001.00000000

Adresse de l'hote
192.168.1.2 & 0.0.0.255 == 0.0.0.2
11000000.10101000.00000001.00000010 &
00000000.00000000.00000000.11111111 ==
00000000.00000000.00000000.00000010
```

Pour obtenir le nombre d'adresses dans un réseau :

$$2^{\text{nb-de-0-dans-le masque}}$$

Subdiviser un réseau en sous-réseau consiste (entre autres) à rajouter des bits 1 au masque. En notant **r** le **nb de bits à 1** dans le masque **réseau** et **s** le **nombre de bits à 1** dans le masque de sous **réseau**, le nombre de réseaux possibles est donné par 2^{s-r} et le nombre d'hôtes du sous-réseau est $2^{32-s}-2$ sauf pour **/31** (seulement 2 hôtes possibles) et **/32** ($2^{32-32}-0=2^0-0=1-0=1$). Pour pouvoir utiliser une fonction de calcul applicables à tous les masques, on utilisera la fonction de ***Heaviside*** ($H(x)$ → $2^{32-s}-2*H(30-s)$)

Ex :

Soit le réseau 192.44.78.0/24, on souhaite le découper en 4 réseaux. On réserve les 2 premiers bits de l'identifiant machine pour identifier les nouveaux sous-réseaux. Toute adresse IP aura 24 + 2 bits en commun.

Le masque serait alors :

```
255.255.255.192                    11111111.11111111.11111111.11000000
```

Les sous-réseaux seront :

192.44.78.0/26	192.44.78.0 à 192.44.78.63
192.44.78.64/26	192.44.78.64 à 192.44.78.127
192.44.78.128/26	192.44.78.128 à 192.44.78.191
192.44.78.192/26	192.44.78.192 à 192.44.78.255

Deux hôtes, bien qu'appartenant au même réseau logique, s'ils sont places dans deux sous-réseaux logiques différents ne pourront communiquer entre eux que par l'intermédiaire d'un routeur. Cette solution est très commode pour les réseaux d'entreprises constitués de réseaux locaux distants.

Découpage de plage, méthode dite magique

- Pour utiliser cette méthode, il faut déterminer le **nombre magique**. Il faut repérer l'**octet significatif** du masque : **c'est celui où la séparation se fait entre les adresses.**
  nombre magique = 256 – octet significatif masque
- Le nombre magique permet de calculer instantanément la 1ere et la dernière adresse de plage. Cela se joue sur les multiples de ce nombre magique. Pour obtenir le premier multiple, on va diviser l'octet correspondant, 168 par le nombre magique, 32. On récupère la partie entière de ce résultat et on le multiplie par le nombre magique, cela nous nous le premier multiple recherché. On ajoute un au quotient précédemment obtenu et on le multiplie par le nombre magique, nous obtenons le multiple supérieur a celui correspondant.
  - La **1ere adresse** sera le **multiple inferieur ou égal à l'octet correspondant** dans l'adresse IP à l'octet significatif du masque
  - La **dernière** sera le **multiple suivant – 1**

Ex :

Masque : 255.224.0.0 Adresse IP : 192.168.0.1
256 -224 (*octet significatif*) = 32 (*nombre magique*)

Pour obtenir le premier multiple, on va diviser l'octet correspondant, 168 par le nombre magique, 32. On recupere la partie entiere de ce resultat et on le multiplie par le nombre magique, cela nous nous le premier multiple recherché
Le premier multiple de 32 plus petit que 168 est 160

Pour obtenir le multiple suprerieur, on reprend la partie entiere du resultat de la division precedente, on ajoute un et on ;ultiplie cette somme par le nombre magique.
Le premier multiple de 32 plus grand que 168 est 192, moins 1 = 191

La plage va de l'adresse 192.160.0.0 à 192.191.255.255

4. QU'EST-CE QUE L'ADRESSE DE BROADCAST ET CELLE DE SOUS-RÉSEAU D'UN SOUS-RÉSEAU

La **1ere adresse d'une plage est l'adresse du réseau**. On ne peut pas l'utiliser pour une machine.
ⓘ Une adresse qui finit par 0 n'est pas forcement une adresse de réseaux. Les adresses réseau sont toujours paires (à cause des 0 binaires qu'elles contiennent dans la partie machine de leur adresse IP).

La **dernière adresse d'une plage est une adresse de broadcast**. Elle ne peut être utilisée par une machine. *Elle sert à identifier toutes les machines du réseau*.
ⓘ Une adresse qui finit par 255 n'est pas forcement une adresse de broadcast. Toutes les adresses de broadcast sont impaires (à cause des 0 binaires contenus dans la partie machine).

5. QUELLES SONT LES DIFFÉRENTES MANIÈRES DE REPRÉSENTER UNE IP AVEC UN MASQUE DE SOUS-RÉSEAU

On utilise la notation décimale pointée avec la notation **CIDR** (*Classless Inter Domain Routing*) pour représenter une IP avec masque. On pourrait aussi employer la notation binaire, mais elle est rarement employée car très longue. On pourrait aussi représenter l'IP et le masque de façon décimale. Mais l'intérêt de la question est l'introduction à la notation CIDR.

```
192.44.78.0 /26
```

On suffixe l'adresse IP avec un slash derrière lequel on indique le **nombre de bits** de poids fort (**à 1**) dans le **masque** de réseau.

6. QUELLE EST LA DIFFÉRENCE ENTRE UNE IP PRIVÉE ET UNE IP PUBLIQUE

Les adresses privées ne peuvent pas être utilisées par Internet, elles sont définies par la RFC (Request For Comment) 1918 sur le réseau local.

La RFC définit la plage précise de ces adresses afin de ne pas utiliser / télescoper / empiéter sur une adresse publique déjà utilisée par quelqu'un d'autre.

Ex :

Si on s'approprie une plage réseau de quelqu'un pour une utilisation privée, le jour où on tentera d'accéder au site ayant cette adresse, cela ne marchera pas, notre machine évaluera qu'on est sur notre propre réseau et n'arrivera pas à le joindre.

Les adresses privées définies par la RFC :

```
10.0.0.0 / 255.0.0.0                          10.0.0.0 à 10.255.255.255
172.16.0.0 / 255.240.0.0                    172.16.0.0 à 172.31.255.255
192.168.0.0 / 255.255.0.0                 192.168.0.0 à 192.168.255.255
```

7. QUELLES SONT LES DIFFÉRENTES CLASSES D'IP

Les classes sont des regroupements de réseau de même taille : ils ont le même nombre d'hôtes maximum.

Classe	Masque	Adresses	Nb de réseaux	Nb d'hôtes	Commentaires
A	**255.0.0.0**	**1.0.0.0** à **126.255.255.255**	126	16 777 214	la valeur du premier octet est comprise entre **1** et **126** ; le premier bit de poids fort est égal à **0**. *Le premier octet représente le réseau et les 3 suivants l'hôte.*
B	**255.255.0.0**	**128.0.0.0** à **191.255.255.255**	16384	65534	la valeur du premier octet est comprise entre **128** et **191** ; *les deux premiers bits de poids fort sont égal à 10. Les deux premiers octets représentent le réseau et les 2 suivants l'hôte.*
C	**255.255.255.0**	**192.0.0.0** à **223.255.255.255**	2 097 152	254	la valeur du premier octet est comprise entre **191** et **223** ; les trois premiers bits de poids fort sont egal à **110**. *Les trois premiers octets représentent le réseau et le suivant l'hôte.*
D	**240.0.0.0**	**224.0.0.0** à **239.255.255.255**	Adresse unique	Adresses unique	la valeur du premier octet est comprise entre **224** et **239** ; les trois premiers bits de poids fort sont egal à **111**. *Les zones d'adresses sont dédiées aux services de multidiffusion vers des groupes d'hôtes (host groups).*
E	Non défini	**240.0.0.0** à **255.255.255.255**	Adresses uniques	Adresses uniques	la valeur du premier octet est comprise entre **240** et **255** ; *ce sont des zones réservées aux expérimentations, elles ne doivent pas être utilisées pour des hôtes ou des groupes d'hôtes.*

Il existe deux adresses particulières :

127.0.0.0 Réservée pour la communication en boucle (on en a une par machine, localhost).

0.0.0.0 Utilisée pour définir une route par défaut sur un routeur.

Les classes d'IP ne sont plus utilisées au profit de la notation CIDR. Les adresses sont désormais distribuées par bloc sans tenir compte de leur classe originelle. C'est l'IANA qui est charge de la distribution des adresses IP.

8. QU'EST-CE QUE LE TCP (TRANSMISSION CONTROL PROTOCOL)

Protocole, de couche 4 transport, **fiable avec connexion qui garantit la livraison sans erreur** a n'importe quel hôte de l'interréseeau d'un flot d'octets émis par une machine. Il **segmente le flot d'octets entrant** en messages discrets et transmet chacun d'eux à la couche Internet. À l'arrivée, le processus TCP destinataire réassemble les messages reçus en un flot de sortie. TCP **assure aussi un contrôle de flux** pour éviter qu'un émetteur rapide ne submerge un récepteur lent de plus de messages qu'il ne peut en traiter. TCP transmet un **segment TCP**.

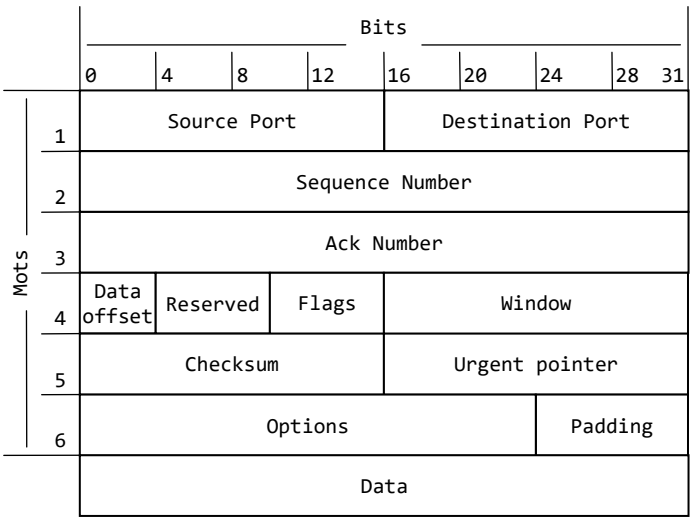


Figure 1 - Segment TCP
Trame Ethernet > Datagramme IP > Segment TCP

Chaque paquet envoyé devra être acquitté par le receveur : **protocole connecté**. *Tant que l'on n'a pas la confirmation du receveur, on renvoie un paquet.* TCP ne prend pas en charge les **diffusions**.

On aura beaucoup d'informations dans l'en-tête TCP pour parvenir à suivre une connexion correctement.

Les **3 premiers paquets envoyés servent à établir la communication**. Ce sont des paquets vides qui ne sont la que pour confirmer que la communication est bien établie. TCP utilisera son en-tête pour dire si un paquet correspond à une demande de connexion ou si c'est un paquet normal. Comme ces paquets seront vides, il leur faudra dire par leur en-tête s'il s'agit d'une demande de connexion, d'une réponse ou d'un acquittement (aknowledgment). On utilise des **flags** (*bits pouvant prendre la valeur de 0 ou 1, selon leur état*) pour renseigner le type de message TCP envoyé.

Le **premier paquet** est une demande de synchronisation **SYN** avec un numéro de séquence x. Un serveur recevant une demande **SYN** doit normalement répondre qu'il est disponible pour communiquer avec le client : il renvoie un acquittement ACK derrière le flag demandant sa synchronisation SYN → **SYN + ACK** en retour. *En effet, le serveur établit sa première communication avec le client*. En fait il va renvoyer un numéro d'acquittement, qui correspond au numéro de la séquence reçue du client + 1, soit ici $x+1$ et un numéro de séquence y. Les flags SYN et ACK sont positionnés et les numéros mis dans les pointeurs Sequence Number et ACK Number.

TCP établit une connexion pour chaque sens de communication.

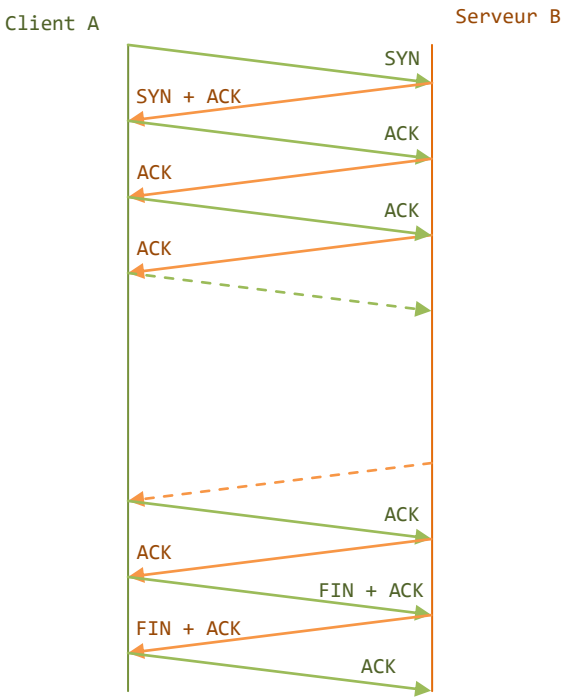
Le client doit accepter la demande de connexion faire par le serveur en renvoyant un paquet avec le flag **ACK** au serveur. Il acquitte la réponse en retournant également un numéro d'acquittement egal au numéro de séquence envoyée par le serveur + 1, ici $y+1$ et un numéro de séquence egal au numéro d'acquittement envoyé par le serveur, ici $x+1$.

Cette technique est appelée **Three Way Handshake** (*poignée de main tripartite*).

Une fois la **communication établie**, les applications peuvent s'échanger des paquets autant qu'elles le veulent. Le principe est alors de positionner le flag **ACK** dans les échanges après que la communication soit établie pour acquitter la réception des paquets précédents.

Quand la **communication** est **finie**, il faut la terminer proprement sans quoi les ports mobilises a cet effet seraient inutilisables pour de futures communications éventuelles. La procédure ressemble à celle de demande de connexion : à la place de SYN, on utilise **FIN**.

Le client envoie son flag **FIN** accompagné de **ACK** pour acquitter le paquet précédent. **Le serveur répond en demandant à son tour la fermeture de la communication et en acquittant la réception de demande de fin de communication**. Enfin le client doit acquitter la demande du serveur, tant que cela n'est pas fait la communication reste ouverte. Les ressources machines ne sont libérées qu'après ce **ACK** final.



Il existe 6 flags différents : **SYN**, **ACK**, **FIN**, **RST**, **PSH** et **URG**.

Chaque octet envoyé (de données) devant être acquitté, s'il était constaté une **incohérence** entre les données reçues et les données envoyées, la **communication** est considérée comme **anormale** et la machine qui s'en rend compte doit prévenir l'autre pour terminer cette communication anormale et mettre en place une nouvelle communication. La machine constatant l'anomalie envoie le flag **RST**, au lieu de la séquence FIN + ACK (x2), ACK.

Si on envoie un paquet **SYN** sur le **port fermé** d'une machine, celle-ci doit retourner **RST**, pour indiquer que le port demandé n'est pas en écoute.

PSH indique que la **donnée doit être transmise de suite** et non stockée dans un buffer, en attendant que celui-ci se remplisse a sa capacité maximale pou être transmis.

URG n'est plus vraiment utilisé par les protocoles modernes, on le conserve pour des raisons de rétrocompatibilité. Il sert à indiquer qu'**une donnée transmise dans le paquet est urgente** et devrait être priorisée. Un pointeur est activé et il indique combien de data dans le segment est urgente depuis le 1er octet.

9. QU'EST-CE QUE L'UDP (USER DATAGRAM PROTOCOL)

Protocole, de couche 4 transport, **orienté datagramme**, **non fiable sans connexion** qui permet aux applications d'assurer elles-mêmes le séquençage et le contrôle de flux au lieu de faire appel à TCP. Ce protocole est largement utilisé par les interrogations ponctuelles de type demande-réponse dans les environnements client-serveur, et par les applications pour lesquelles **la livraison des données à temps est plus importante que leur précision**, comme pour la transmission de son de sons ou d'images.

Il n'y a **pas de connexion**, mais il est tout de même possible de mettre en place un processus d'acquittement. **UDP prend en charge la diffusion**. **UDP ne fournit pas de mécanismes de vérification d'erreurs étendues**.

Ce mode est utilisé pour les requêtes **DNS** ou **DHCP**. Il est moins gourmand en ressources.

Caractéristiques de UDP :

- UDP possède un mécanisme permettant d'identifier les processus d'application à l'aide de numéros de port UDP.
- UDP est orienté datagrammes (sans connexion), ce qui évite les problèmes liés à l'ouverture, au maintien et à la fermeture des connexions.
- UDP est efficace pour les applications en **diffusion/multidiffusion**. Les applications satisfaisant à un modèle du type « interrogation-réponse » peuvent également utiliser UDP. La réponse peut être utilisée comme étant un accusé de réception positif à l'interrogation. Si une réponse n'est pas reçue dans un certain intervalle de temps, l'application envoie simplement une autre interrogation.
- UDP ne séquence pas les données. **La remise conforme des données n'est pas garantie**.

- UDP peut éventuellement vérifier l'intégrité des données (et des données seulement) avec un total de contrôle.
- UDP est **plus rapide, plus simple et plus efficace** que TCP mais il est moins robuste.

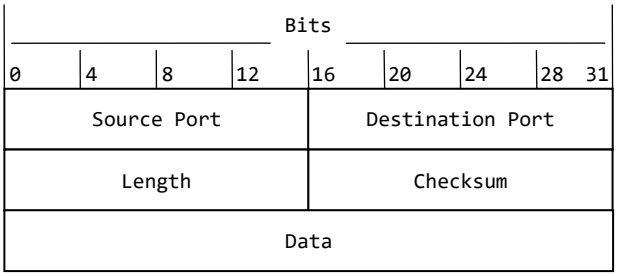
Le protocole UDP permet une transmission sans connexion, mais aussi sans sécurité. Pourtant de nombreuses applications reposent sur UDP :

- TFTP
- DNS
- DHCP
- NFS
- SNMP
- -RIP
- Jeux en réseaux
- Streaming
- Televison et radio par Internet

L'en-tête a une taille fixe de 8 octets.
Le champ **Source Port** occupe 16 bits. Il indique :

- le numéro de port du processus émetteur,
- le numéro de port où on peut adresser les réponses lorsque l'on ne dispose d'aucun autre renseignement.
- si sa valeur est 0, cela signifie qu'aucun numéro de port n'est attribué.

Le champ **Destination Port** identifie le processus correspondant à l'adresse IP de destination auquel on envoie les données UDP. UDP effectue le **démultiplexage** des données à l'aide de **numéros de port**. Lorsque UDP reçoit un datagramme sans numéro de port, il génère un message d'erreur ICMP indiquant qu'il est impossible de contacter le port et il rejette le datagramme.



Le champ **Length** contient la longueur du paquet UDP en octets (en-tête + données). La valeur minimale est 8 et correspond à un paquet où le champ de données est vide.

Le pseudo en-tête de préfixe de l'en-tête UDP contient l'adresse d'origine, l'adresse de destination, le protocole (UDP = 17) et la longueur UDP. Ces informations sont destinées à prévenir les erreurs de routage.

10. QUELLES SONT LES DIFFÉRENTES COUCHES D'UN RÉSEAU

Cela dépend du modèle utilisé :

Selon le **modèle OSI** :

1 **couche physique** > 2 **couche liaison de données** > 3 **couche réseau** > 4 **couche transport** > 5 **couche session** > 6 **couche présentation** > 7 **couche application**

Selon le **modèle TCP/IP** (*le plus couramment utilisé et nommé d'après les principaux protocoles utilisés en son sein*) :

1 couche **liaison** (*ou accès réseau selon les différentes sources consultées*) > 2 **Internet** > 3 **Transport** > 4 **Application**

La couche **physique** est plus ou moins **regroupée** dans la couche **accès réseau** et les couches **sessions** et **présentation** sont **supprimées**, leurs fonctions étant **reportées** sur la couche **application**.

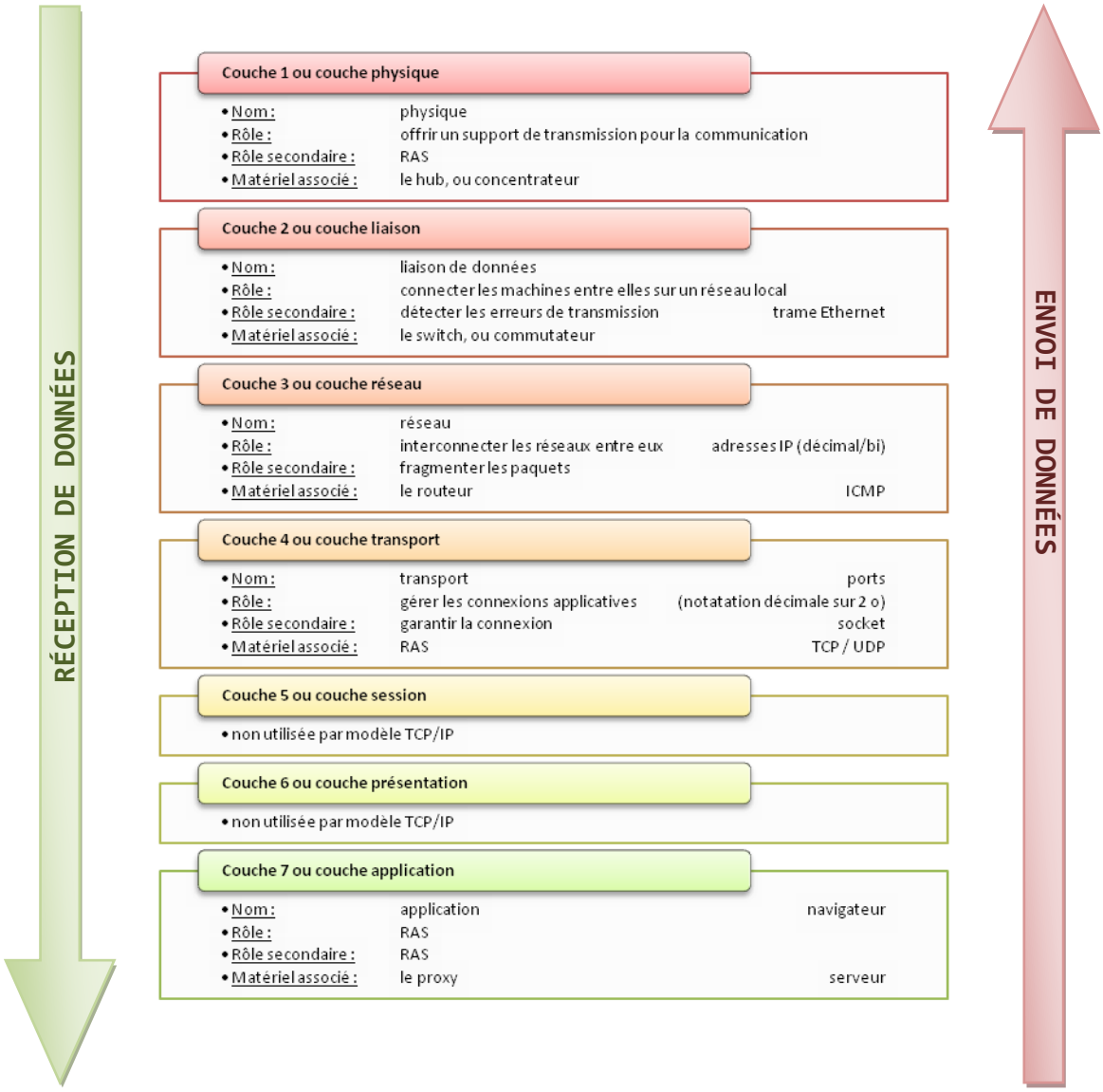
La couche **liaison** n'est pas vraiment une couche a proprement parler, mais plutôt d'une **interface** entre les hôtes et les liens de transmission

11. QU'EST-CE QUE LE MODÈLE OSI

Il s'agit d'une norme créée en 1984 que doivent respecter les gens communiquant sur Internet et s'y connectant. Son objectif est de **normaliser les communications pour garantir un maximum d'évolutivité et d'interopérabilité entre les ordinateurs**.

C'est un modèle en couche, **chaque couche ayant un rôle défini**. À l'heure actuelle les couches 5 et 6 ne sont pas utilisées.

Le modèle OSI est un **modèle théorique**, le modèle sur lequel s'appuie Internet est le modèle **TCP/IP**, on n'utilise pas les couches 5 et 6 mais on passe par la couche 7 pour gérer la session (cookies) et la présentation (ascii, Unicode…) d'un utilisateur. Le modèle OSI ne constitue pas une architecture réseau car il ne spécifie pas les services et protocoles précis à utiliser dans chaque couche. **Il précise simplement ce que chaque couche doit faire**.



12. À QUOI SERT UN SERVEUR DHCP ET QU'EST-CE QUE LE PROTOCOLE DHCP

Protocole de couche 7 application. (*Dynamic Host Configuration Protocol*). **RFC 2131** et **RFC 2132**.

Son rôle est d'assurer la **configuration automatique des paramètres IP d'une machine**, notamment en lui attribuant automatiquement une adresse **IP** et un **masque de sous-réseau**. DHCP peut aussi configurer l'adresse de la **passerelle par défaut**, des serveurs de noms **DNS**.

Ce protocole peut fonctionner avec **IPv4**. Il fonctionne aussi avec **IPv6**, il est alors appelé DHCPv6. Toutefois, en IPv6, les adresses peuvent être auto configurées dans DHCP.

DHCP utilise **UDP** et **IP**.

Le client, dépourvu d'IP, envoie en diffusion broadcast un datagramme (**DHCPDISCOVER**) qui s'adresse au **port 67** de n'importe quel serveur à l'écoute sur ce port. Ce datagramme comporte entre autres l'adresse physique (MAC) du client.

Tout serveur DHCP ayant reçu ce datagramme, envoie une offre DHCP (**DHCPOFFER**), s'il lui reste des disponibilités à l'attention du client sur son **port 68**, identifié par son adresse physique. Cette offre comporte l'adresse IP du serveur ainsi que l'adresse IP et le masque de sous-réseau proposés au client. Le client peut recevoir plusieurs propositions suite à sa requête.

Le client retient une offre en retournant sur le réseau un datagramme DHCP (**DHCPREQUEST**). Ce datagramme comporte l'adresse IP retenue par le client (la première qui lui est parvenue en général). Cela a pour effet de demander au serveur choisi l'assignation de cette adresse, l'envoi éventuel des valeurs de paramètres et d'informer les autres serveurs qu'ils ont fait une offre qui n'a pas été retenue.

Le serveur DHCP émet alors un datagramme d'acknowledgment (**DHCPACK**) pour confirmer la demande d'assignation précédemment formulée. Dans ce datagramme, le serveur spécifie la durée du bail de cette adresse (dont découle deux valeurs T1 et T2 déterminant le comportement du client en fin de bail). Peut être également fourni a ce moment-la 'IP de la passerelle par défaut, les IP des serveurs DNS ou encore celles des serveurs NBN (WINS).

Un serveur DHCP a une adresse IP statique.

13. À QUOI SERT UN SERVEUR DNS ET QU'EST-CE QUE LE PROTOCOLE DNS

Protocole de couche 7 application.

Domain Name System (*système de nom de domaine*)

Un serveur DNS **traduit un nom d'hôte en adresse IP**. Par exemple, lorsque dans le barre de recherche de notre navigateur préféré nous entrons www.google.com, celui-ci fait appel à un serveur DNS pour traduire le nom de domaine www.google.com en adresse IP 212.58.205.196.

On peut obtenir cette information grâce à la commande *nslookup* suivie du nom de domaine. À l'inverse, on obtient un nom de domaine avec la commande *whois* suivie de l'adresse IP en question.

14. QUELLES SONT LES CONFIGURATIONS MINIMALES POUR FAIRE COMMUNIQUER 2 APPAREILS EN UTILISANT IP

Pour utiliser IP afin de communiquer, il nous faut pour le client une **carte réseau**, une adresses **IP**, une **passerelle par défaut** et un **routeur** qui a une table de **NAT** (*correspondance entre l'IP publique du routeur et les IP privées qui ont accès au routeur par la passerelle se situant sur le réseau domestique privé*).

Ce routeur doit avoir une **interface** (*carte réseau*) dans chacun des réseaux auxquels il est connecté. Soit il a lui-même accès, par l'une de ses interfaces a l'adresse IP de destination, soit il n'y a pas accès et alors il adresse le paquet au routeur suivant dans sa table de routage.

La deuxième machine ayant enfin récupéré le paquet par un routeur, il lui faut le même matériel que pour la première pour recevoir la communication en toute conformité.

15. COMMENT MARCHE LE ROUTAGE IP

Le **routage nous permet d'envoyer un message en dehors de notre réseau**. Le routeur est le matériel de couche 3 réseau permettant de connecter les réseaux entre eux

Un **routeur possède minimum deux interface réseau** (cartes réseaux) pour au moins être connecté à deux réseau différents en même tps. En général un routeur, dont c'est vraiment le travail, a un peu plus d'interfaces que ce minimum. On pourrait utiliser un ordinateur avec deux cartes réseaux comme un routeur, en activant le routage dessus.

Ce qui différencie un routeur d'un ordinateur, c'est qu'un **routeur accepte de relier des paquets qui ne lui sont pas destinés** tandis qu'un **ordinateur les jettera** a la poubelle.

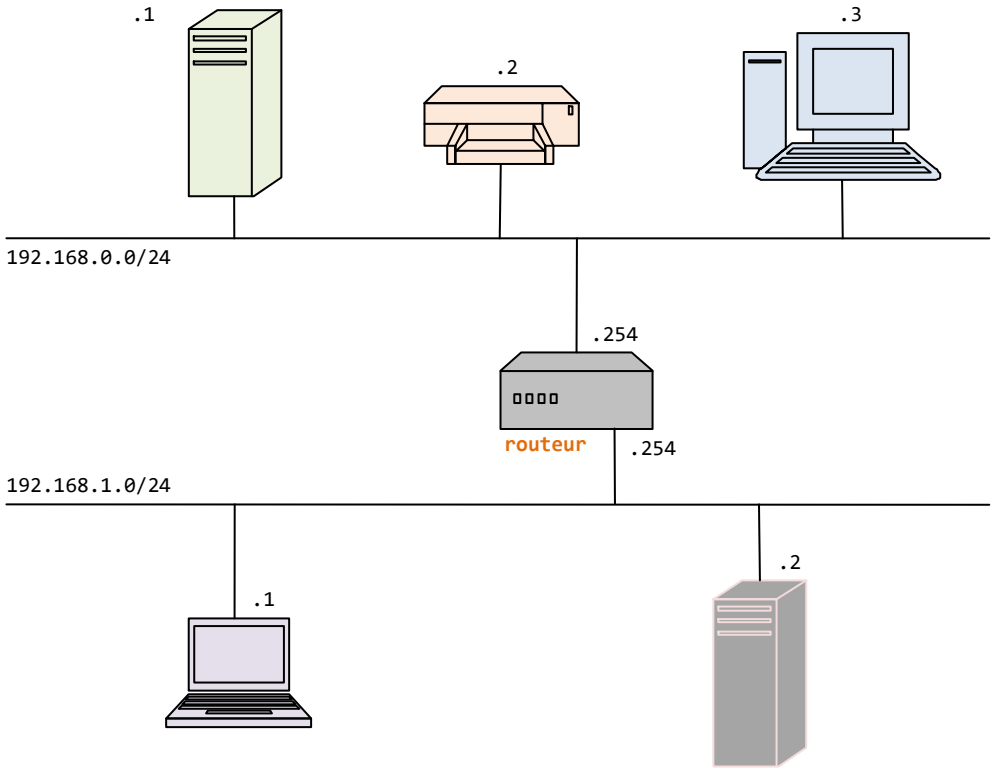
Un routeur aiguille ses paquets grâce à sa table de routage.

Une table de routage **liste tous les routeurs auxquels il conviendra de faire suivre le datagramme pour joindre une destination donnée**. La destination donnée ne sera pas une machine mais un **réseau**.

Le principe est d'avoir d'un coté la liste des réseaux que l'on veut joindre et de l'autre, la liste des routeurs à qui envoyer le datagramme pour joindre ces réseau. **Ces routeurs sont des passerelles vers les autres réseaux**. La table de routage indique quelle passerelle utiliser pour joindre un réseau. (pour obtenir sa table de routage : *route print* sous Windows ou *netstat –r*).

Méthode pour écrire sa table de routage :

- On indique les réseaux auxquels on est connecté.
- On indique la route par défaut.
- Puis on indique tous les autres réseaux qu'on ne peut pas joindre avec les deux étapes précédents.



16. QU'EST-CE QU'ON APPELLE UNE PASSERELLE PAR DÉFAUT ET À QUOI ÇA SERT

La **passerelle par défaut** (*default gateway*) est l'adresse par laquelle la machine émettrice d'un paquet pourra le transmettre au **routeur** qui se chargera soit de le router vers le prochain routeur soit de le transmettre a la machine destinataire si celle-ci fait partie du réseau d'une des interfaces du routeur. C'est l'élément qui nous permettra de se connecter à Internet, et il est à la première ligne de chaque table de routage, normalement. Sans cela, notre machine ne pourrait communiquer qu'avec les hôtes du même réseau. **La passerelle par défaut est toujours un routeur. La passerelle par défaut fait partie du réseau de l'utilisateur.**

Par convention, la route par défaut se note default ou 0.0.0.0/0. On l'utilise pour rediriger un paquet lorsque celui fait mention d'une destination qui n'est pas inscrite dans la table de routage de la passerelle.

17. QU'EST-CE QU'UN PORT D'UN POINT DE VUE IP ET COMMENT CELA EST UTILISÉ POUR SE CONNECTER À UN AUTRE APPAREIL

C'est un numéro associé à un type d'application. On l'appelle également **port réseau**. Il identifie un point où des données ou des informations sont envoyées. Les protocoles **UDP** et **TCP** s'appuient sur ce système pour échanger des informations avec la bonne application, celle qui est concernée par le transfert de données. En effet, une même adresse IP peut être le point d'arrimage de plusieurs services applicatifs, on appelle cela le **multiplexage**. C'est pour les différencier qu'on utilise les ports réseaux.

Les données voyagent à partir d'un port de l'appareil initial et se dirigent vers l'extrémité réceptrice de la ligne. Un numéro de port est un entier codé sur 16 bits, les valeurs possibles vont de 0 à 65535. On utilise un numéro de **port source** et un numéro de **port de destination** pour déterminer les processus utilisés pour envoyer et recevoir les données.

L'IANA a défini plusieurs niveaux dans l'usage des ports (*RFC 1700*) :

- **0 – 1023** : Les ports sont réservés à des services régulièrement employés, par convention, le port 80 est utilisé par le protocole HTTP et le port 443 est utilisé par le protocole HTTPS pour communiquer avec le client. Le port 0 n'est pas utilisé
- **1024 – 49 151** : Ces ports sont enregistrés ou semi-réservés. Les entreprises, les organisations ou même les particuliers peuvent s'inscrire pour utiliser ces numéros de port afin de fournir des services réseau avec l'IANA
- **49 152 – 65 535** : Ces numéros de port font référence aux ports éphémères utilisés par les programmes clients.

La combinaison « **adresse IP : numéro de port** » constitue ce qu'on appelle une « **socket** » (*traduire comme connecteur, prise*). Une socket identifie pleinement le service qui est concerné sur une machine donnée.

Un serveur doit rester attentif aux requêtes clients, sous peine de les rater. On installe des petits programmes (*daemon*) qui tournent en tache de fond et écoutent continuellement sur un numéro de port donné, par exemple un serveur FTP sera en écoute sur son port 21 d'éventuelles requêtes clients.

En revanche le client qui émet la requête ne dispose pas de port d'écoute attitré, car c'est un privilège réservé aux machines de type serveurs. En effet, le client n'a rien d'autre à écouter que les réponses à ses requêtes. En envoyant sa requête, **il spécifiera sur quel port il écoutera la réponse**, de manière a ce que le serveur construise un socket efficace pour ladite réponse.

Pour ne pas mélanger les réponses, si plusieurs requêtes sont adressées, le client indiquera autant de ports différents qu'il y a de port occupés dans la table NAT du routeur. C'est le NOS du client qui choisit les ports à distribuer.

Le **port forwarding** consiste à rediriger un port de notre routeur vers un port donné sur une machine locale.

ⓘ Chaque port ouvert est une porte ouverte par laquelle peut entre un intrus.

18. BIBLIOGRAPHIE

https://irp.nain-t.net/
https://openclassrooms.com/fr/courses/857447-apprenez-le-fonctionnement-des-reseaux-tcp-ip
https://www.speedcheck.org/fr/wiki/port/
http://www.gipsa-lab.grenoble-inp.fr/~christian.bulfone/MIASHS-L3/PDF/3-Les protocoles UDP TCP.pdf
https://perso.liris.cnrs.fr/alain.mille/enseignements/emiage/emiage%20-%20ModuleC214/TJ/214_4_5.htm#:~:text=DHCP%20signifie%20Dynamic%20Host%20Configuration,qui%20utilise%20UDP%20et%20IP.
https://fr.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol
Réseaux – 5e edition, Andrew Tanenbaum – David Wetherall