Université Polytechnique Hauts-De-France (UPHF)
Institut des Sciences et Techniques de Valenciennes (I.S.T.V.)

Synthèse du cours réseaux TCP/IP

Licences Info L3, LP SIO

Henri TSOUNGUI
Ing. CNAM, Enseignant titulaire
UPHF I.S.T.V., dec. 2018
henri.tsoungui@uphf.fr
http://tsoungui.fr

Musique 'La Dolce Fiamma' de Jean Chrétien BACH
Chantée par le contre-ténor Philippe Jaroussky

L’auteur

Henri TSOUNGUI

Ingénieur CNAM en Systèmes d’Information

Option Conception et Gestion des Systèmes d'Information

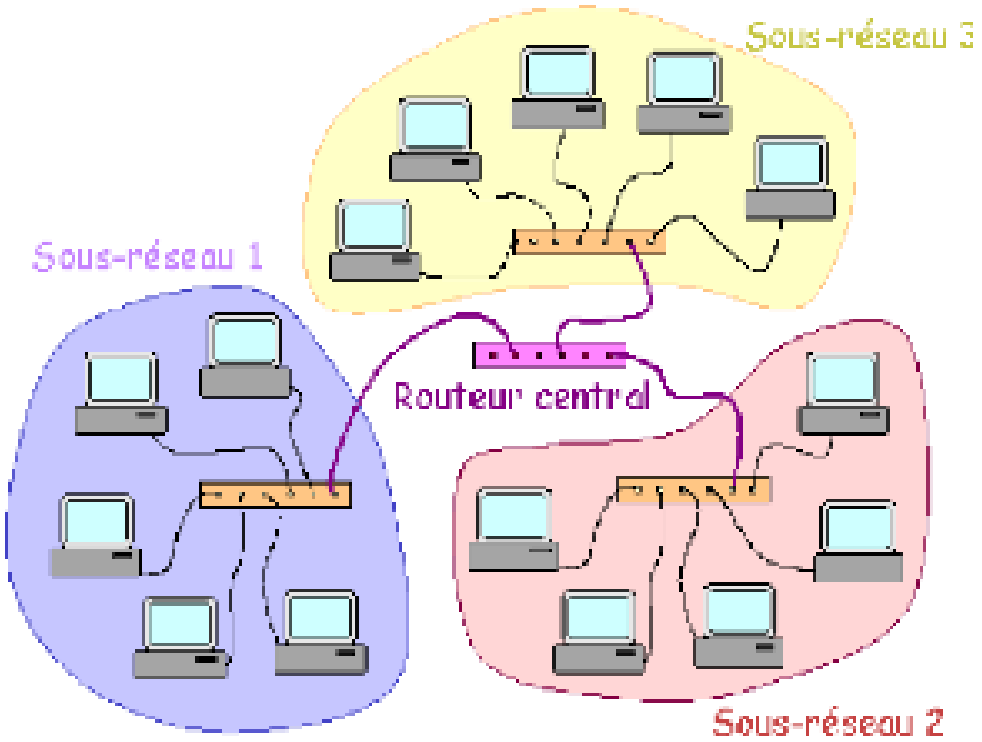
Professeur Certifié Major au CAPET D (Economie Gestion et Info)

Institut des Sciences et Techniques de Valenciennes (ISTV) de l'Université Polytechnique des Haut-De-France (UPHF)

Site perso : http://www.tsoungui.fr

henri.tsoungui@uphf.fr

Structure et buts d'un réseau

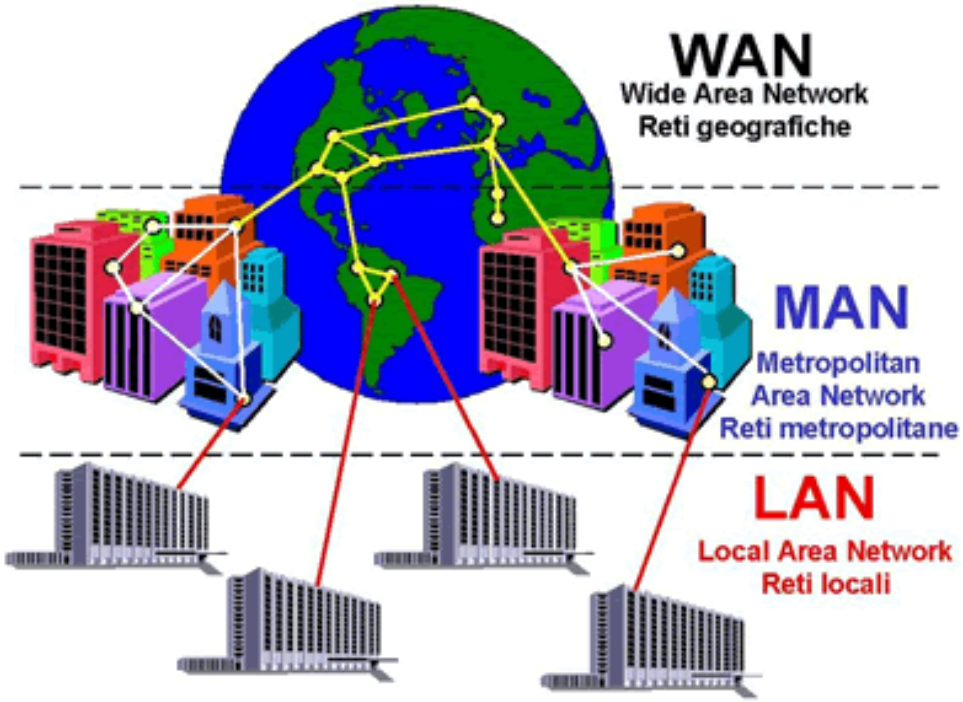


Equipements matériels

-postes de travail (ordinateurs), câbles

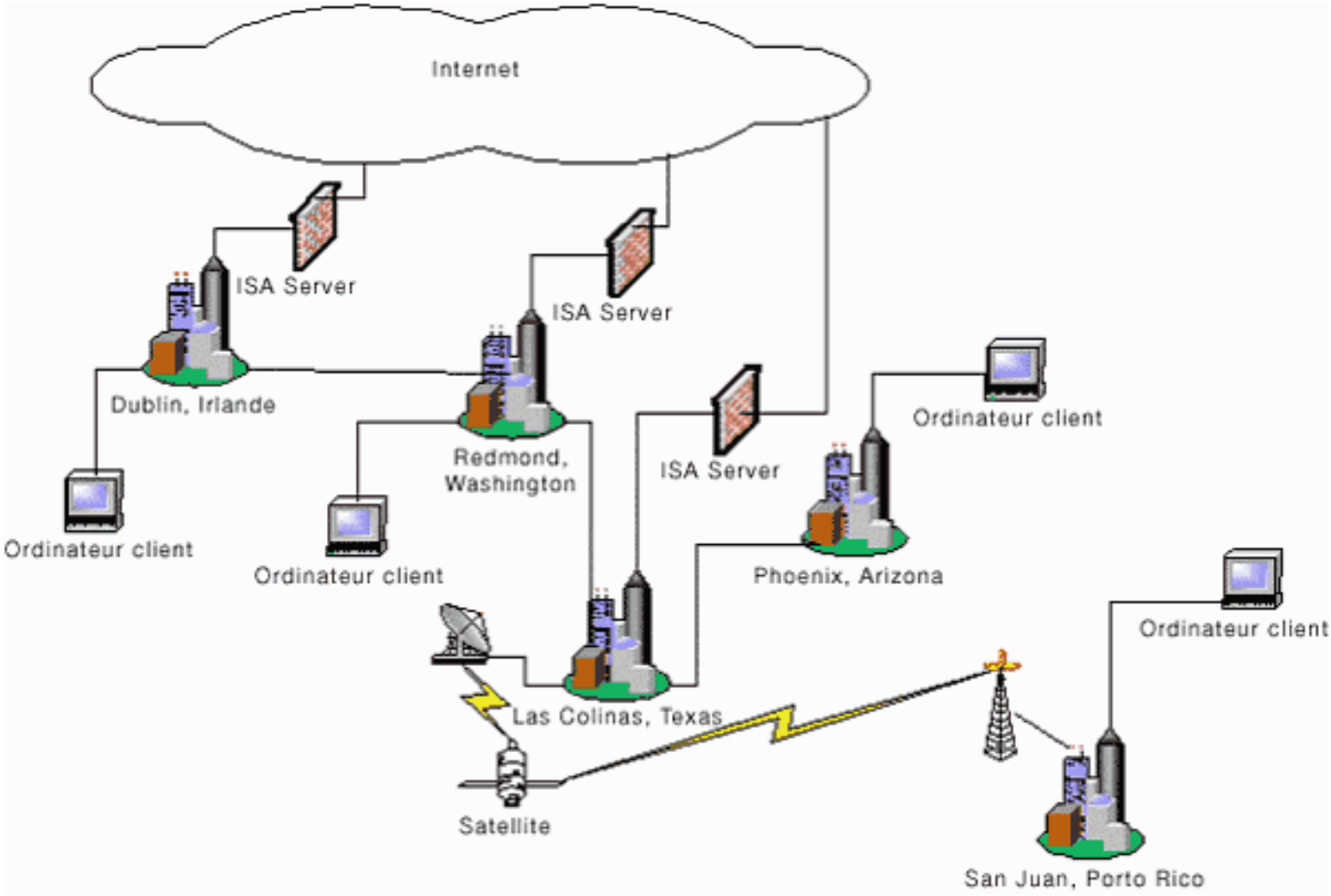
-cartes réseau, concentrateurs(hubs), commutateurs(switches), routeurs(routers)

Etendue géographique d'un réseau

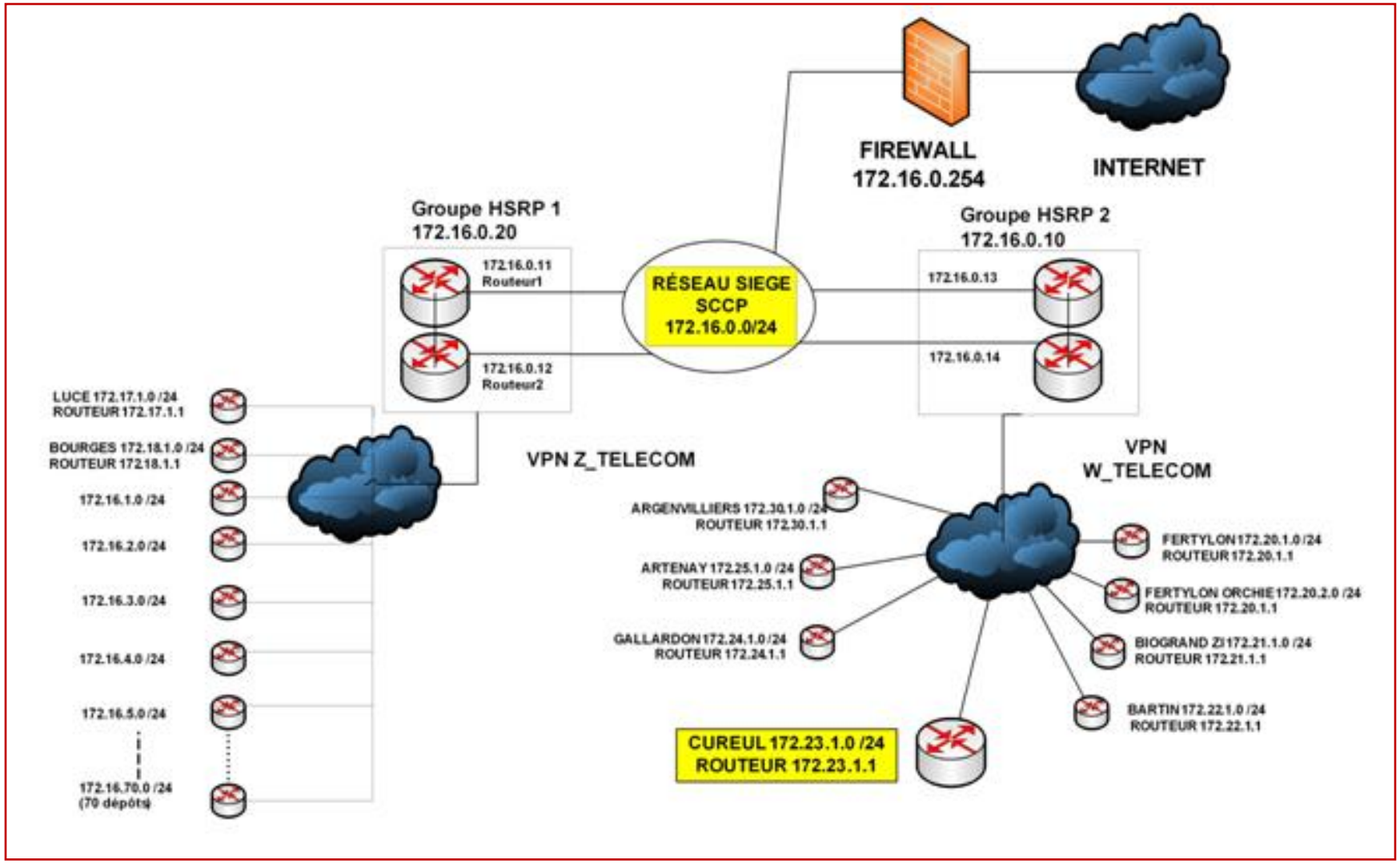


LAN - Local Area Network : réseau local de faible dimension (moins de 2km)
CAN - Campus Area Network : réseau de campus ou petite cité
MAN - Metropolitan Area Network : réseau métropolitain
WAN - Wide Area Network : réseau étendu

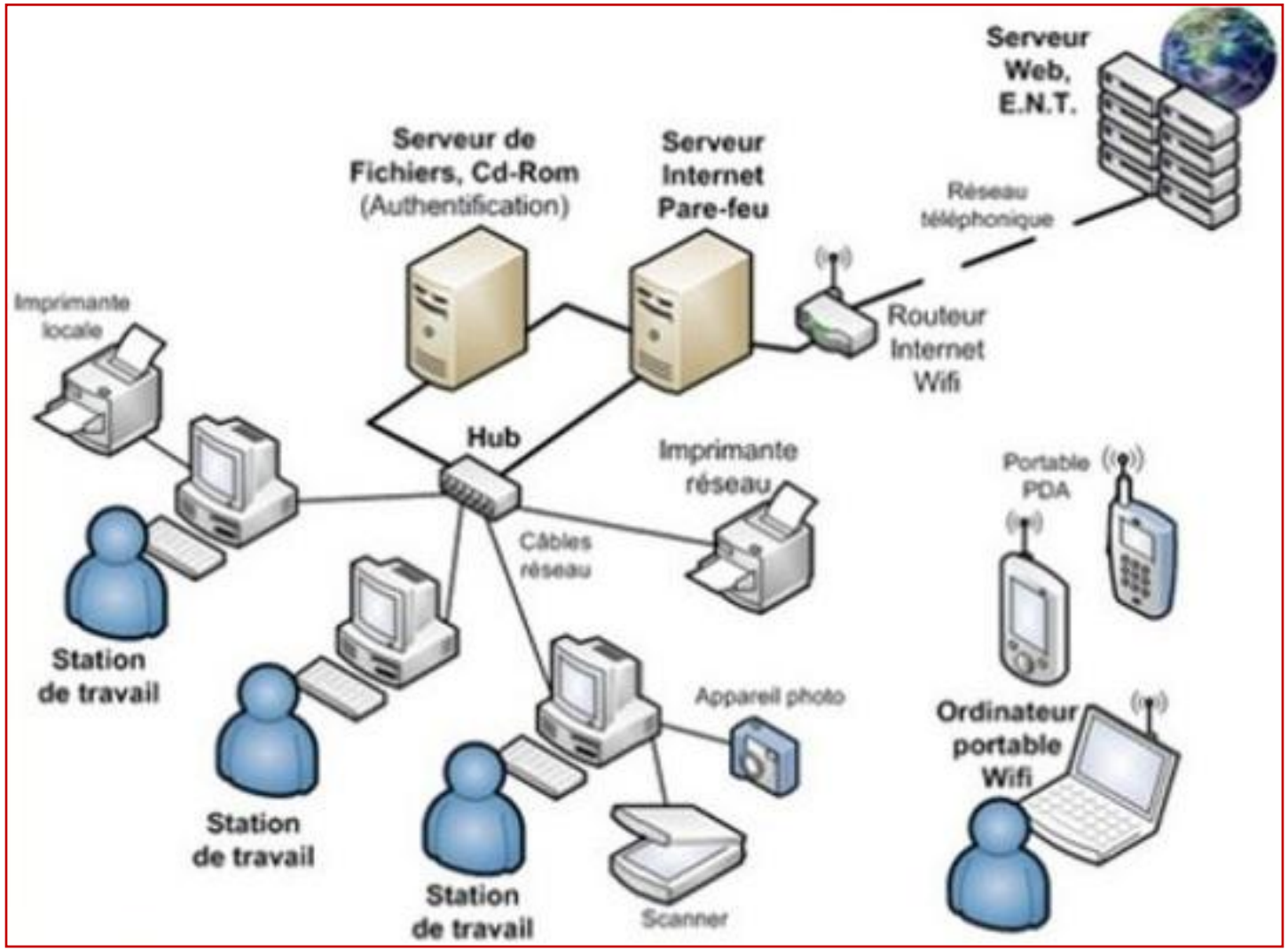
Exemples de réseaux



Exemples de réseaux



Ex de réseau d'établissement scolaire

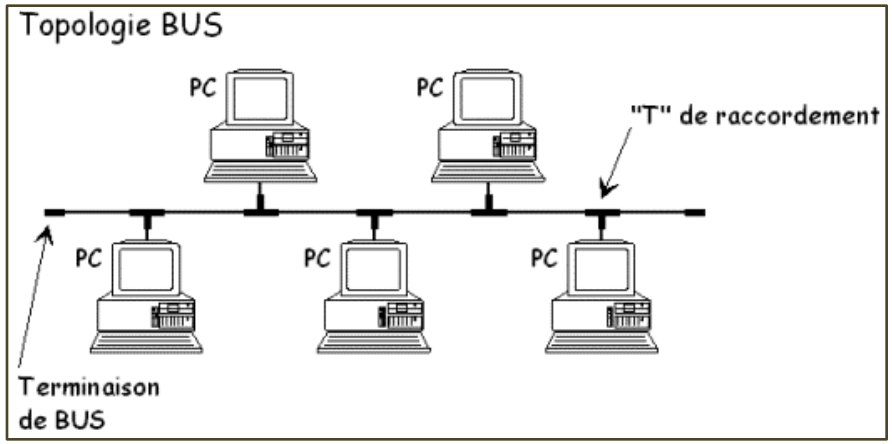


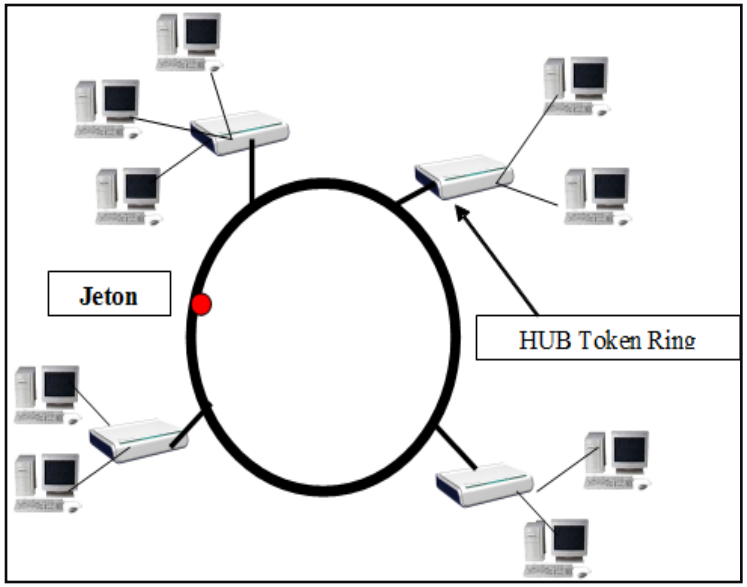
Buts des réseaux : partage des ressources

- **Partager : mettre à la disposition de plusieurs membres**
- **Ressources :**
  - **Stockage (espace disques)**
  - **Calcul (processeurs)**
  - **Communication (connexion Internet par exemple)**

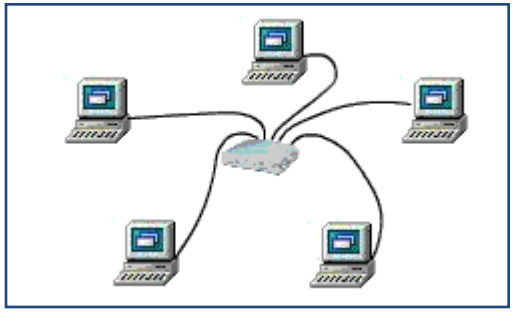
Le partage suppose une bonne gestion des DROITS d'accès aux ressources disponibles

Topologies classiques et modernes des réseaux





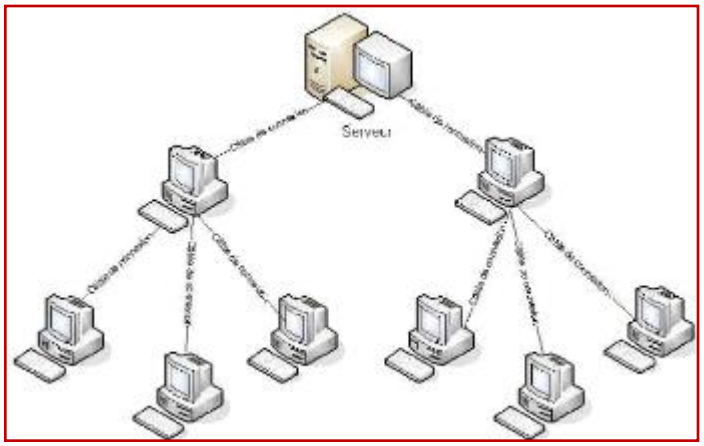
Topologie étoile sur MAU



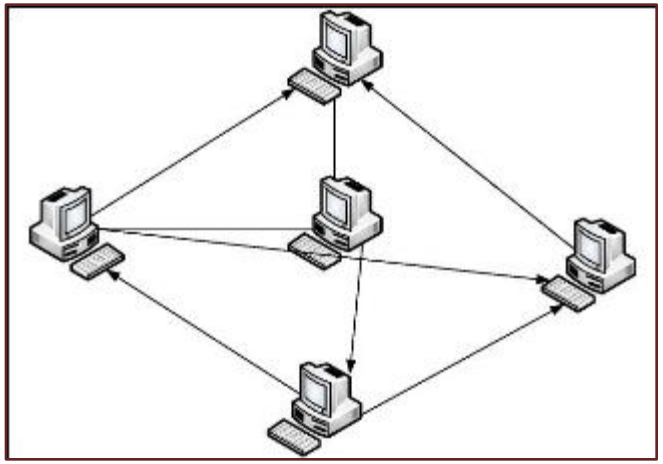
Topologie Anneau

- **Le Bus : ancienne topologie utilisant les câbles coaxiaux avec un débit très limité (10 Mbps)**
- **L'anneau : réseau en boucle d'IBM dans lequel circulait un jeton donnant la main à chaque poste pour accéder au réseau et émettre (16 Mbps)**
- **L'étoile : topologie plus moderne et performante selon les composants matériels (plusieurs GBps)**

Autres topologies



Topologie ARBRE



Topologie MAILLEE

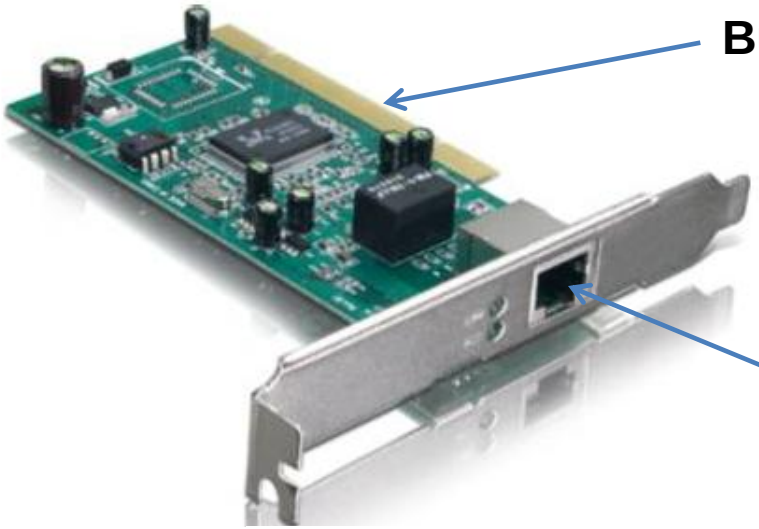
Topologie étoile dans le détail

Carte d'interface réseau

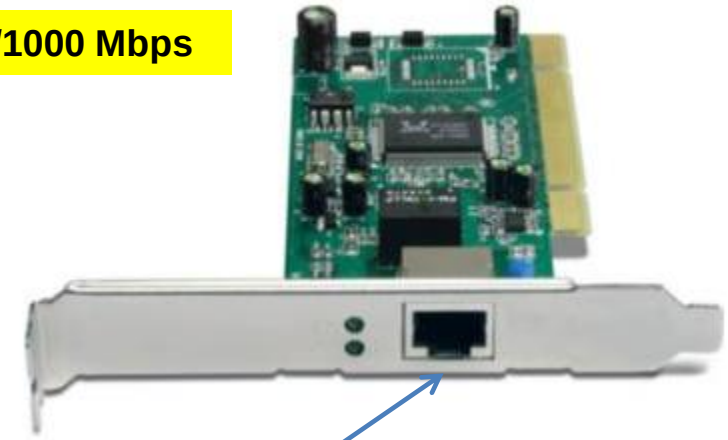
MAU : Multi Access Unit : Hub ou Switch

Câble Paires torsadées STP (Shielded Twisted Pair) ou UTP (Unshielded T wisted Pair)

Exemple de carte PCI Gigabit 10/100/1000 Mbps



Bus PCI

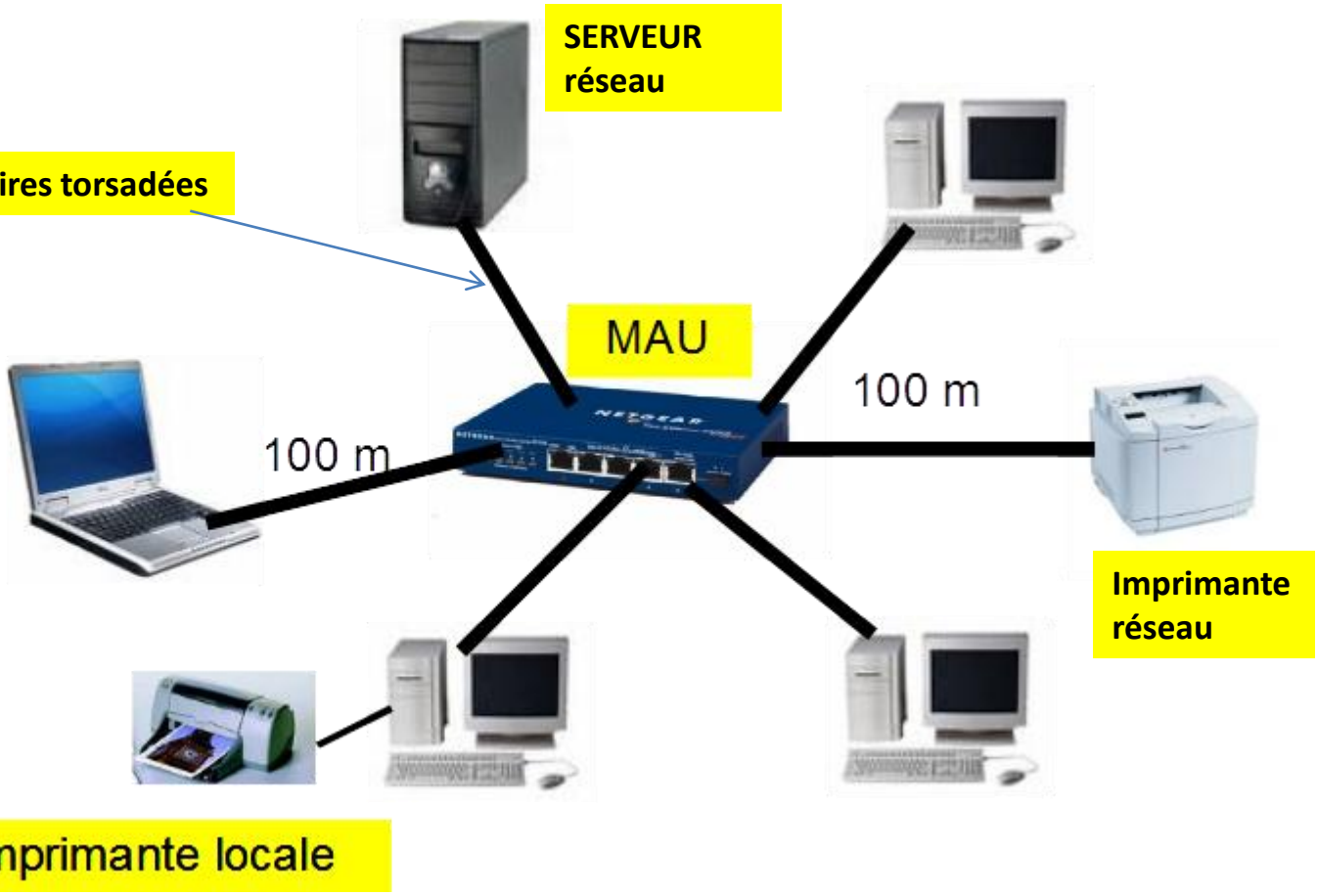


Connecteur RJ-45

Carte réseau 2 ports RJ-45

Carte réseau 4 ports RJ-45

Topologie étoile

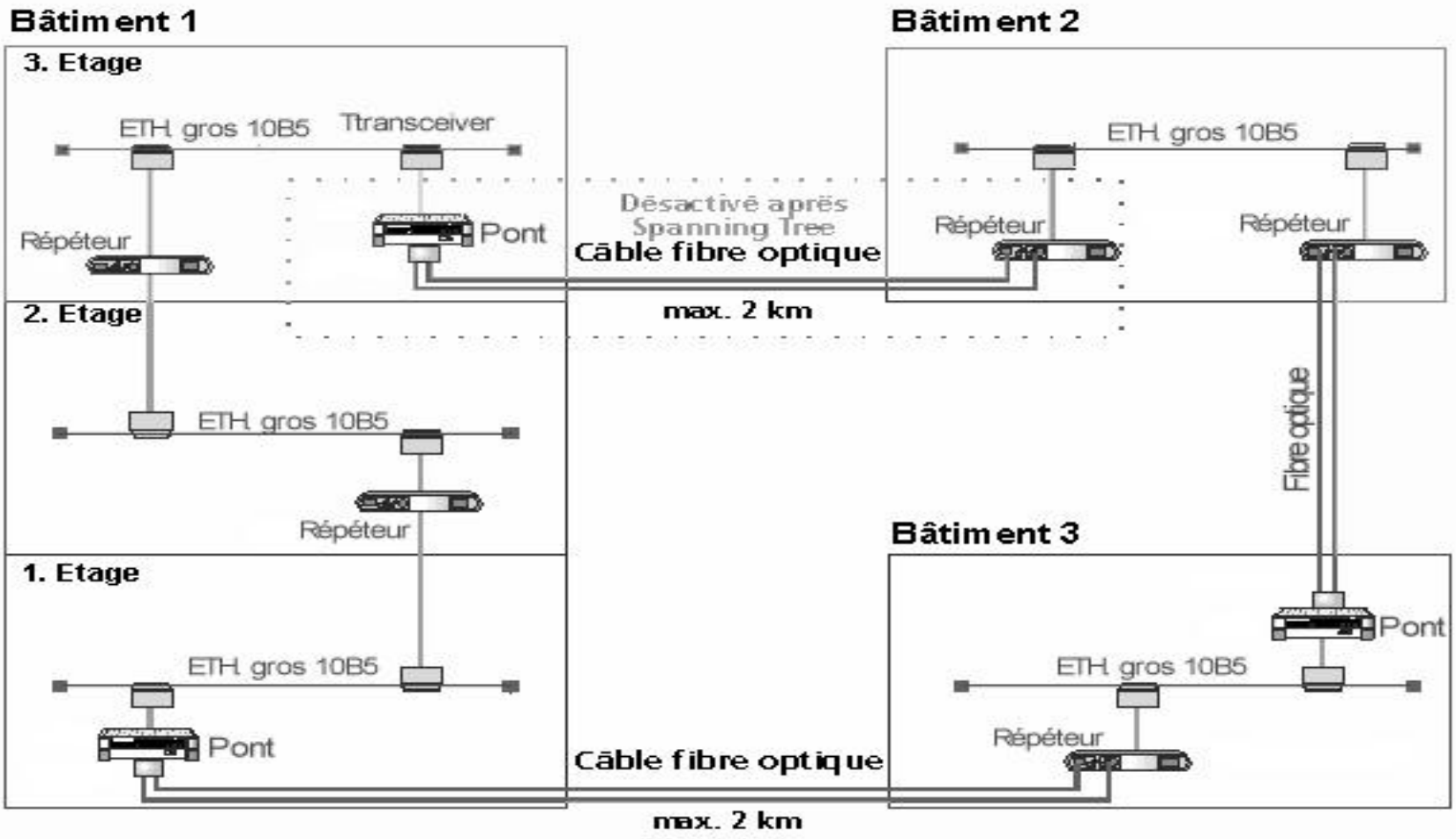


Types de câblage et débits

Tableau comparatif des vitesses. Le 1000 base T est le plus courant

Type	Vitesse	Distance	Media
10BASE-T	10 Mb / s	100m	Cuivre
100BASE-TX	100 Mb/s	100m	Cuivre
100BASE-FX	100 Mb/s	412 m - 2 Km	half Duplex Multi mode Fibre optique Full Duplex multi mode Fibre optique
1000 Base LX	1000 Mb/s 1000 Mb / s	3 Km 550m	Single mode Fibre optique (SMF) Multi-mode Fibre optique (MMF)
1000 Base SX	1000 Mb/s 1000 Mb/s	550m 275m	Multi-mode Fibre optique (50u) Multi-mode Fibre optique (62.5 u)
1000 Base C (pas supportée par les applications industrielles standards)	1000 Mb / s	25m	Cuivre, 4 paires UTP5
1000 Base T - 1000 Base TX IEEE 802.3 ab ratifié le 26 juin 1999,	1000 Mb / s	100m	Cuivre, câble catégorie 5e, transmission sur 4 paires (250 Mbits/paire)
1000 BASE LH	1000 Mb/s	70 km	Fibre optique

Extensions d'un réseau local



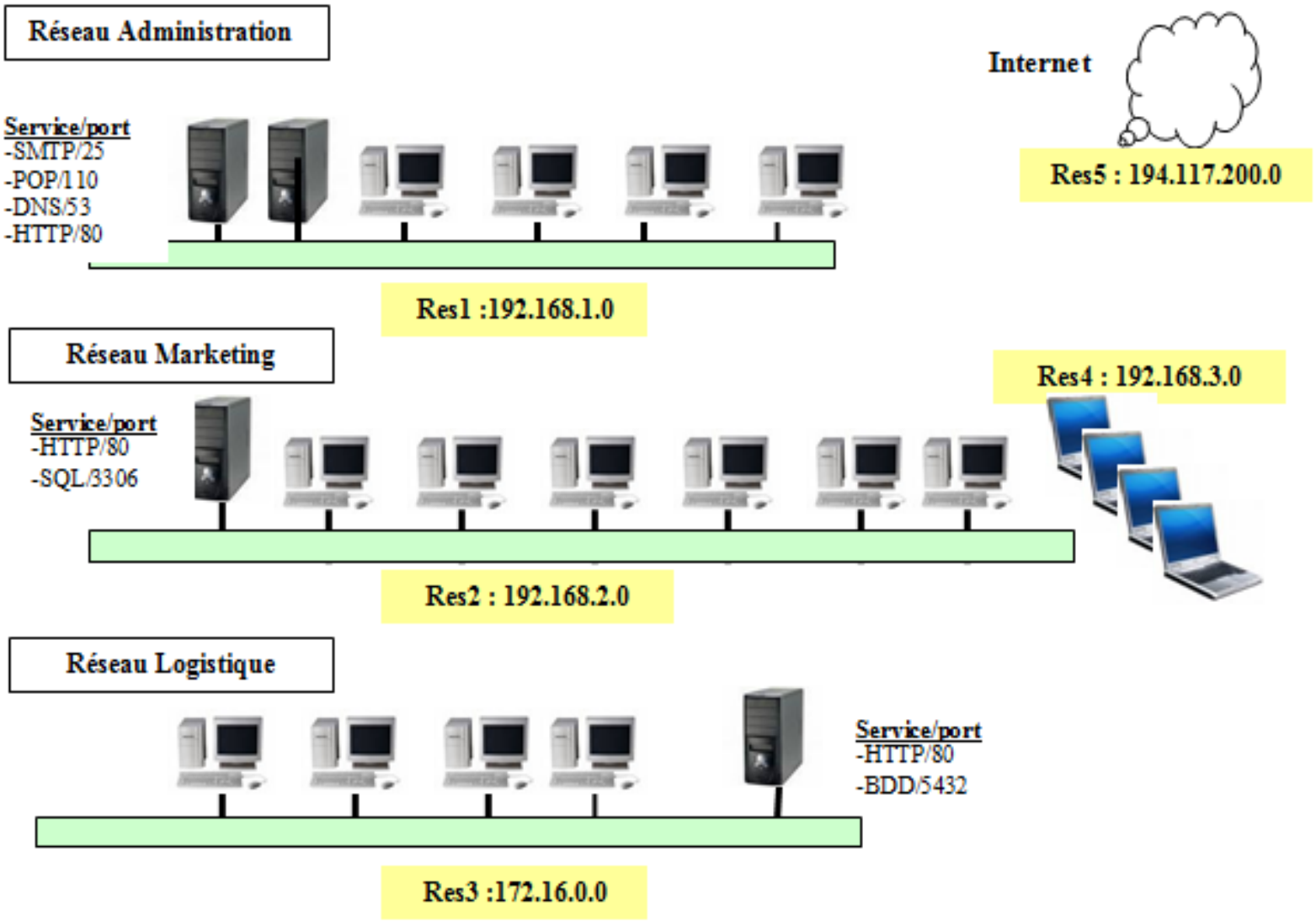
Règle des 5-4-3

Un réseau ETHERNET FIN ne doit comporter

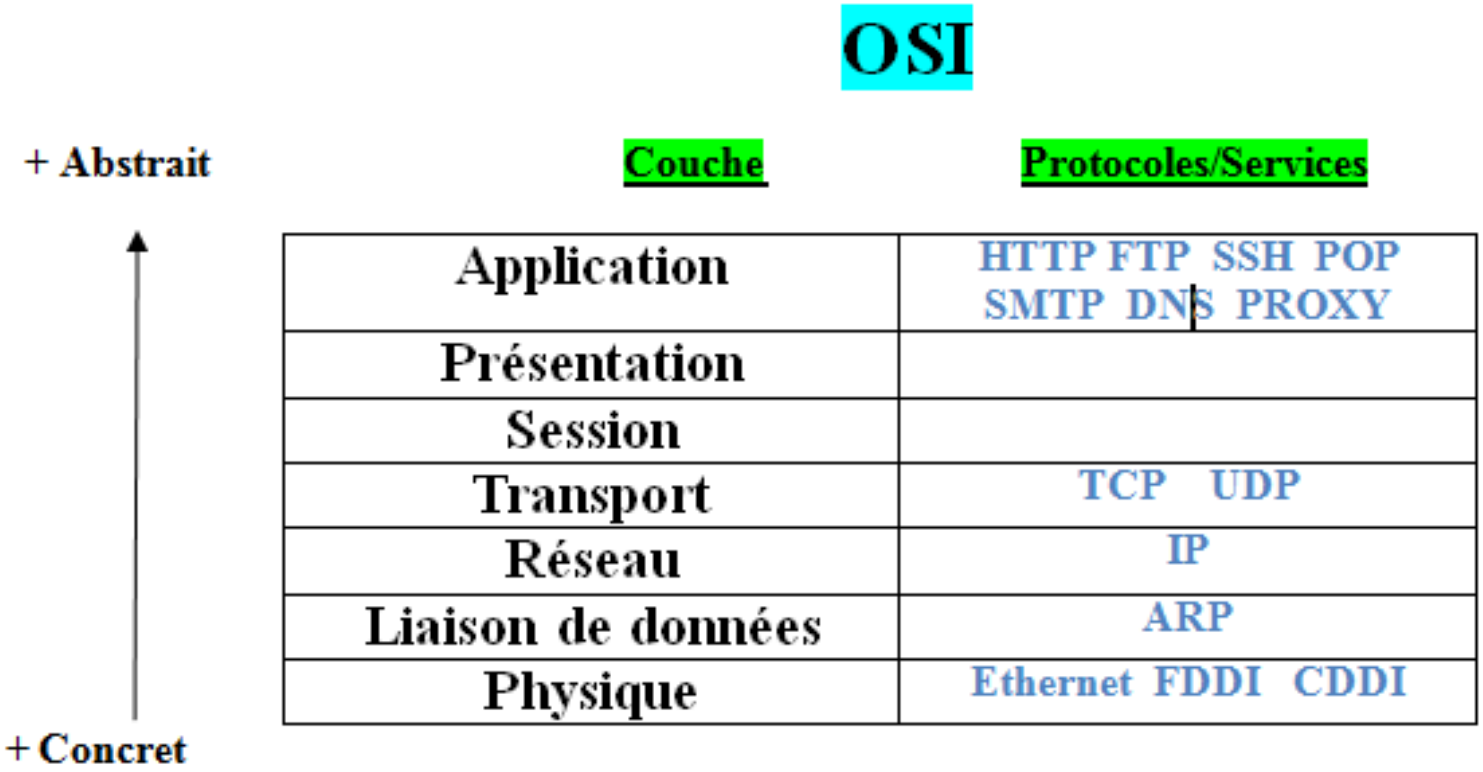
- **5 segments** de câbles au plus reliés par
- **4 répéteurs**, mais
- **3 segments** seulement peuvent héberger des stations,

c'est la règle des 5-4-3.

- Deux segments doivent donc rester inexploités, ils servent de liaisons inter-répéteurs et permettent d'augmenter la longueur totale du réseau. L'IEEE 802.3 recommande un maximum de 30 nœuds (ordinateurs, répéteurs,…) par segment, et un maximum de 1024 ordinateurs pour la totalité d'un réseau.


Réseau Administration
Internet
Service/port
-SMTP/25
-POP/110
-DNS/53
-HTTP/80
Res5 : 194.117.200.0
Res1 :192.168.1.0
Réseau Marketing
Res4 : 192.168.3.0
Service/port
-HTTP/80
-SQL/3306
Res2 : 192.168.2.0
Réseau Logistique
Service/port
-HTTP/80
-BDD/5432
Res3 :172.16.0.0

Adressage IP

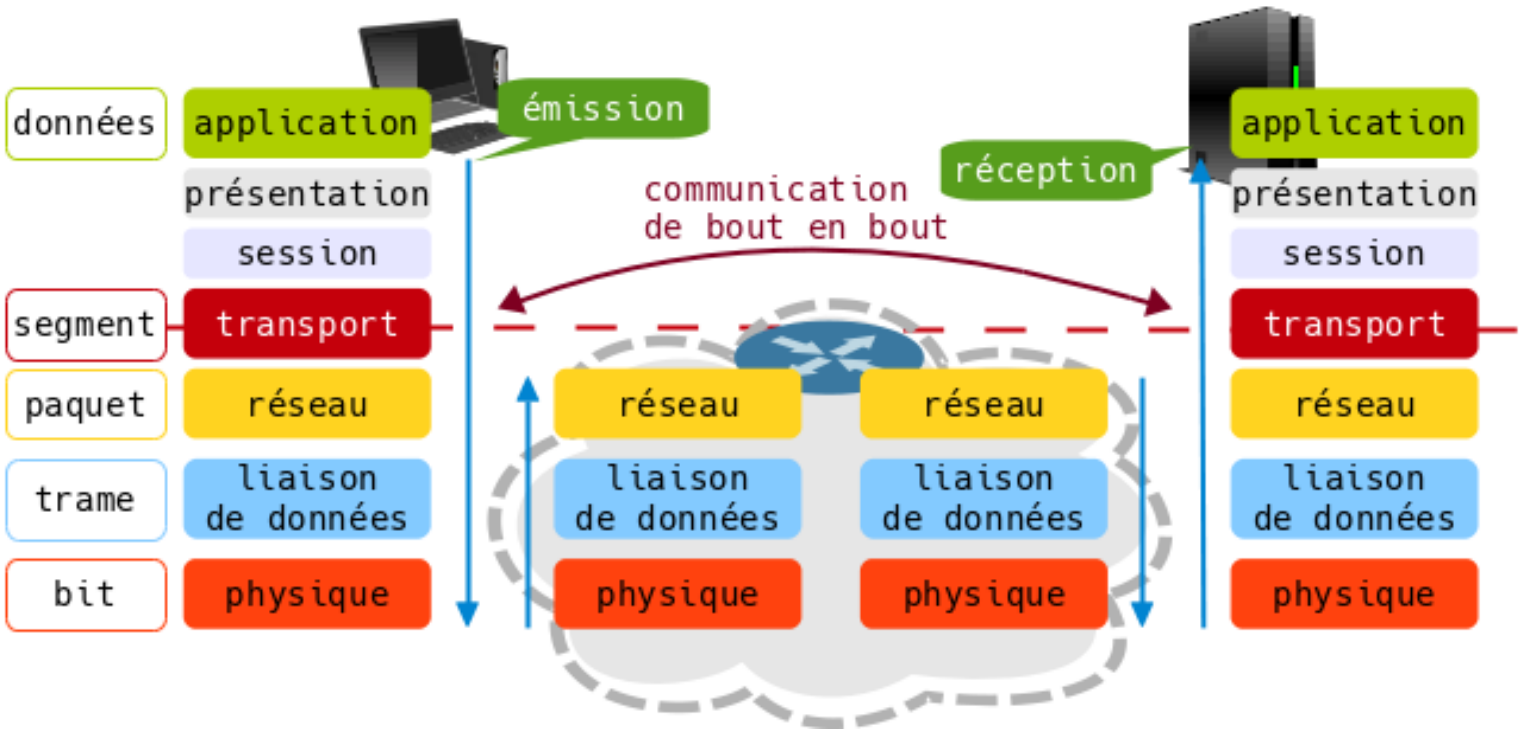


Modèle TCP/IP du DoD

(DoD : Department of Defense)

TCP/IP	
Application Présentation	Application
Session Transport	Transport
Réseau	Internet
Liaison de données Physique	Interface Réseau

Modèle OSI de l'ISO



Identification des composants d'un réseau

Elle peut être réalisée par :

-un nom (PRN, PC1, PC2, etc)

-un codage quelconque

-une **adresse IP** (Internet Protocol)

-Format d'une adresse X. Y . Z . T

avec 0 <= (X, Y, Z, T) <= 255 en notation décimale

Exemples : 129.30.25.246

15.60.34.20

-Une adresse IP comporte 2 parties :

-L'id réseau (net-id) identificateur réseau

-L'id-hôte (host-id) identificateur d'hôte/composant

Masques de sous-réseau et masques par défaut

- Le *masque de sous-réseau* permet de distinguer les ***deux parties de l'adresse IP***.
- Il a le même format que l'adresse IP et recouvre un certain nombre de bits
- Partie des bits de poids fort => partie réseau
- Partie des bits de poids faible => partie hôte
- Les masques par défaut dépendent de la CLASSE d'adresse (1er octet de l'IP)

Quelques règles à mémoriser
(écriture octale)

- Le masque est obtenu en mettant à « 1 » tous les bits de l'id-réseau, les autres à « 0 »
- L'IP réseau est obtenue en conservant l'id-réseau et en mettant à « 0 » tous les bits de l'id-hôte
- L'IP de diffusion est obtenue en conservant l'id-réseau et en mettant à « 1 » tous les bits de l'Id-hôte

Classes de réseaux et masques

On distingue les réseaux en **classes**. Chaque classe a un **masque par défaut** et des caractéristiques différentes
-en nombre de réseaux et
-nombre de composants ou hôtes

- En fonction du résultat de la conversion du

1er octet en binaire $X_{(2)}$ = **xxxx xxxx** . Y . Z . T

  - Si **0xxx xxxx** => classe **A** masque **255.0.0.0**
  - Si **10xx xxxx** => classe **B** masque **255.255.0.0**
  - Si **110x xxxx** => classe **C** masque **255.255.255.0**
  - Si **1110 xxxx** => classe **D** masque **255.255.255.255**

  Les classes d'utilisation courante sont les classes A, B et C

Nombre de réseaux et nombre de composants

- Une adresse IP : 4 octets

  xxxx xxxx . xxxx xxxx . xxxx xxxx . xxxx xxxx

  Nombre de valeurs décimales par octet : $2^8 = 256$

  D'où pour les 4 octets $2^8 * 2^8 * 2^8 * 2^8$

  On déduit pour les 3 classes :

  Classe A : 1 octet 2^8 **réseaux et** $2^8 * 2^8 * 2^8 = 2^{24}$ **hôtes**

  Classe B : 2 octet s 2^{16} **réseaux et** $2^8 * 2^8 = 2^{16}$ **hôtes**

  Classe C : 3 octets 2^{24} **réseaux et** $2^8 = 256$ **hôtes**

Adresse de réseau

* Si on peut identifier la partie réseau (net_id)
  – Il suffit d'annuler la partie hôte pour obtenir l'adresse IP du réseau Ex: 173.80.12.56

Partie réseau **173.80**

Partie hôte **12.56**

IP réseau : **173.80.0.0**

- Par calcul :

IP réseau = IP hôte & masque

Règles de calcul de l'opérateur & :

0 &0 -> 0 , 0 & 1 = 1 & 0 -> 0 , 1 & 1 -> 1

Adresse de diffusion

- La **diffusion** consiste à communiquer avec plusieurs composants en même temps. S'il s'agit de communiquer avec des groupes de composants, on parle de **multi-diffusion**.

Adresse de diffusion ou broadcast :

-Elle est obtenue en passant tous les bits de l'hôte à 1 ou par calcul :

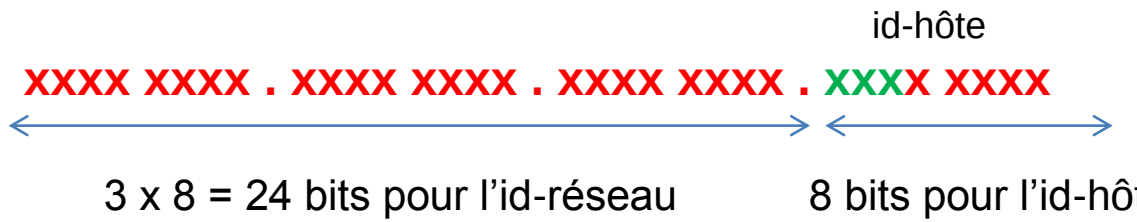
IP diffusion = IP_hôte OU inv(Masque)

- Règles de l'opérateur OU (noté **V**)
  - **1 V 1 -> 1**
  - **1 V 0 = 0 V 1 -> 1**
  - **0 V 0 -> 0**

  Exemples

Sous-réseaux IP

Le masque de sous-réseau permet d'augmenter le nombre de réseaux, **sans augmenter le nombre d'hôtes adressables**. Il s'agit d'étendre l'id réseau aux premiers bits de l'ID machine afin de créer des **sous-réseaux**.
Conséquence immédiate, les masques ne sont plus ceux définis par défaut !
Soit un réseau de classe C d'adresse 192.168.20.0 / 255.255.255.0
Le format de l'IP est le suivant :



Règle de création des sous-réseaux :

-pour créer des sous-réseaux, on **récupère quelques bits** qui font partie de l'ID hôte et on les met à « 1 » dans le masque

si 1 bit => 2^1 **= 2 sous-réseaux 1xxx xxxx**
si 2 bits => 2^2 **= 4 sous-réseaux … 11xx xxxx**
si n bits => 2^n sous-réseaux 111x xxxx

Ex : Pour créer **4** sous-réseaux avec l'adresse 130 . 90 . 0 . 0, on utilise **deux bits** du 3ème octet : **1100 0000** => **192** d'où le nouveau masque de **255.255.192.0**

Avantages des sous-réseaux

- Limitation des domaines de diffusion
- Segmentation des sous-réseaux d'où un cloisonnement des domaines de diffusion
- Limitation de la propagation des virus et des messages des différents services

Sur-réseaux IP et agrégation

- **Pour créer des sur-réseaux (ou super-réseaux), on récupère quelques bits de la partie réseau pour les incorporer dans l'id-hôte. Ce qui permet d'augmenter le nombre d'hôtes au détriment des réseaux :**

Par ex, en classe C , on a 24 bits pour le réseau et 8 bits pour les hôtes => 2^8 -2 = 256 -2 =254 adresses

XXXX XXXX . XXXX XXXX . XXXX XXXX . XXXX XXXX

2x 8 + 6 = 22 bits pour l'id-réseau 10 bits pour l'id-hôte

L'adresse IP peut alors s'écrire **X.Y.Z.T/22**

Ex : 194.20.35.0/22

Nombre de sur-réseaux : 2^2 = 4 sur-réseaux

Nombre d'adresses de composants par sur-réseau 2^10 -2 = 1024 -2

Exercice

- Soit le réseau suivant 192.168.30.0/24. Il comporte au plus **254** adresses utiles.
- Quel découpage effectuer pour disposer de **504** adresses utiles , **852** adresses utiles ?

-Préciser et justifier votre réponse

(Adresse en notation CIDR, masque, etc)

Sous forme binaire, on utilise, par habitude, et non par obligation des bits à la valeur 1 contigus, ce qui donne les possibilités suivantes qu'on retrouve dans les masques de sous-réseaux :

Binaire	Décimal
0000 0000	**0**
1000 0000	**128**
1100 0000	**192**
1110 0000	**224**
1111 0000	**240**
1111 1000	**248**
1111 1100	**252**
1111 1110	**254**
1111 1111	**255**

Réseaux privés

Classe	Réseau CIDR	Adresses réseaux	Masque
A	10.0.0.0/8	10.0.0.0 - 10.255.255.255	255.0.0.0
B	172.16.0.0/12	172.16.0.0 - 172.31.255.255	255.240.0.0
C	192.168.0.0/16	192.168.0.0 - 192.168.255.255	255.255.0.0

Classe réseau privé	Réseau IP *(1ère écriture)*	Réseau IP *(2nd écriture)*	Nombre de sous-réseaux	Exemple d'adresses IP	Nombre de machines par réseau
A	10.0.0.0 / 255.0.0.0	10.0.0.0 / 8	1	De 10.0.0.1 à 10.255.255.254	16 777 214
B	172.16.0.0 / 255.255.0.0	172.16.0.0 / 16	1 / 16	De 172.16.0.1 à 172.16.255.254	65 024
	172.17.0.0 / 255.255.0.0	172.17.0.0 / 16	2 / 16	De 172.17.0.1 à 172.17.255.254	65 024
	...	...	...	...	65 024
	172.31.0.0 / 255.255.0.0	172.31.0.0 / 16	16 / 16	De 172.31.0.1 à 172.31.255.254	65 024
C	192.168.0.0 / 255.255.255.0	192.168.0.0 / 24	1 / 256	De 192.168.0.1 à 192.168.0.254	254
	192.168.1.0 / 255.255.255.0	192.168.1.0 / 24	2 / 256	De 192.168.1.1 à 192.168.1.254	254
	...	...	...	...	254
	192.168.255.0 / 255.255.255.0	192.168.255.0 / 24	256 / 256	De 192.168.255.1 à 192.168.255.254	254

Notion de service réseau

- **SERVICE = Programme /daemon + Port**
- **Exemples de services/port d'écoute**

Service	Programme	Port (TCP)
FTP	ftpd	21
HTTP	httpd	80
SSH	sshd	22
MYSQL	mysqld	3306
DNS	bind	53

Service HTTP

- Daemon/programme : httpd (Apache)
- Port d'écoute (TCP/UDP) : 80
- Client HTTP : tout navigateur/browser
- Sécurisation des accès
  - **htaccess** (Fichier **.htaccess** et apache2.conf)
    - Basic (cryptage 128 bits), cryptage MDA (> 128 bits)
    - Faiblesse du mode de sécurisation
  - **SSL** => protocole sécurisé **https**
    - Plus difficile à « casser »

Savoir faire/trouver

- Qui, quel hôte a accédé au serveur ?
  - Adresse IP, Domaine ?
  - (Fichier **access.log**)
- Réglementer les accès aux pages
  - Directives **Allow**, **Deny**

Sécurisation par htaccess

- Création du répertoire à sécuriser mkdir /chemin/dossier
  - Dans l'arborescence du site mkdir /var/www/repertoire
- Insertion du fichier **.htaccess** dans le répertoire à protéger

AuthUserFile /chemin/fichier_users_apache

AuthGroupFile /chemin/fichier_groupes

AuthName " Acces controle "

AuthType Basic

require valid-user (tous ceux qui ont un compte)

ou require user nom

ou require group nom_groupe

Procédure htaccess (suite)

- **Création d'un bloc dans apache2.conf :**

```
<Directory  /var/www/dossier_a_proteger>
        AllowOverride All   (pour activer le contrôle)
        Order allow, deny
        Allow from all          # Contrôle pour tous users
        Allow from 192.168.
        Deny from  domaine.com
        Deny from  192.168.20.0/24
</Directory>
```

- Création d'un compte utilisateur pour Apache :

htpasswd [-c] .fichier_users dupont

(attention : -c est utilisé à la première exécution).

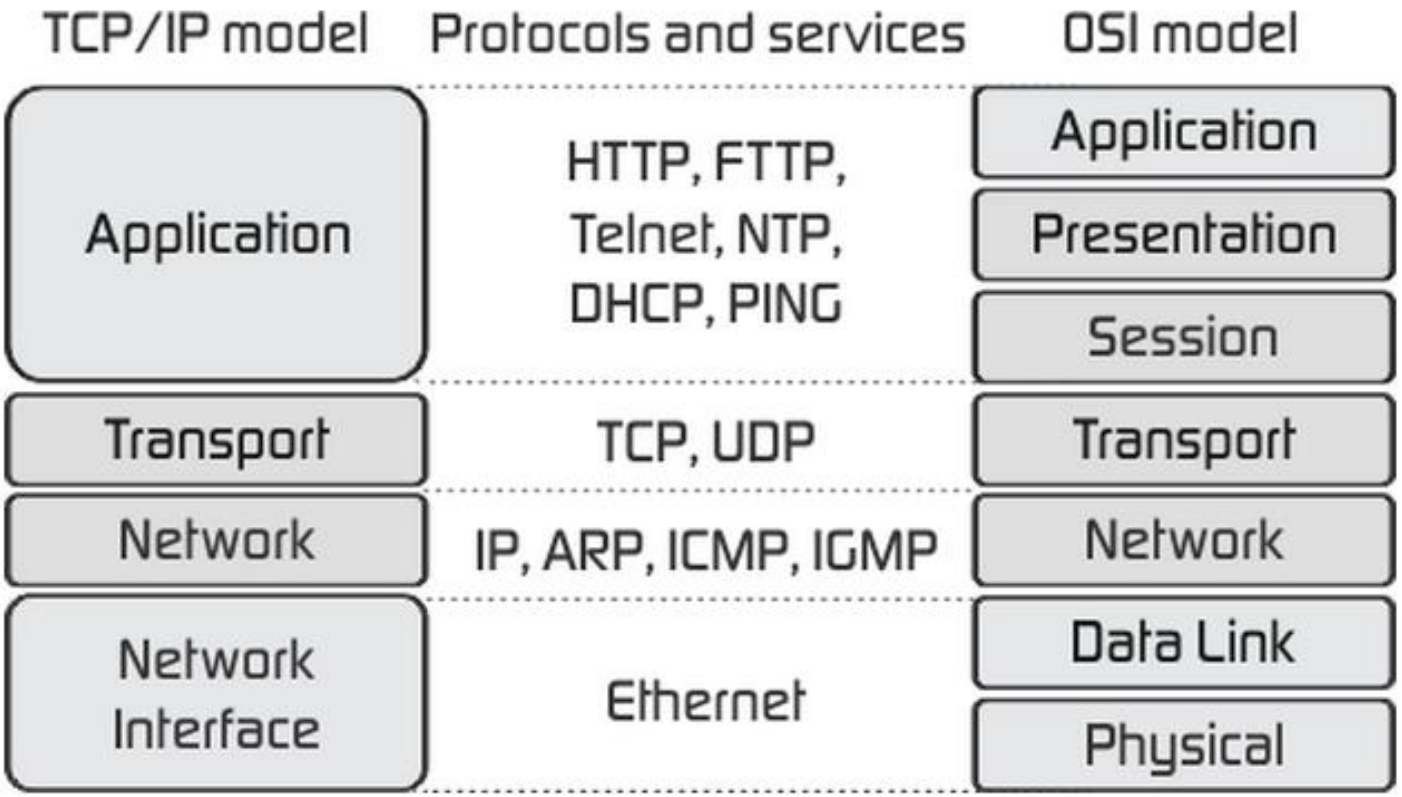
fichier_users est le nom du fichier d'authentification.

Routage inter-réseaux

- **Principe et fonctionnement du routage IP**

- Lorsque le **routeur** reçoit une trame, il examine son contenu
- -si la destination (réseau) est la même que la source, il ne fait pas passer la trame, cette dernière est transmise à la bonne carte destinataire sur le réseau local (table ARP). On parle dans ce cas d'une ***remise directe***
- -si la destination est différente de la source, le routeur transmet la trame à son ***interface connectée à la destination***, c'est la redirection/routage de la trame
- -si la destination est inconnue du routeur (absence dans sa table de routage), le routeur recherche une ***destination par défaut*** (route par défaut) et envoie la trame vers ce réseau
- -sinon, la trame est bloquée.

Rappel : protocole des couches



Format d'un Datagramme IP

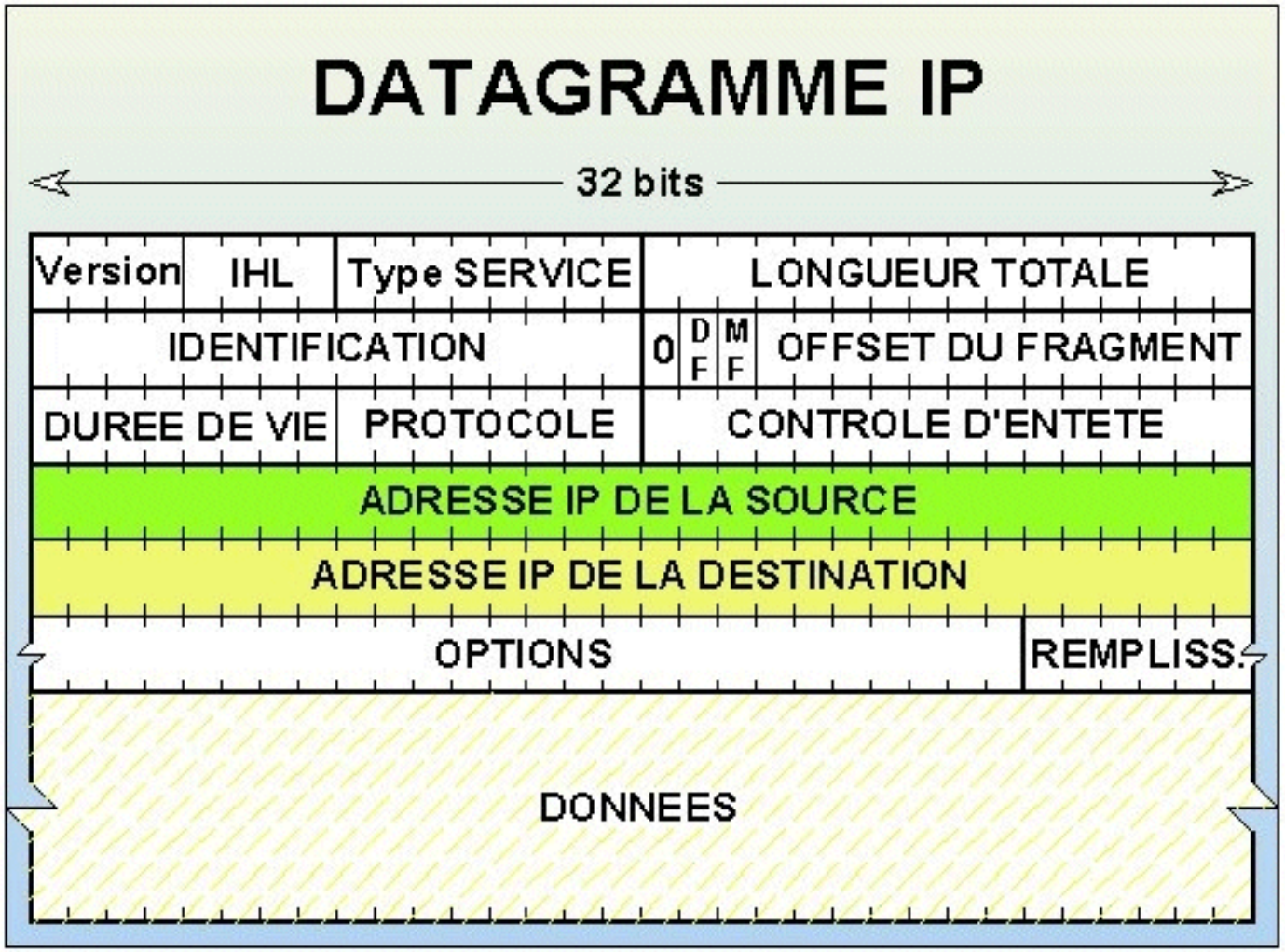


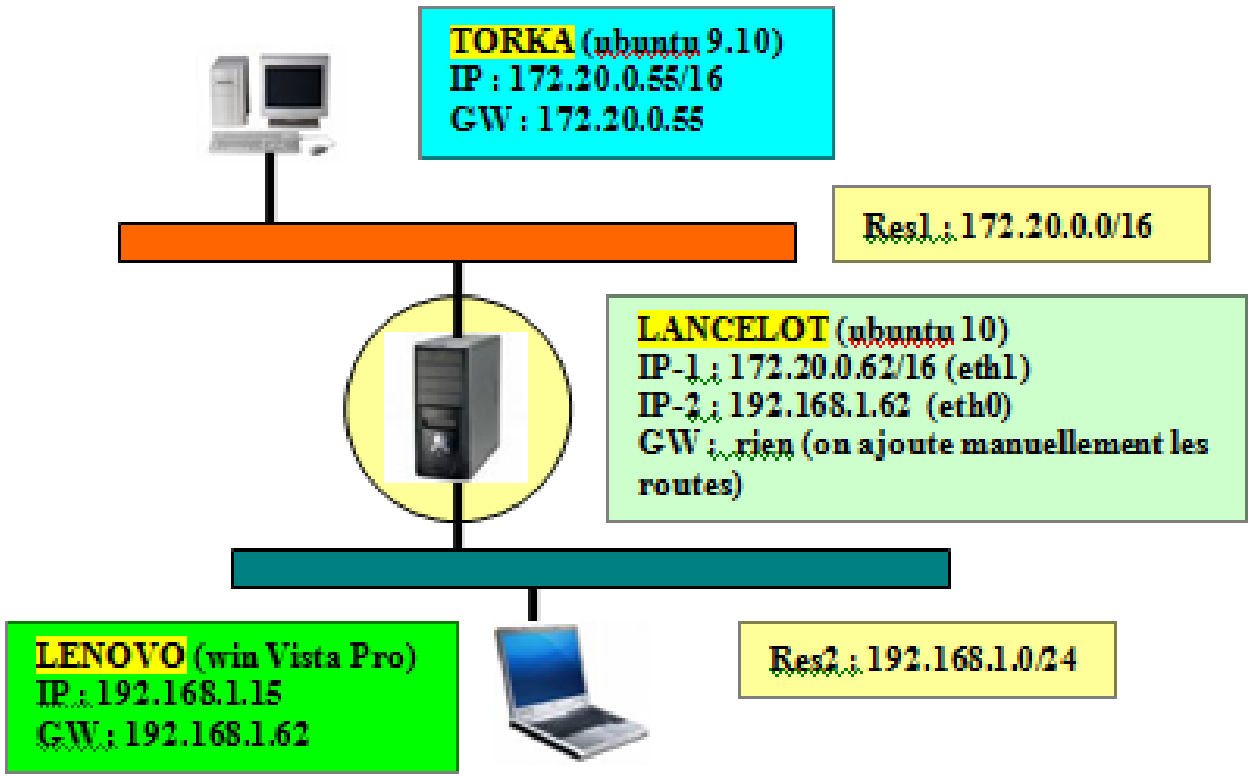
Table de routage

- Elle liste les **réseaux destination** connus et précise l'**interface utilisée** pour les atteindre

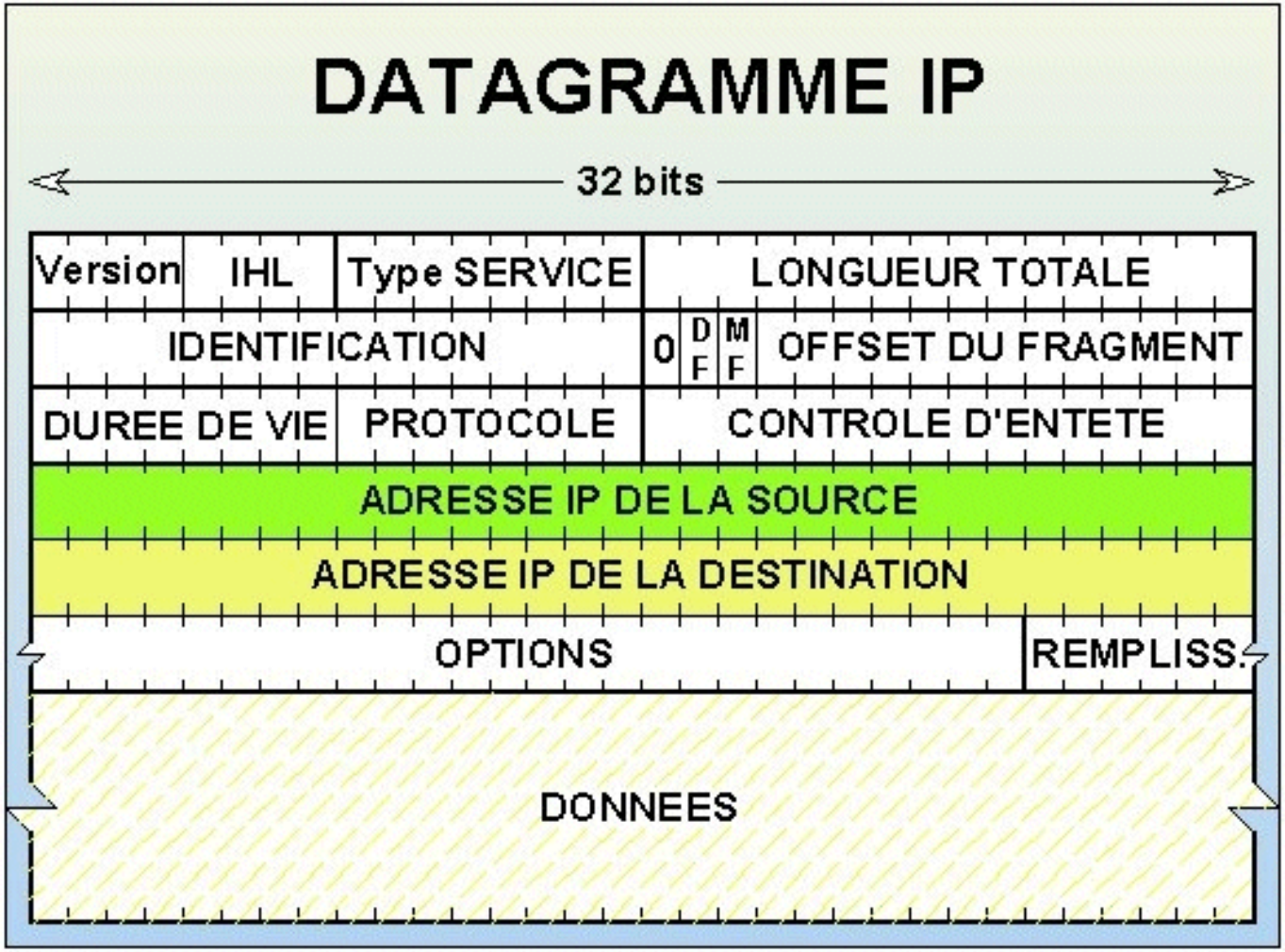
```
===========================================================================

IPv4 Table de routage
===========================================================================
Itinéraires actifs :
Destination réseau    Masque réseau  Adr. passerelle   Adr. interface Métrique
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
      192.168.1.0    255.255.255.0         On-link      192.168.1.16    276
     192.168.1.16  255.255.255.255         On-link      192.168.1.16    276
    192.168.1.255  255.255.255.255         On-link      192.168.1.16    276
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link      192.168.1.16    276
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link      192.168.1.16    276
===========================================================================
Itinéraires persistants :
  Adresse réseau    Masque réseau  Adresse passerelle Métrique
          0.0.0.0          0.0.0.0      192.168.1.62  Par défaut
===========================================================================
```

Routage des datagrammes

Mise en œuvre du routage entre deux réseaux (routeur sous linux)



Format (simplifié) d'un datagramme ou trame IP



Routage des datagrammes

- **Entre 3 réseaux et plus** :

res1 : 192.168.10.0/24 res2 : 172.20.0.0/16 res3 : 130.100.0.0/16

Routeur « R » avec les interfaces eth0 : 192.168.10.254, eth1 : 172.20.0.254 et eth2 : 130.100.0.254

Les « passerelles par défaut » sont :

eth0 pour le réseau res1, eth1 pour res2 et eth2 pour res3.

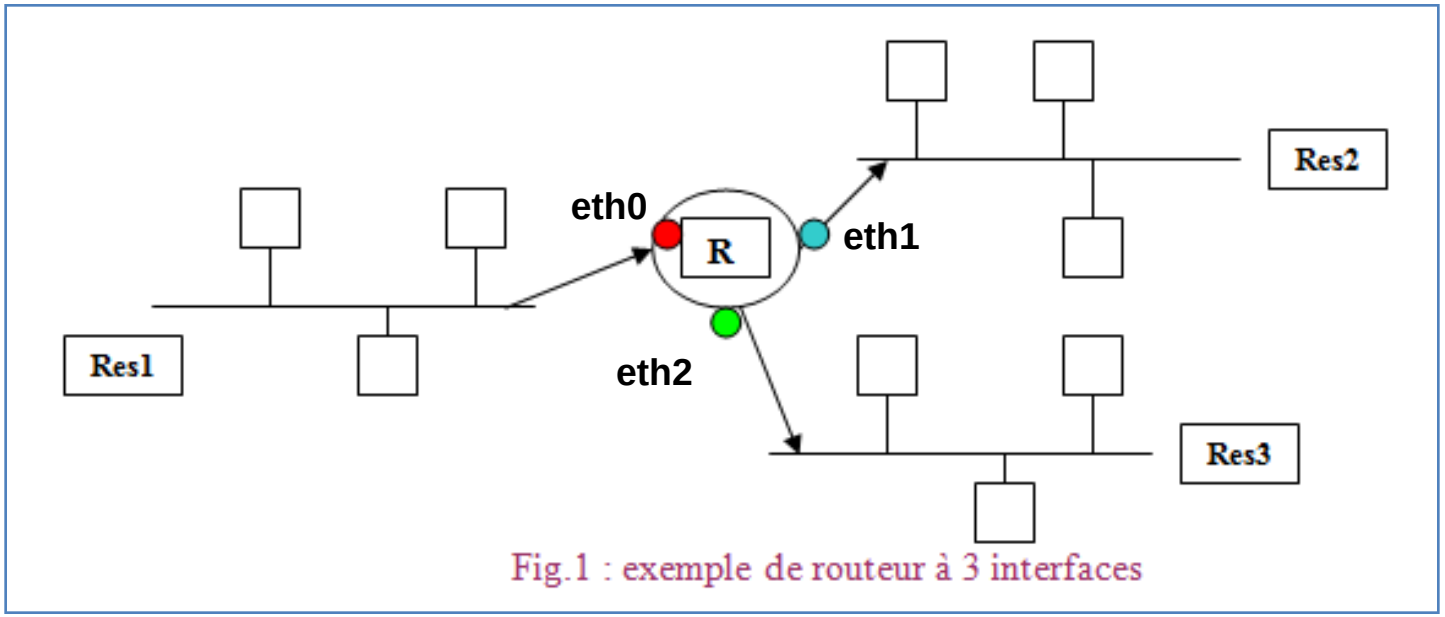


Fig.1 : exemple de routeur à 3 interfaces

Activation de la fonction de routage sur le routeur

- Pour **activer le routage**, il faut mettre à **1** le paramètre **ip_forward** qui par défaut vaut **0** (routage des datagrammes désactivé). Il suffit donc de faire en ligne de commande :
- **echo 1 > /proc/sys/net/ipv4/ip_forward**
- **Vérifier avec cat / proc/sys/net/ipv4/ip_forward**
- Pour rendre cette activation permanente, modifier le fichier **/etc/syscntl.conf** et décommenter la ligne :

  net.ipv4.ip_forward = 1
- Activation en faisant **sysctl -p /etc/sysctl.conf**

Manipulation de routes statiques

- Les commandes **route** permettent d'ajouter, supprimer ou modifier les routes statiques. Syntaxe différente win - linux
  - Par exemple, on **ajoute les nouvelles routes** pour atteindre les réseaux voisins avec **route add –net destination**

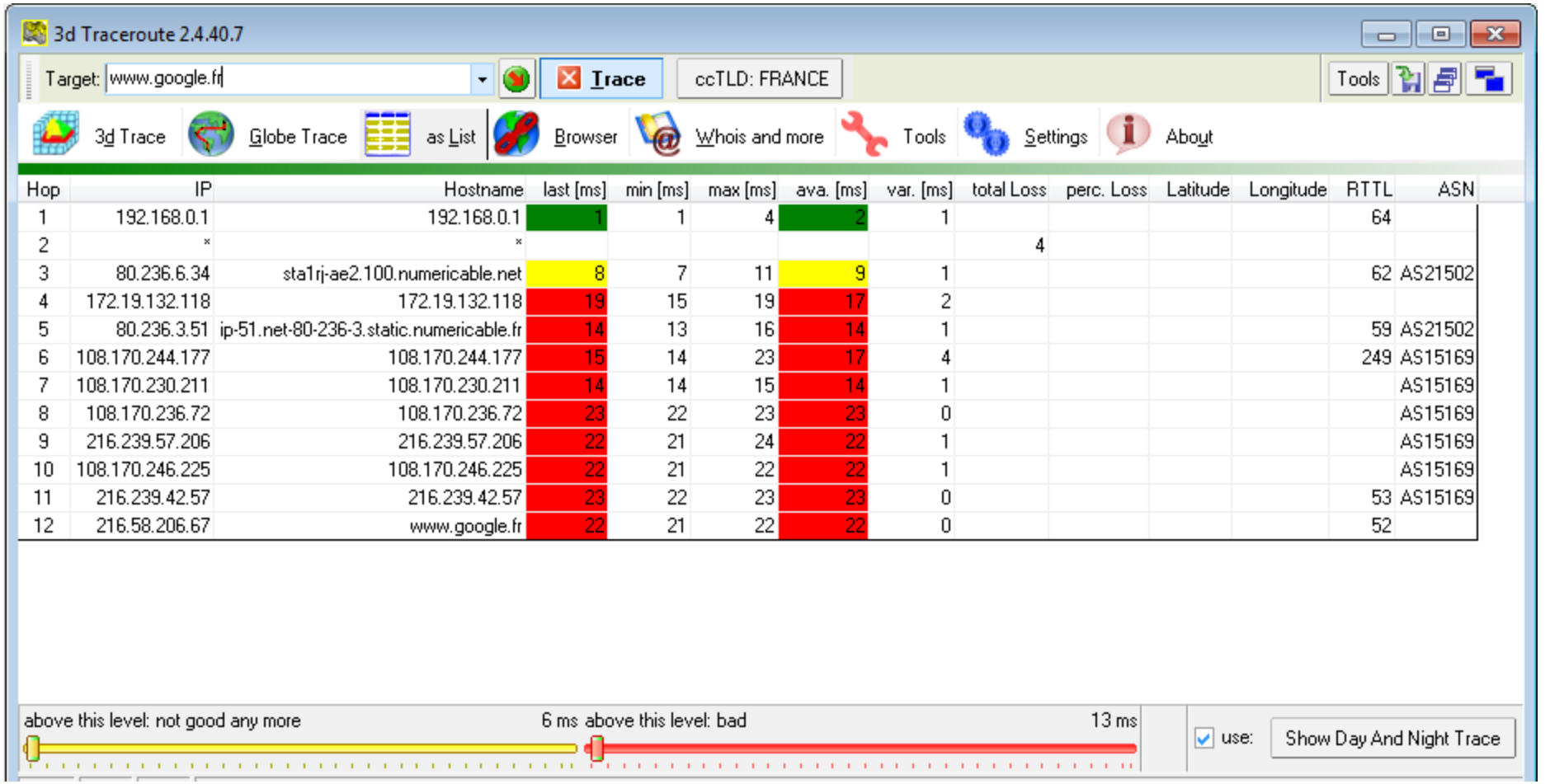
route add –net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.62

route add –net 172.20.0.0 netmask 255.255.0.0 gw 172.20.0.62

- On peut supprimer une route avec route del -net destination ou route delete destination

-

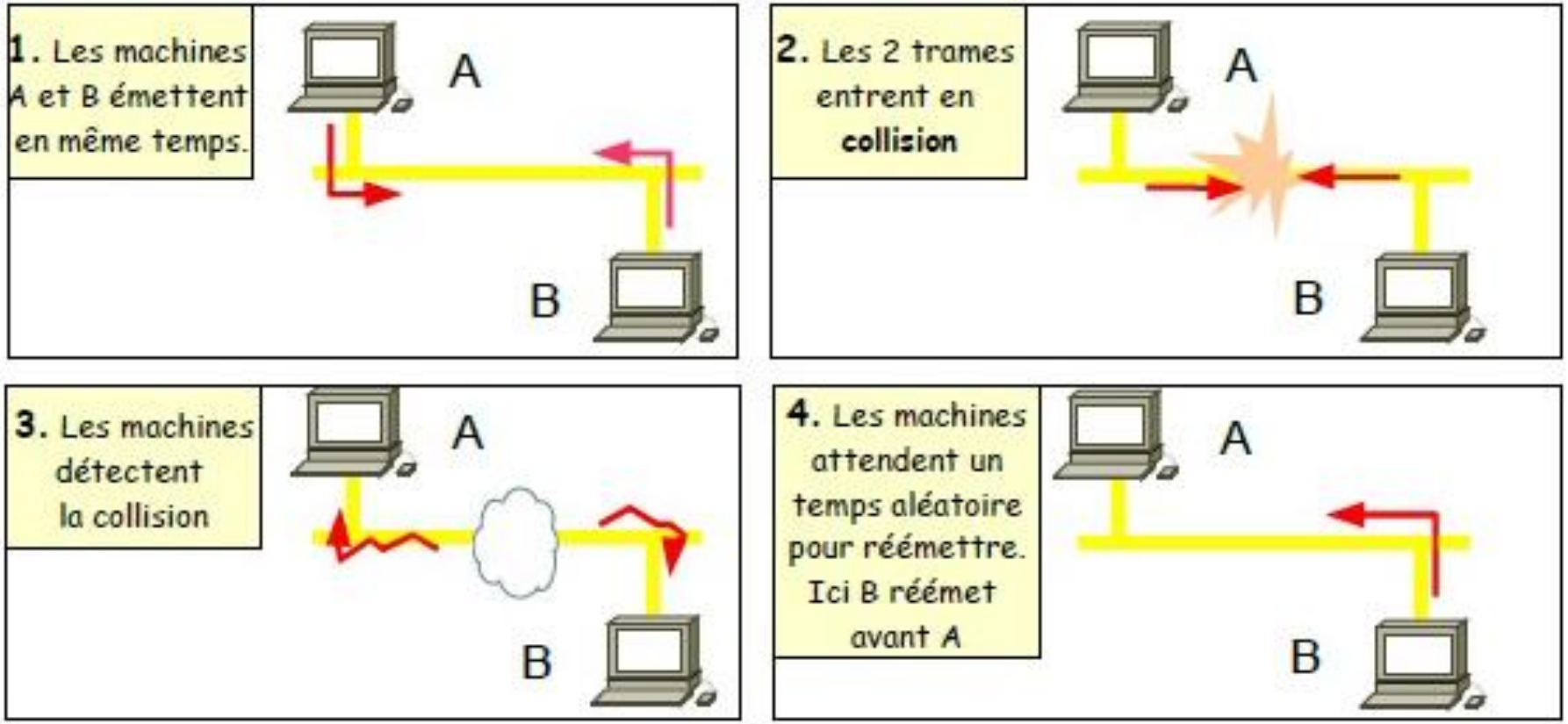
Traceurs de routes



Hop	IP	Hostname	last [ms]	min [ms]	max [ms]	ava. [ms]	var. [ms]	total Loss	perc. Loss	Latitude	Longitude	RTTL	ASN
1	192.168.0.1	192.168.0.1	1	1	4	2	1					64	
2	*	*						4					
3	80.236.6.34	sta1rj-ae2.100.numericable.net	8	7	11	9	1					62	AS21502
4	172.19.132.118	172.19.132.118	19	15	19	17	2						
5	80.236.3.51	ip-51.net-80-236-3.static.numericable.fr	14	13	16	14	1					59	AS21502
6	108.170.244.177	108.170.244.177	15	14	23	17	4					249	AS15169
7	108.170.230.211	108.170.230.211	14	14	15	14	1						AS15169
8	108.170.236.72	108.170.236.72	23	22	23	23	0						AS15169
9	216.239.57.206	216.239.57.206	22	21	24	22	1						AS15169
10	108.170.246.225	108.170.246.225	22	21	22	22	1						AS15169
11	216.239.42.57	216.239.42.57	23	22	23	23	0					53	AS15169
12	216.58.206.67	www.google.fr	22	21	22	22	0					52	

Le principe du **CSMA/CD**

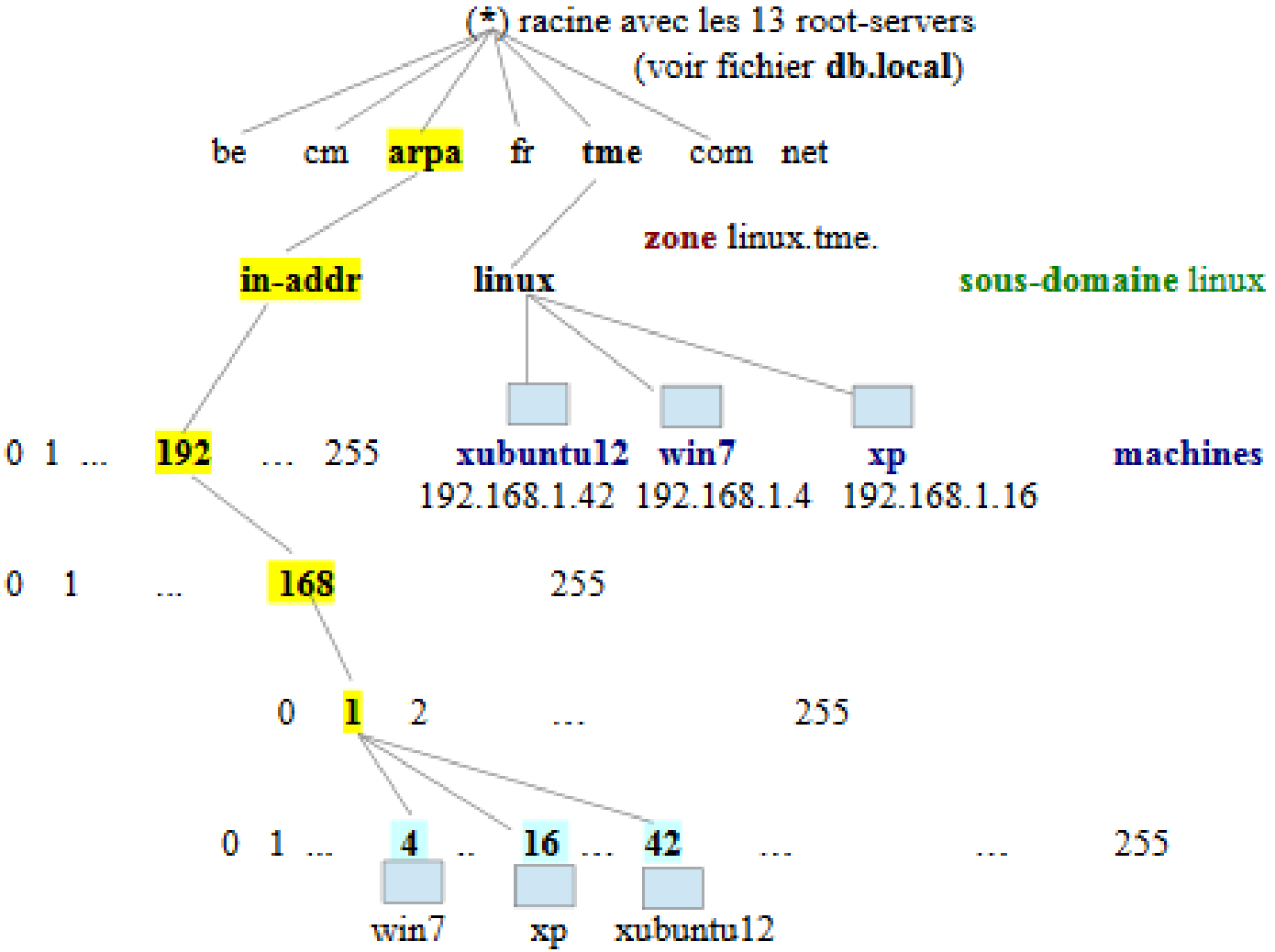
Une machine qui souhaite transmettre sur le réseau écoute le câble. Si la voie n'est pas libre, elle attend jusqu'à ce que l'autre machine ait fini de transmettre. Si deux machines commencent à émettre en même temps et qu'il y a collision, elles arrêtent d'émettre, attendent toutes deux un temps aléatoire pour réemettre de manière à ne plus entrer en collision.



Le service DNS

- But : **traduire ou convertir les noms en adresses IP et inversément**
  - Nom → @ IP : résolution directe
  - @IP → Nom : résolution inversée
- Le service fait suite au **fichier hosts** qui proposait cette correspondance
  - Dans **linux** : /etc/**hosts**
  - Dans **windows** : C:\windows\system32\drivers\etc**hosts**

Le service DNS



Enregistrements DNS (RR)

- **SOA** (Start Of Authority) : Serveur d'autorité sur la zone
- **NS** (Name server) : Serveur de nom
- **A** (Address) : Adresse
- **CNAME** (Canonical Name) : Alias
- **MX** (Mail eXchanger) : Serveur de messagerie
- **TXT** (Texte) Texte simple

Fchiers de zone DNS

Fichier de zone directe /etc/bind/linux.tme.hosts

```
$TTL    86400
@       IN      SOA     xubuntu12.linux.tme.    root.linux.tme. (
                        2014032105              ; Serial
                         604800                 ; Refresh
                          86400                 ; Retry
                        2419200                 ; Expire
                          86400 )               ; Negative Cache TTL
;---    serveur
@                               IN  NS   xubuntu12.linux.tme.
xubuntu12.linux.tme.            IN   A 192.168.1.42
;--------------- clients ---------------------------------
xubuntu12.linux.tme.            IN   A 192.168.1.42
win7.linux.tme.                 IN   A 192.168.1.4
xp.linux.tme.                   IN   A 192.168.1.16
```

Fichiers de zone

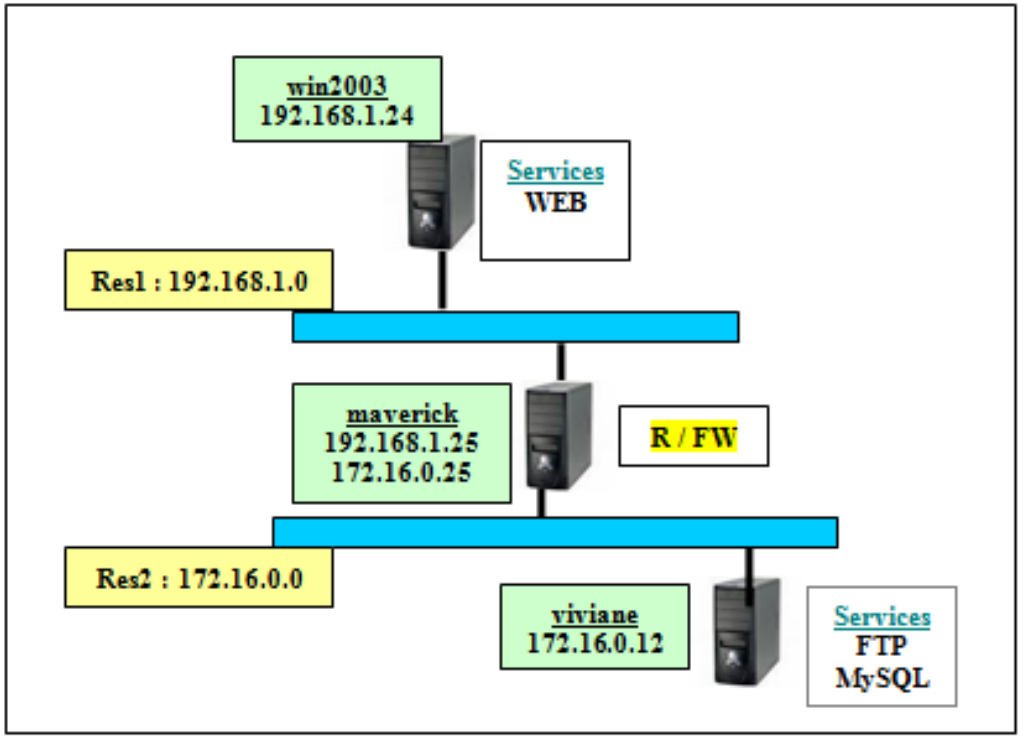
Résolution inversée(REVerse) : IP => nom

Fichier associe : /etc/bind/linux.tme.rev

```
$TTL    604800
@       IN      SOA     xubuntu12.linux.tme. root.linux.tme. (
                        2014032105      ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
; ---   serveur
@       IN      NS      xubuntu12.linux.tme.
42      IN      PTR     xubuntu12.linux.tme.
; ------------- clients ---------------------------
4       IN      PTR     win7.linux.tme.
16      IN      PTR     xp.linux.tme.
```

Filtrage de paquets

Architecture du réseau logique



Règles de filtrage à paramétrer et tester

Destination	Source	Protocole	Port	ACTION
192.168.1.25	192.168.1.24	ICMP (ping)		REFUSER
192.168.1.24	192.168.1.25	TCP	80	REFUSER
192.168.1.25	192.168.1.10	TCP	3306	ACCEPTER

Le principe du filtrage des paquets TCP

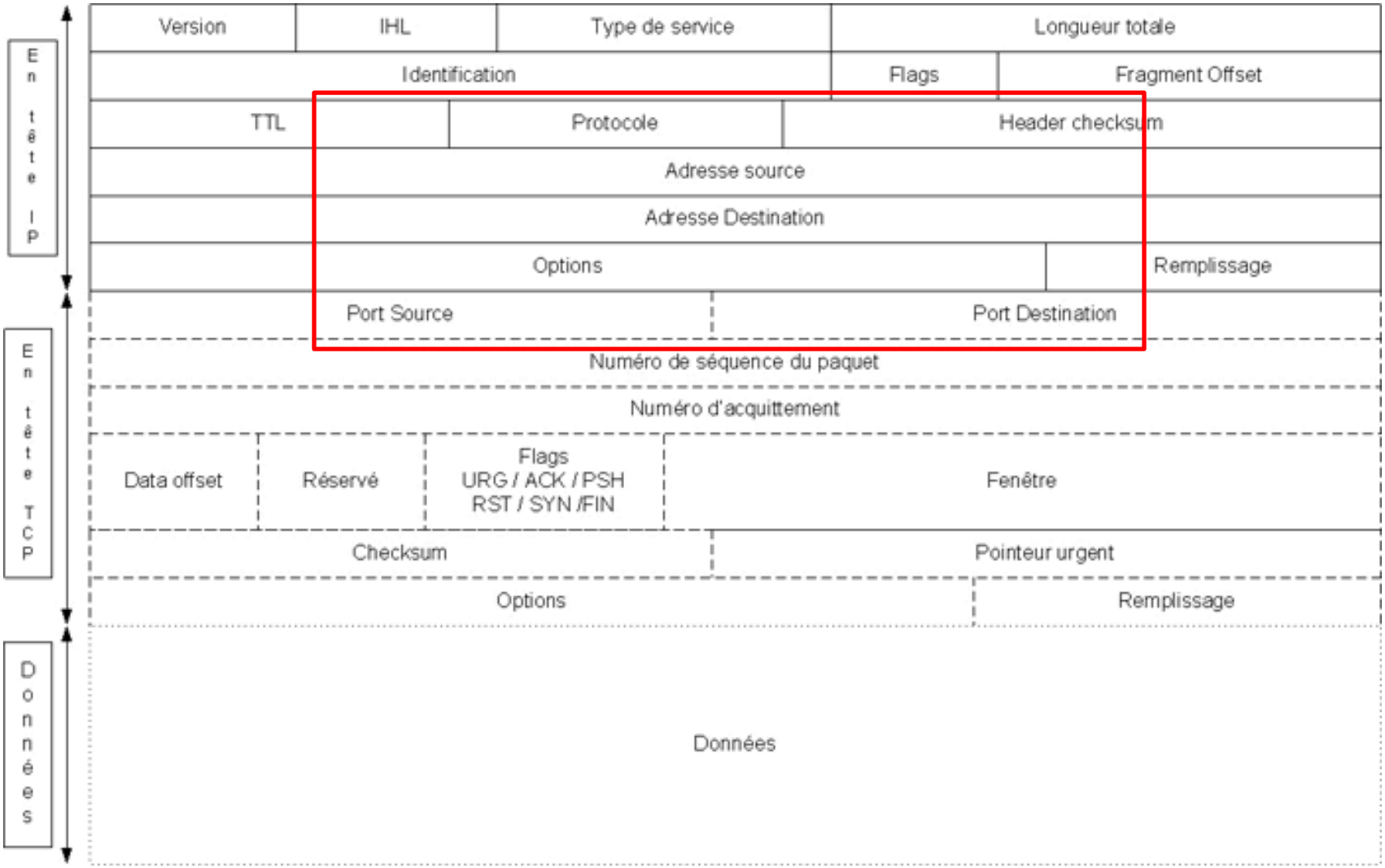
- Le filtre encore appelé pare-feu ou **firewall** fonctionne au niveau Transport (couche 4 de l'OSI) et a besoin d'informations supplémentaires en plus de sa connaissance des réseaux **source** et **destination** des datagrammes : les **protocoles** et les **ports**.
- En effet, le **pare-feu** est d'abord un routeur puisque sa fonction est d'orienter les données satisfaisant à certaines conditions.

  Il examine les protocoles concernés par chaque paquet de données et applique les règles prévues par son configurateur.

Il existe des pare-feux en mode graphique comme **GUFW** (**Gnome Uncomplicated FireWall**) qui utilise **iptables (Netfilter)** du noyau linux.

Iptables montre toute sa puissance en ligne de commande. Pour le maîtriser, il vaut mieux commencer son apprentissage en ligne de commande comme souvent sous linux.

Structure des paquets



Fonctionnement du filtrage

- **Le travail du filtre de paquet consiste à examiner les ports source et destination, les adresses IP source et destination ainsi que les protocoles concernés par le paquet.**
- **Il s'appuie sur des règles pour décider de la suite, ACTION, à donner au paquet analysé :**
- **Laisser passer (ACCEPT) dans la syntaxe iptables**
- **Bloquer le paquet (DROP) pour refuser le passage**
- **Refuser le passage et supprimer le paquet**
- **Loguer, c'est-à-dire, enregistrer (LOG) dans un fichier, la tentative de traversée du filtre**
- **Masquer le paquet (MASQUERADE) ou le renvoyer (REJECT)**

L'utilitaire *iptables*

(permet d'écrire les règles du pare-feu)

- Syntaxe de iptables :

iptables [-t TABLE (Filter, NAT, Mangle)]

-A Chaîne (INPUT, OUTPUT, FORWARD)

-i interface_source

-j interface_sortie

-s IP_source -d IP_destination

--dport port_destination

-p protocole

-j ACTION (ACCEPT, DROP, REJECT, LOG)

Exemples de règles

- **# iptables -A INPUT -p tcp -i eth0 --dport 80 -j ACCEPT**
- **# iptables -A INPUT -p tcp -i eth0 --dport ssh -j ACCEPT**
- **# iptables -I INPUT 2 -i lo -j ACCEPT** # Autorisation du trafic local (lo)
- **# iptables -P INPUT DROP** # On bloque tout le reste
- Pour autoriser à faire des "pings" sur des IP externes (en sortie) :

  # iptables -A OUTPUT -p icmp -m state --state NEW, ESTABLISHED,RELATED -j ACCEPT
- # Pour autoriser les pings en entrée :

  # iptables -A INPUT -p icmp -j ACCEPT

Note

Ce document ne représente qu'une partie des notions et thèmes que j'enseigne.

La totalité du cours sera bientôt disponible

…

Cordiales salutations.

Henri TSOUNGUI

Ingénieur CNAM en Informatique

Option Conception et Gestion des Systèmes d'Information

Professeur Certifié Major au CAPET D (Economie et Gestion)

Institut des Sciences et Techniques de Valenciennes (ISTV)

Université de Valenciennes et du Hainaut-Cambrésis (UVHC)

henri.tsoungui@univ-valenciennes.fr