

# Internet

1. Accès au réseau
  - 2 accès spécifiques Internet
  - Filaire Ethernet, et sans fil WiFi
2. Architecture
  - Organisation globale
  - Acteurs publics et privés
  - Adressage
3. Protocoles
  - IP le protocole d'Internet
  - Couche 3 et couche 4
  - Routage
4. Services
  - Web
  - Focus sur le VPN

2

RES101  
Internet

## 1. ACCÈS AU RÉSEAU

3

# Réseaux locaux

- Réseau local
  - LAN : Local Area Network
  - Taille < 1km, bâtiment ou ensemble de bâtiments
  - Connexion des utilisateurs terminaux à Internet
- Internet
  - Filaire : Ethernet
  - Sans fil : WiFi
- Transmission point-à-point
  - Couche 1 Physique
  - Couche 2 Liaison des données

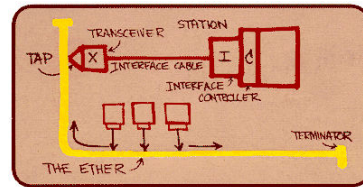
4

# Ethernet

5

# Le standard Ethernet

- Ethernet correspond à la norme IEEE 802.3
- Rôles :
  - Couche Physique
    - Transmission de bits sur l'interface physique
  - Couche Liaison
    - Accès au médium partagé
    - Formation de trames
- Historique :
  - 1975 : Projet de recherche des laboratoires Xerox Parc. « Ethernet expérimental » à 2.94 Mbps sur câble coaxial.
  - 1980 : Ethernet Version I, proposé par DEC et INTEL à 10 Mbps.
  - 1982 : DEC, INTEL et Xerox (DIX) proposent Ethernet version II
  - 1984 : Standards 802.2+802.3 proposés par l'IEEE (compatible avec DIX v2) : thick Ethernet 10 Base 5 (10Mbps)
  - 1985 : 802.3a (thin Ethernet 10 Base 2)
  - 1990 : 802.3i (10 Base T)
  - 1993 : 10 Base F
  - 1995 : Fast Ethernet 100 Mbps
  - 1997 : Mode Full duplex
  - 1998 : Gigabit Ethernet 1 Gbps
  - 2003 : Gigabit Ethernet 10 Gbps
  - 2015 : 100 Gbps Ethernet



Dessin initial de Robert Metcalfe

6

# Le standard Ethernet

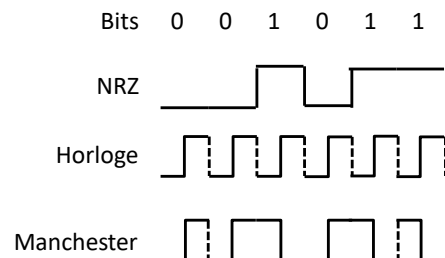
- Types de câblage
  - Nomenclature X base t
    - X est le débit en Mbps
    - Base signifie que la transmission s'effectue en bande de base
    - t donne la longueur maximum d'un segment à l'origine, ou le type de câble maintenant
  - 10 base 5 (obsolète)
    - Câble coaxial épais avec prises vampire
    - Topologie en bus (longueur max 500m)
  - 10 base 2 (obsolète)
    - Câble coaxial fin avec prises en T
    - Topologie en bus (longueur max 185m)
  - 10 base T
    - Paire torsadée (T = Twisted pair) en cuivre (câble téléphonique) avec connecteur RJ45
    - Topologie en étoile
    - 100 base T : Fast Ethernet (principalement 2 paires torsadées)
    - 1000 base T : Gigabit Ethernet (4 paires torsadées)
    - 10G base T, 100G base T ...
  - 10 base F (presque jamais commercialisé)
    - Fibre optique
    - Topologie en étoile



7

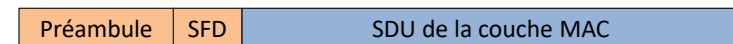
## Codage

- Codage**
  - Représentation du signal numérique en signal analogique
- NRZ (Non Return to Zero)**
  - Bit 0 => 0V et Bit 1 => 5V
  - Problème pour différencier les séquences « 0011 » et « 01 » par exemple en l'absence de synchronisation
- Manchester**
  - Ajout d'une horloge
  - Bit 0 => transition 0V->5V et Bit 1 => transition 5V->0V
- Code en bloc**
  - Introduction de redondance par blocs de bits avant le codage
  - Exemple 4B5B : 4 bits de données sont transformés en 5 bits



8

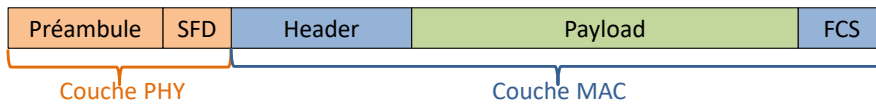
## Trame physique



- Les séquences de bits envoyées sur le support physique sont délimitées en « trames »
- Une trame physique correspond au PDU (Protocol Data Unit) de la couche 1
- Elle comprend le SDU (Service Data Unit) venant de la couche supérieure encadré par un marqueur de début de trame et éventuellement un marqueur de fin de trame, ce qui constitue le PCI (Packet Control Information)
- Le marqueur de début de trame est une séquence connue de bits qui permet la synchronisation, on l'appelle préambule, suivi d'une séquence indiquant le début de la trame MAC
  - Exemple : 01010101
- La longueur d'une trame physique Ethernet peut être variable ou fixe selon la technologie

9

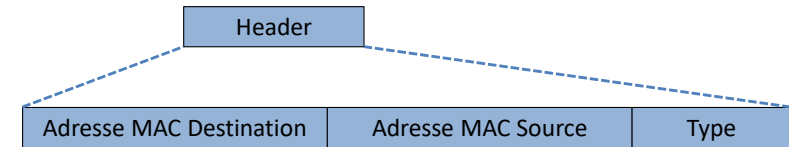
# Trame Ethernet



- Préambule :
  - Synchronisation de la couche physique
- SFD (Start Frame Delimiter) :
  - Indique le début de la trame
  - 10101011
- Ethernet header
  - En-tête
- Payload :
  - Contenu de la trame issu des couches supérieures (IP)
- FCS (Frame Check Sequence) :
  - Bits de redondance pour la détection d'erreur
  - Checksum basé sur CRC

10

# En-tête Ethernet



- Adresses MAC sources et destination
  - de la liaison point-à-point
  - sur 6 octets
- Type
  - Indique le type du payload, i.e. le protocole de la couche supérieure (3)
    - 0x0800 : IPv4
    - 0x86DD : IPv6
    - 0x0806 : ARP
  - Sur 2 octets
  - Dans Ethernet I, ce champ indiquait la longueur de la trame. Par rétro-compatibilité, la règle suivante est appliquée :
    - Si valeur ≤ 1536 : le champ EtherType indique la longueur de la trame (Ethernet I)
    - Si valeur > 1536 : le champ EtherType indique le protocole de couche 3 (Ethernet II)

11

# Adresse MAC

- Une adresse MAC est
  - Universelle et unique : chaque interface reliée à Ethernet a une adresse différente
  - Composée de 6 octets
    - 3 octets = Identifiant du constructeur (OUI: Organizationally Unique Identifier)
    - 3 octets = Identifiant de l'interface attribué par le constructeur (16 millions d'adresses par OUI)
  - Écrite en hexadécimal séparé par « : »
- Exemple :

98:01:A7:90:51:25

Constructeur    Équipement  
(carte Ethernet)

12

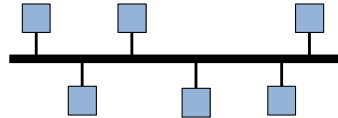
# Adresses MAC particulières

- Une adresse MAC est composée de 6 octets, chaque octet correspond à 8 bits
- Si le 7<sup>ème</sup> bit du 1<sup>er</sup> octet = 1
  - Les 3 premiers octets ne correspondent pas à un OUI
  - L'adresse MAC est locale (non unique globalement)
  - Si le 8<sup>ème</sup> bit du 1<sup>er</sup> octet = 0 alors c'est une adresse MAC en unicast (1 récepteur)
  - Si le 8<sup>ème</sup> bit du 1<sup>er</sup> octet = 1 alors c'est une adresse MAC multicast (plusieurs récepteurs)
    - Multicast signifie que l'adresse désigne plusieurs équipements
    - Il est possible de configurer un groupe d'équipements locaux identifié par une adresse multicast
    - Si tous les bits de l'adresse sont à 1 alors c'est une adresse de broadcast (tout le monde est récepteur)
- L'adresse de broadcast correspond à une adresse MAC dont l'ensemble des bits vaut 1 :
  - ff:ff:ff:ff:ff:ff
  - Elle permet de joindre tous les nœuds d'un réseau local

13

# Fonctionnement d'Ethernet

- Une trame Ethernet est envoyé sur le canal
- Tous les équipements présent sur le canal lisent l'en-tête de la trame
- Si un équipement reconnaît son adresse dans le champ « Adresse MAC destination » alors il ouvre la trame et transmet le payload à la couche supérieure identifiée par le champ « EtherType »
- Si le canal est partagé :
  - Topologie en bus (Ethernet 10base2, 10base5)
  - Half duplex (1 paire torsadée utilisée dans les 2 sens)
  - La couche MAC implémente une technique d'accès multiple
  - Principe du CSMA/CD (rappel)
    - Une station écoute le canal avant d'émettre
    - Si le canal est libre elle émet
    - Si une collision est détectée, un signal de brouillage est émis
    - En cas de canal occupé ou de collision détectée, la station attend un délai calculé selon la règle du BEB
- Ethernet moderne :
  - Plus de canal partagé
  - Full duplex (1 paire torsadée au moins pour chaque sens)
  - Grâce aux switches (commutateurs)



14

# Switch

- Un switch ou commutateur Ethernet est un équipement qui implémente une couche 1 et une couche 2 Ethernet
- Il dispose de plusieurs interfaces/ports (entre 2 et 128)
- Un switch à 2 ports est un pont (bridge)
- Il permet :
  - d'isoler les domaines de collisions
  - de connecter différents supports physiques



15

## Table de commutation

- La table de commutation d'un switch est le tableau dans lequel les adresses MAC connues de celui-ci sont associées à ses interfaces :

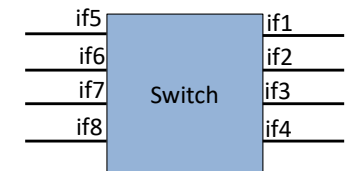
Adresse MAC	Interface / Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4

- Remplissage de la table de commutation :
  - Dès que le switch reçoit une trame, il note l'adresse MAC source pour l'associer à l'interface sur laquelle la trame a été reçue
  - Les informations de la table de commutation ont une durée de vie limitée

16

## Table de commutation

- On considère le switch ci-contre

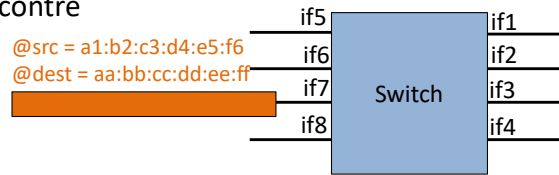


Adresse MAC	Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4

17

## Table de commutation

- On considère le switch ci-contre



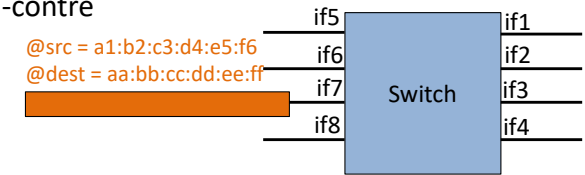
- Il reçoit une trame sur son port if7 dont l'adresse source est a1:b2:c3:d4:e5:f6

Adresse MAC	Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4

18

## Table de commutation

- On considère le switch ci-contre



- Il reçoit une trame sur son port if7 dont l'adresse source est a1:b2:c3:d4:e5:f6
- Il ajoute une ligne dans sa table de commutation

Adresse MAC	Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4
a1:b2:c3:d4:e5:f6	if7

19

## Table de commutation

- La table de commutation d'un switch est le tableau dans lequel les adresses MAC des équipements en lien direct avec celui-ci sont associées à ses interfaces

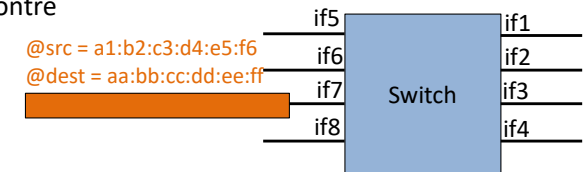
Adresse MAC	Interface / Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4

- Utilisation de la table de commutation :
  - Lorsque le switch reçoit une trame, il lit l'adresse MAC destination
  - Si l'adresse lue est présente dans la table de commutation, la trame est envoyée sur l'interface correspondante
  - Si l'adresse lue est inconnue, la trame est envoyée sur toutes les interfaces du switch sauf celle sur laquelle elle a été reçue
  - Si l'adresse lue est l'adresse de broadcast, la trame est envoyée sur toutes les interfaces du switch sauf celle sur laquelle elle a été reçue

20

## Table de commutation

- On considère le switch ci-contre



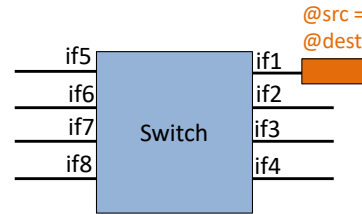
- Il reçoit une trame dont l'adresse destination est aa:bb:cc:dd:ee:ff

Adresse MAC	Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4
a1:b2:c3:d4:e5:f6	if7

21

## Table de commutation

- On considère le switch ci-contre



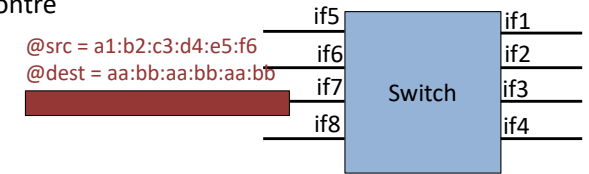
- Il reçoit une trame dont l'adresse destination est aa:bb:cc:dd:ee:ff
- L'adresse existe dans la table de commutation
- Il envoie la trame sur son port if1

Adresse MAC	Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4
a1:b2:c3:d4:e5:f6	if7

22

## Table de commutation

- On considère le switch ci-contre



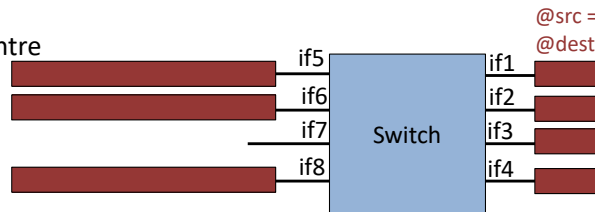
- Il reçoit une trame dont l'adresse destination est aa:bb:aa:bb:aa:bb

Adresse MAC	Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4
a1:b2:c3:d4:e5:f6	if7

23

## Table de commutation

- On considère le switch ci-contre



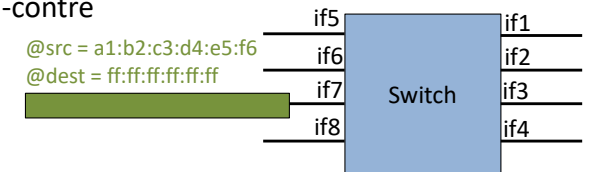
- Il reçoit une trame dont l'adresse destination est aa:bb:aa:bb:aa:bb
- L'adresse n'existe pas dans la table de commutation
- Il envoie la trame sur tous ses ports sauf if7 d'où la trame a été reçue

Adresse MAC	Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4
a1:b2:c3:d4:e5:f6	if7

24

## Table de commutation

- On considère le switch ci-contre



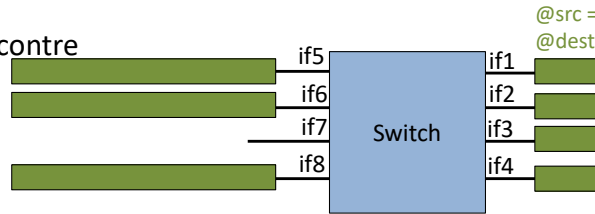
- Il reçoit une trame dont l'adresse destination est l'adresse de broadcast ff:ff:ff:ff:ff:ff

Adresse MAC	Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4
a1:b2:c3:d4:e5:f6	if7

25

## Table de commutation

- On considère le switch ci-contre



- Il reçoit une trame dont l'adresse destination est l'adresse de broadcast ff:ff:ff:ff:ff:ff
- Il envoie la trame sur tous ses ports sauf if7 d'où la trame a été reçue

Adresse MAC	Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4
a1:b2:c3:d4:e5:f6	if7

26

## Table de commutation

- La table de commutation d'un switch est le tableau dans lequel les adresses MAC des équipements en lien direct avec celui-ci sont associées à ses interfaces

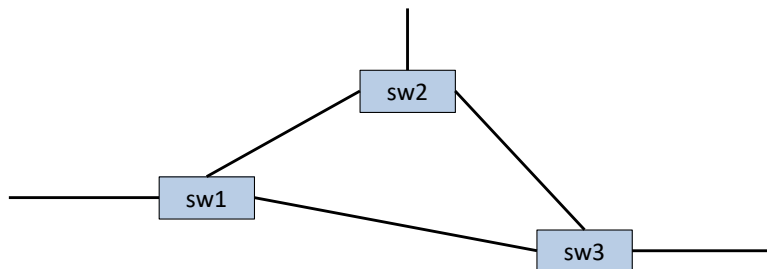
Adresse MAC	Interface / Port
aa:bb:cc:dd:ee:ff	if1
01:02:03:04:05:06	if2
aa:01:bb:02:cc:03	if3
dd:04:ee:05:ff:06	if4

- Remarques :
  - Il est possible que plusieurs adresses correspondent à un même port
    - C'est le cas par exemple si un port est WiFi
    - Ou si un port est relié à un autre switch, lui-même connecté à d'autres stations
  - Il n'est pas possible qu'une adresse corresponde à plusieurs ports
    - Si une adresse déjà connue se manifeste sur un autre port en envoyant une trame, la ligne correspondant à l'ancien port est effacée et est remplacée par la ligne correspondant au nouveau port

27

## Broadcast storm

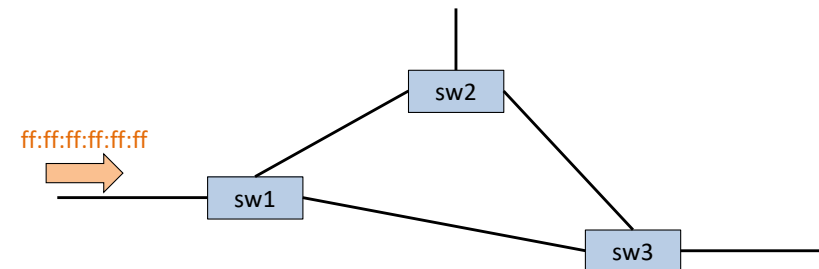
- S'il existe une boucle dans le réseau de switches



28

## Broadcast storm

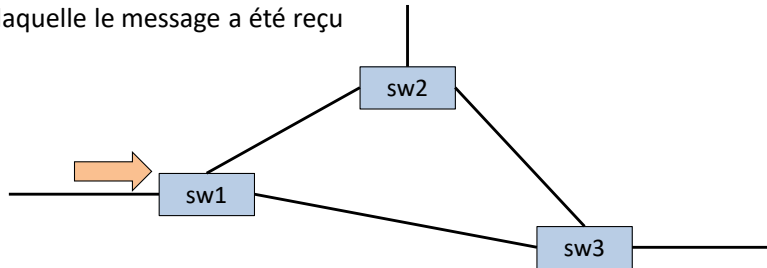
- S'il existe une boucle dans le réseau de switches
- Et un message de broadcast arrive



29

## Broadcast storm

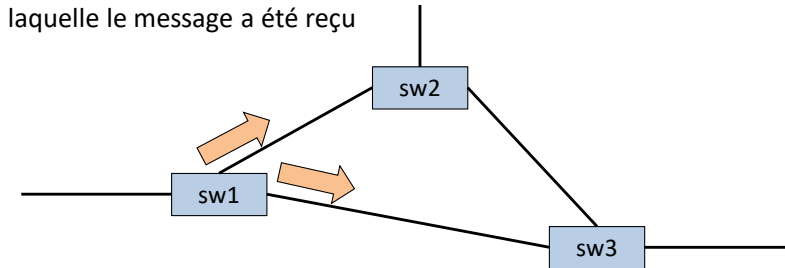
- S'il existe une boucle dans le réseau de switches
- Et un message de broadcast arrive
- Chaque switch répète le message sur toutes ses interfaces sauf celle par laquelle le message a été reçu



30

## Broadcast storm

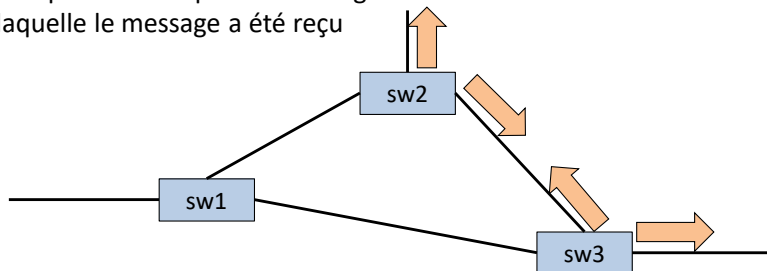
- S'il existe une boucle dans le réseau de switches
- Et un message de broadcast arrive
- Chaque switch répète le message sur toutes ses interfaces sauf celle par laquelle le message a été reçu



31

## Broadcast storm

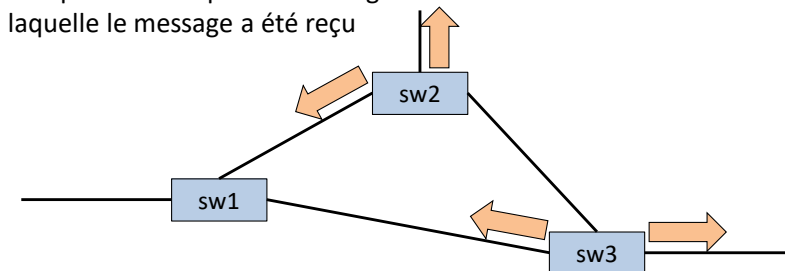
- S'il existe une boucle dans le réseau de switches
- Et un message de broadcast arrive
- Chaque switch répète le message sur toutes ses interfaces sauf celle par laquelle le message a été reçu



32

## Broadcast storm

- S'il existe une boucle dans le réseau de switches
- Et un message de broadcast arrive
- Chaque switch répète le message sur toutes ses interfaces sauf celle par laquelle le message a été reçu

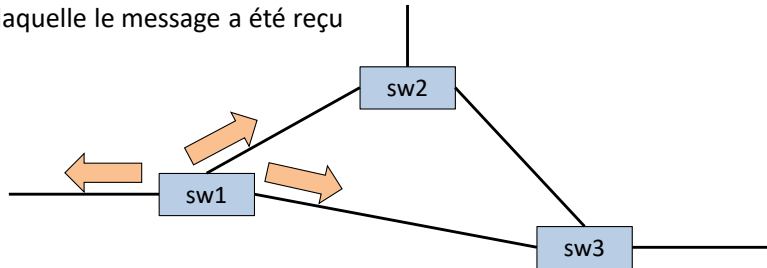


33



## Broadcast storm

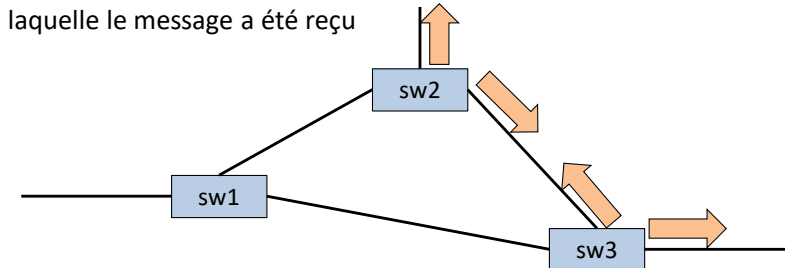
- S'il existe une boucle dans le réseau de switches
- Et un message de broadcast arrive
- Chaque switch répète le message sur toutes ses interfaces sauf celle par laquelle le message a été reçu



34

## Broadcast storm

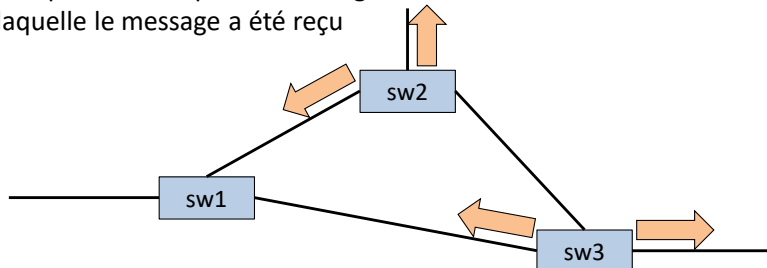
- S'il existe une boucle dans le réseau de switches
- Et un message de broadcast arrive
- Chaque switch répète le message sur toutes ses interfaces sauf celle par laquelle le message a été reçu



35

## Broadcast storm

- S'il existe une boucle dans le réseau de switches
- Et un message de broadcast arrive
- Chaque switch répète le message sur toutes ses interfaces sauf celle par laquelle le message a été reçu



- Le message tourne en boucle et cela ne s'arrête jamais
- Il faut une topologie sans boucle

36

## Spanning Tree Protocol

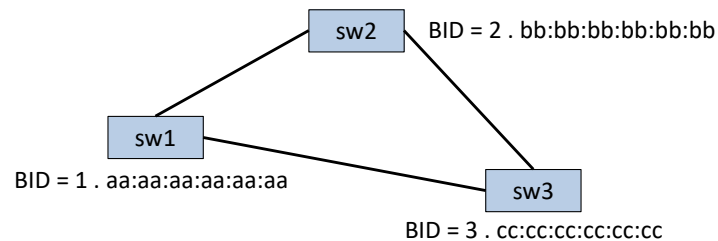
- Dans les réseaux, les boucles sont communes pour la redondance
- Et inévitables en général
- Il faut donc un moyen de gérer la présence de boucle
- De manière dynamique
- Principe :
  - Certains liens sont désactivés, c'est-à-dire qu'ils ne sont pas utilisés même s'ils existent
  - L'ensemble des liens actifs est sans boucle
  - Déterminer l'ensemble des liens actifs revient à trouver un arbre couvrant dans un graphe
- Le Spanning Tree Protocol (STP) est un protocole qui permet de construire un tel arbre de manière
  - Dynamique (s'adapte aux modifications de configuration)
  - Décentralisée (chaque switch l'implémente)

37

# Spanning Tree Protocol

- Étapes
  - Choix de la racine « root bridge » (1 pour tout le réseau local)
  - Choix des « root ports » (1 pour chaque switch)
  - Choix des « Designated ports » (1 pour chaque lien)

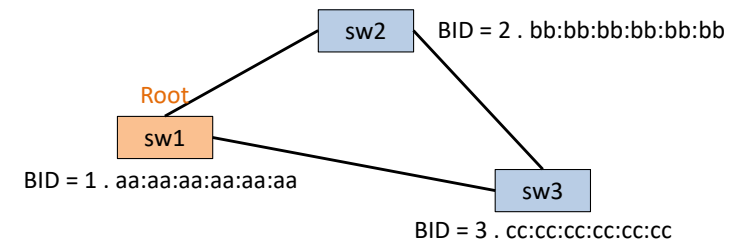
- Chaque switch a un BID
  - Bridge Identifier
  - Un bridge/pont est un switch avec 2 ports
  - Composé de
    - Priorité (2 octets) généralement en décimal
    - Adresse MAC (6 octets)



38

# Root bridge

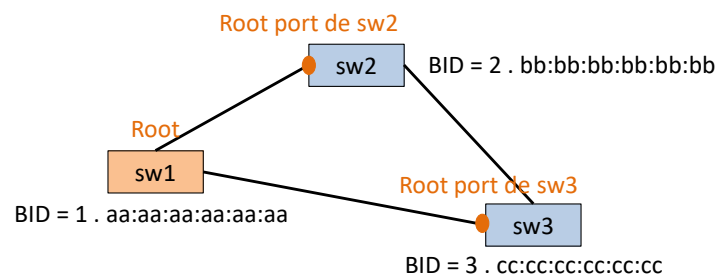
- La racine ou root bridge est le plus petit BID
  - La priorité la plus petite
  - A priorités égales, l'adresse MAC la plus petite
    - Cela revient à un choix aléatoire
- Root war
  - Au démarrage, chaque switch propage son BID
  - Quand l'algorithme converge, une seule racine est élue



39

# Root ports

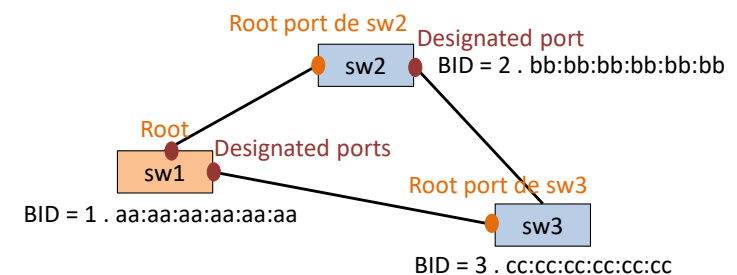
- Chaque switch calcule son plus court chemin vers la racine
- Le coût d'un lien dépend de son débit (spécification IEEE)
- Le coût d'un chemin est la somme des coûts des liens qui le composent
- Le root port d'un switch est son port le plus proche de la racine
- Chaque switch a un root port, excepté la racine



40

# Designated ports

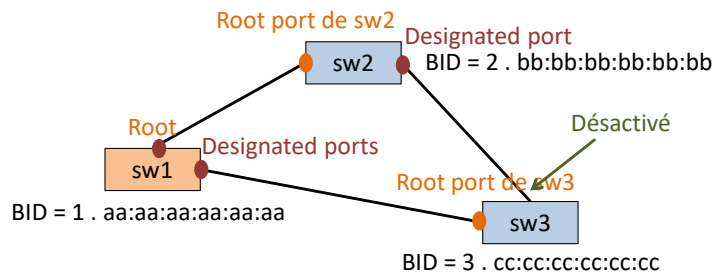
- Chaque lien a un designated port
- Le designated port d'un lien est celui qui est le plus proche de la racine
- C'est le port utilisé pour envoyer et recevoir du trafic depuis et vers la racine
- Le switch qui contient le designated port est le designated switch pour ce lien
- Un port qui n'est pas ni root port, ni designated port est désactivé



41

# Designated ports

- Chaque lien a un designated port
- Le designated port d'un lien est celui qui est le plus proche de la racine
- C'est le port utilisé pour envoyer et recevoir du trafic depuis et vers la racine
- Le switch qui contient le designated port est le designated switch pour ce lien
- Un port qui n'est pas ni root port, ni designated port est désactivé



42

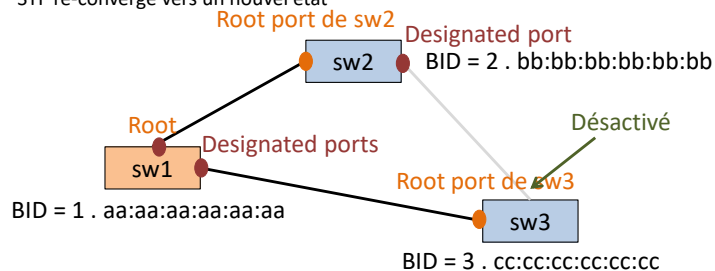
# Convergence

- Chaque switch envoie périodiquement
  - Messages BPDU (Bridge Protocol Data Unit)
  - Contenant
    - Son propre identifiant BID
    - L'identifiant BID de sa racine
    - Sa distance à sa racine
- Pour choisir le root switch puis les root ports et designated ports, on applique les mêmes critères:
  1. Lowest BID (=> racine)
  2. Lowest path cost (=> root et designated ports)
  3. Lowest sender BID (en cas d'égalité de 2.)
  4. Lowest port ID (en cas d'égalité de 2. et 3.)

43

# Fonctionnement

- Une trame reçue sur un root port est retransmise sur les designated ports (trafic venant de la racine)
- Une trame reçue sur un designated port est retransmise sur le root port (trafic vers la racine)
- Un port ni root ni designated est désactivé et ne peut retransmettre de trames
- Les liens dont un port est désactivé (en émission et en réception) ne sont plus utilisés
- Même si les autres liens ont trop de trafic, le lien désactivé reste inutilisé
- En cas de perte d'un lien
  - Les messages BPDU propagent les nouvelles valeurs
  - STP re-converge vers un nouvel état



44

# WiFi

45



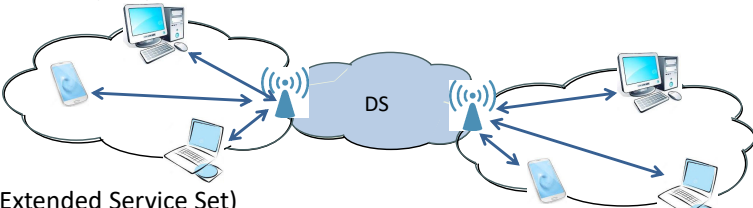
# Le standard WiFi

- Le WiFi correspond à la norme 802.11
- Rôles :
  - Couche physique
    - Emission et réception radio
    - Plusieurs versions
  - Couche liaison de données MAC
    - Accès au canal radio partagé
    - Formation de trames
- Historique :
  - 1997 : 802.11 première version du WiFi (bande 2,4GHz)
  - 1999 : 802.11a « WiFi 2 » haut-débit avec l'introduction de l'OFDM (bande 5GHz)
  - 1999 : 802.11b « WiFi 1 » bande 2,4 GHz
  - 2003 : 802.11g « WiFi 3 »
  - 2004 : 802.11i remplacement de WEP par WPA pour le chiffrement
  - 2008 : 802.11r handover entre les points d'accès
  - 2009 : 802.11n « WiFi 4 » ajout du MIMO et de l'agrégation de porteuses
  - 2014 : 802.11ac « WiFi 5 »
  - 2021 : 802.11ax « WiFi 6 »



46

# Architecture

- 
- 
- BSS (Basic Service Set)
    - Un point d'accès (AP)
    - Toutes les communications passent par l'AP
    - La bande passante est partagée entre toutes les stations
  - IBSS (Independent Basic Service Set)
    - Pas de point d'accès
    - Réseau Ad-Hoc
- 
- ESS (Extended Service Set)
    - Plusieurs BSS reliés par un système de distribution (DS)

47

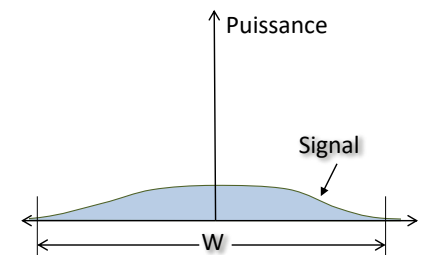
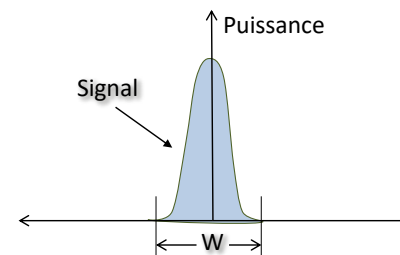
# Couche physique

- Plusieurs couches PHY
  - DSSS (Direct Sequence Spread Spectrum)
  - OFDM (Orthogonal Frequency Division Multiplexing)
  - 2 versions présentes uniquement dans la norme d'origine :
    - FHSS (Frequency Hopping Spread Spectrum)
    - IR (InfraRed)
- 2 bandes de fréquences
  - Parmi les bandes libres
  - Bande 2,4 GHz
    - Entre 2,40 et 2,472 GHz en Europe
    - Division en 13 canaux de 22 MHz espacés de 5 MHz (recouvrement entre canaux)
    - Cohabitation avec le micro-onde, le bluetooth...
  - Bande 5 GHz
    - Entre 5,15 et 5,35 GHz et entre 5,47 et 5,725 GHz
    - Division en 19 canaux de 20 MHz ne se recouvrant pas
    - Possibilité d'agréger des canaux adjacents
    - Cohabitation avec les radars météo, usages militaires...

48

# Étalement de spectre DSSS

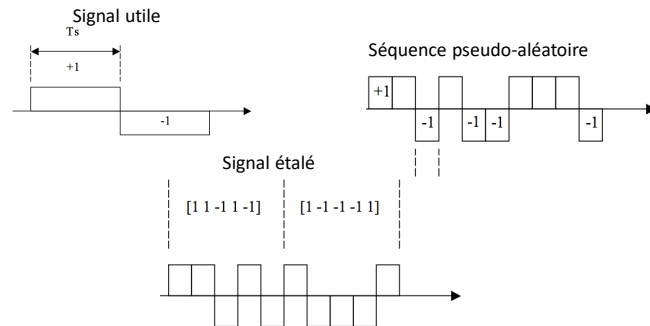
- Principe
  - La formule de Shannon lie le débit C à la largeur de bande de fréquence W utilisée :
 
$$C = W \log_2(1 + SINR)$$
  - Le SINR est le rapport Signal utile sur Interférences et bruit (Noise)
  - Une bande W plus large permet de compenser un mauvais SINR
  - Ou de limiter la puissance d'émission



49

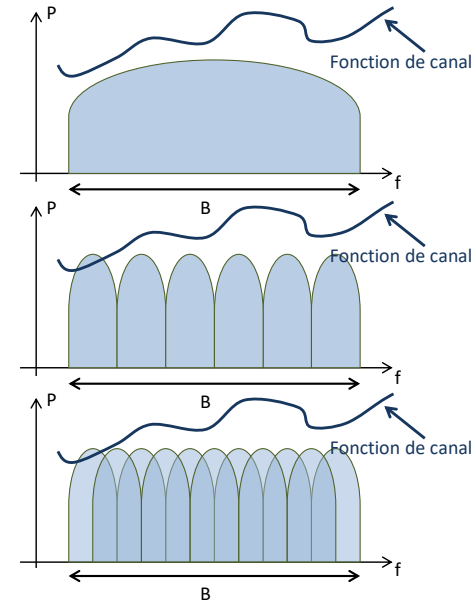
# Étalement de spectre DSSS

- Fonctionnement
  - On combine le signal utile avec une séquence pseudo-aléatoire (séquence de Barker) de période plus faible et donc de bande plus large
  - Les équipements d'un même BSS utilisent tous la même séquence



50

# OFDM



- Liaison monoporteuse
  - Largeur de bande B
  - Débit D
  - La réponse du canal varie sur la bande B
- Liaison multiporteuse
  - N porteuses
  - Largeur de bande par porteuse B/N
  - Débit D
  - La réponse du canal ne varie pas sur la bande B/N
- Liaison OFDM
  - 2N porteuses
  - Formes d'ondes orthogonales => pas d'interférences malgré les chevauchements
  - Largeur de bande par porteuse B/N
  - Débit 2D
  - La réponse du canal ne varie pas sur la bande B/N

51

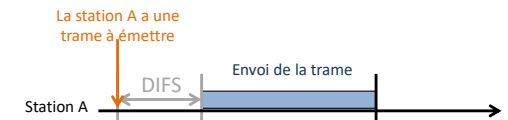
# Couche MAC

- Gestion de l'accès multiple sur le canal radio partagé par l'ensemble des utilisateurs du Service Set
- 3 techniques :
  - DCF** (Distributed Coordination Function)
    - Toutes les stations implémentent la même technique
    - CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance)
    - RTS-CTS optionnel
  - PCF** (Point Coordination Function)
    - La gestion du partage est centralisée au point d'accès
  - HCF** (Hybrid Coordination Function)
    - Période dédiée où le canal est réservé pour le trafic QoS
    - Le reste du temps en mode distribué

52

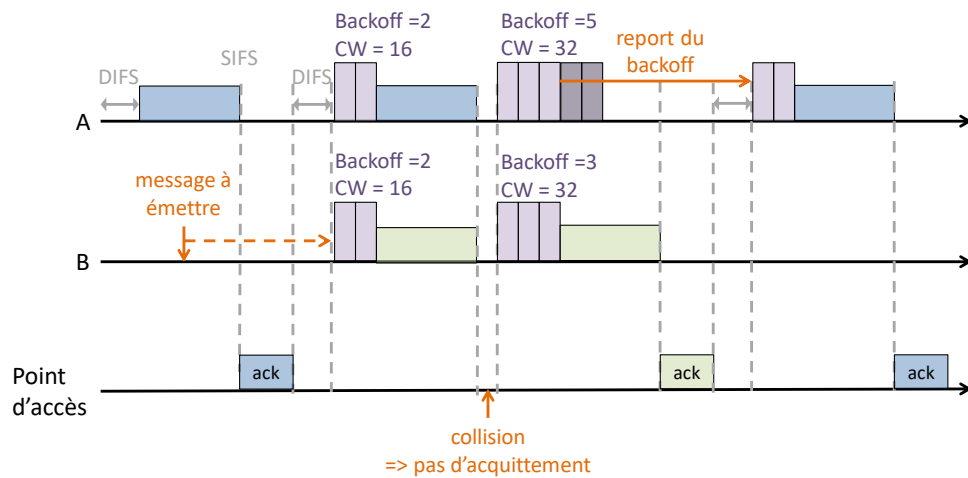
# CSMA/CA

- La norme DCF utilise le mécanisme CSMA/CA :
- Délai d'attente
  - Délai fixe d'écoute obligatoire avant toute émission (CarrierSense)
  - DIFS (DCF Inter-Frame Space)
- Backoff et fenêtre de contention
  - Quand le canal est occupé, il y a un délai d'attente aléatoire (backoff) lors de sa libération à la suite du DIFS
  - Une station qui souhaite émettre tire un backoff dans une fenêtre de contention (Contention Window)
  - La taille de la fenêtre de contention suit la règle du BEB (Binary Exponential Backoff) et initialement de 16
  - Si la contention est perdue, la valeur du backoff restant est reportée au prochain essai
- Acquittements et retransmissions
  - Toutes les trames (sauf broadcast) sont acquittées
  - Délai entre une trame et son acquittement : SIFS (Short IFS)
  - En l'absence d'acquiescement :
    - Retransmission de la trame
    - La taille de la fenêtre de contention double (règle BEB)
    - Jusqu'à atteindre 1024 où en cas d'échec, la trame est alors supprimée



53

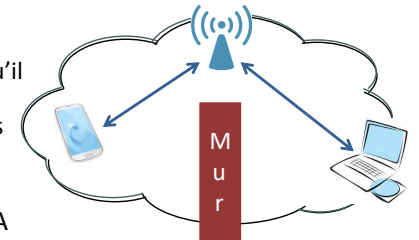
# CSMA/CA



54

# Problème du terminal caché

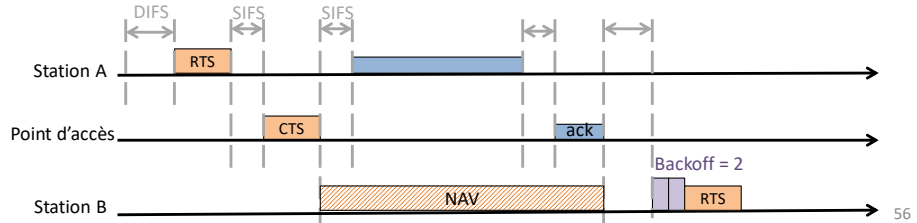
- Si 2 stations ne se « voient » pas réciproquement
  - L'une ne sait pas quand l'autre émet
  - Et peut penser que le canal est libre alors qu'il est occupé
  - Collision possible au niveau du point d'accès
- L'écoute et l'attente ne suffisent pas pour éviter les collisions
- Il faut compléter le mécanisme de CSMA/CA
- Si une station veut absolument éviter une retransmission
  - Car une trame est longue
  - Et coûteuse à réémettre
- Il faut un moyen d'annuler le risque de collision



55

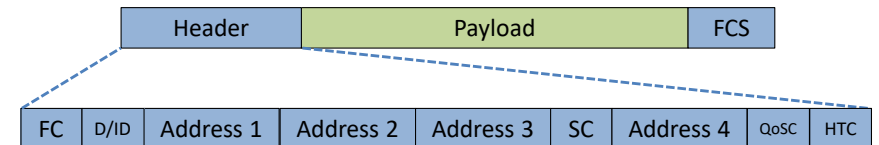
# RTS-CTS

- La norme DCF propose le mécanisme RTS-CTS (optionnel)
- Principe
  - Une station réserve le canal au préalable pour envoyer une trame
  - Messages
    - RTS (Request To Send)
      - La station demande au point d'accès l'autorisation d'envoyer
      - Contient le temps estimé de la transmission
    - CTS (Clear To Send)
      - Le point d'accès autorise l'envoi pendant le temps estimé
      - Et prévient ainsi toutes les stations du Service Set de la réservation du canal pendant une durée donnée
  - Les autres stations se mettent en mode « NAV » pendant le temps où le canal est réservé
    - NAV : Network Allocation Vector
    - Virtual carrier sensing (lorsque le physical carrier sensing n'est pas possible)



56

# Trame WiFi



- Header
  - FC (Frame Control) : version du protocole, type de trame (association/beacon, signalisation, données), ...
  - D/ID (Duration ID) : durée de la transmission
  - Champs adresses
    - Sur 6 octets (adresse MAC)
    - Address 1 : destination
    - Address 2 : source
    - Address 3 : point d'accès récepteur
    - Address 4 : point d'accès émetteur (pour les ESS)
    - SC (Sequence Control) : utilisé pour la fragmentation
  - QoSC (QoS Control) et HTC (High Throughput Control) : optionnels
- Payload :
  - Contenu de la trame issu des couches supérieures (IP)
- FCS (Frame Check Sequence) :
  - Bits de redondance pour la détection d'erreur
  - Checksum basé sur CRC

57

# Association à un réseau WiFi

- SSID
  - Service Set Identifier
  - « Nom du réseau »
  - Le point d'accès et les stations doivent avoir le même SSID pour s'associer
  - Diffusé en clair périodiquement sur la voie balise (beacon)
    - Possibilité de désactiver l'émission périodique
  - Diffusé en clair dans la trame d'association
- Étapes
  - Écoute (« scanning »)
    - Connaître le canal du point d'accès
    - Synchronisation fréquentielle puis temporelle
  - Association
    - Requête d'association
    - Réponse par une trame « probe » contenant le SSID
  - Allocation d'une adresse IP
  - Authentification optionnelle

58

# Sécurité

- WEP (Wired Equivalent Policy)
  - Authentification avec clé unique
  - Chiffrement par flot à partir de la clé unique et d'un vecteur d'initialisation envoyé en clair
  - Protection d'intégrité
  - Optionnel
  - Une écoute clandestine permet de retrouver la clé en quelques secondes
- WPA puis WPA2 (WiFi Protected Access)
  - WPA est une solution intermédiaire pour remplacer WEP en attendant WPA2
  - Mode personnel et mode entreprise
  - Authentification par clé unique ou par utilisateur suivant le mode
  - Chiffrement par flot à partir d'une clé périodiquement modifiée et d'un vecteur d'initialisation haché, ou par bloc
  - Protection d'intégrité
  - Obligatoire depuis 2006

59

RES101

Internet

## 2. ARCHITECTURE

60

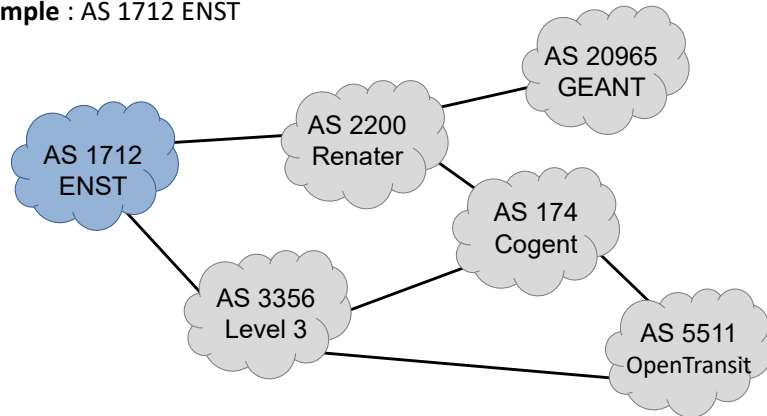
# Internet

- Internet : Interconnexion des réseaux (networks)
- Réseau à commutation de paquets composé de millions de réseaux publics ou privés
- Gestion d'Internet
  - IETF (Internet Engineering Task Force)
    - Aspects architecturaux et techniques
    - Élaboration des standards RFC (ex : IP)
  - ICANN (Internet Corporation for Assigned Names and Numbers)
    - Attribution des noms de domaines
    - Attribution des adresses IP (département IANA (Internet Assigned Numbers Authority))
  - ISOC (Internet Society)
    - Support organisationnel et financier de l'IETF

61

# Architecture d'Internet

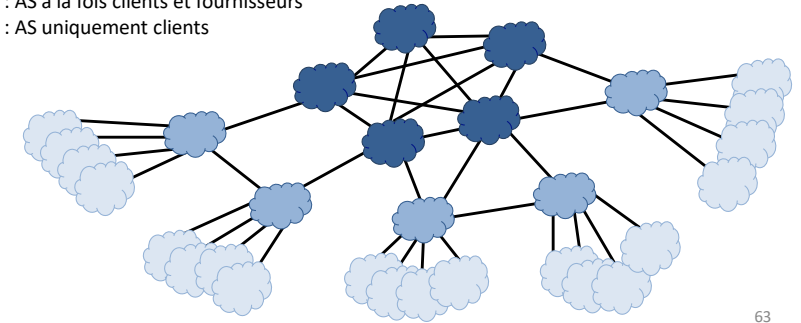
- Internet est composé d'AS (Autonomous Systems)
- Chaque AS est géré par une unique entité administrative indépendamment des autres
- Il y a plus de 64 000 AS (2019)
- **Exemple** : AS 1712 ENST



62

# Autonomous Systems

- Relations entre AS
  - Relation client/fournisseur
    - Utilisation du lien payant pour le client
  - Relation pair à pair
    - Utilisation du lien gratuite si trafic bilatéral
    - Uniquement pour le trafic de ou vers ses propres clients
- Trois types d'AS
  - Tier 1 : AS uniquement fournisseurs reliés par un réseau totalement maillé
  - Tier 2 : AS à la fois clients et fournisseurs
  - Tier 3 : AS uniquement clients



63

# Adresse IP

- Une adresse IP permet d'identifier et de localiser une machine de manière unique connectée à Internet
- L'IANA est composé de 5 registres régionaux :
  - RIPE NCC (Réseaux IP Européens Network Coordination Center) : Europe, Russie, Moyen-Orient, Groenland
  - ARIN (American Registry for Internet Numbers) : Etats-Unis, Canada
  - APNIC (Asia Pacific Network Information Center) : Asie du Sud-Est, Australie, Nouvelle-Zélande
  - AfriNIC (African Network Information Center) : Afrique
  - LACNIC (Latin American and Caribbean Network Information Center) : Amérique latine, Caraïbes
- L'IANA possède toutes les adresses IP et les répartit entre les 5 registres
- Chaque registre possède un pool d'adresses IP à attribuer ou vendre à la demande
- Un AS possède des adresses IP achetées auprès de son registre régional et gère l'attribution de celles-ci à ses machines
- **Exemple** : l'AS 1712 ENST gère toutes les adresses IP 137.194.x.x

64

# Adresse IPv4

- Une adresse IPv4 est
  - Sur 32 bits
  - Écrite comme 4 nombres décimaux compris entre 0 et 255 séparés par des points
  - **Exemple** : 137.194.200.21
- L'adressage IP est hiérarchique
  - Les premiers bits identifient le sous-réseau d'une machine : on l'appelle le préfixe
  - Les derniers bits identifient la machine à l'intérieur du sous-réseau
  - **Exemple** : 137.194.x.x est le sous-réseau de Télécom Paris (AS1712), et 137.194.200.21 est une machine sur ce sous-réseau
- Une adresse IP est donc l'adresse associée à la longueur du préfixe
  - On donne la longueur en nombre de bits
  - On l'écrit à la suite de l'adresse après un « / »
  - **Exemple** : pour 137.194.200.21, le préfixe est de 2 octets soit 16 bits donc on écrit 137.194.200.21/16

65



# Classfull addressing

- **Principe** : La longueur du préfixe est donnée implicitement par la valeur de l'adresse
- Les adresses sont rangées dans 5 classes :
  - Classe A : adresses de 0.0.0.0 à 127.255.255.255
    - Préfixe de longueur 8 (premier nombre décimal)
    - 1<sup>er</sup> bit = 0
    - Environ 16 millions d'adresses disponibles par sous-réseau
  - Classe B : adresses de 128.0.0.0 à 191.255.255.255
    - Préfixe de longueur 16 (2 premiers nombres décimaux)
    - 2 premiers bits = 10
    - Environ 65000 adresses disponibles par sous-réseau
    - Exemple : 137.194.200.21/16
  - Classe C : adresses de 192.0.0.0 à 223.255.255.255
    - Préfixe de longueur 24 (3 premiers nombres décimaux)
    - 3 premiers bits = 110
    - 256 adresses disponibles par sous-réseau
  - Classe D : adresses de 224.0.0.0 à 239.255.255.255
    - Préfixe de longueur 4
    - 4 premiers bits = 1110
    - Adresses réservées pour le multicast
  - Classe E : adresses de 240.0.0.0 à 255.255.255.255
    - Préfixe de longueur 4
    - 4 premiers bits = 1111
    - Adresses réservées par l'IANA
- **Limitations** : peu flexible, gaspillage d'adresses.

66

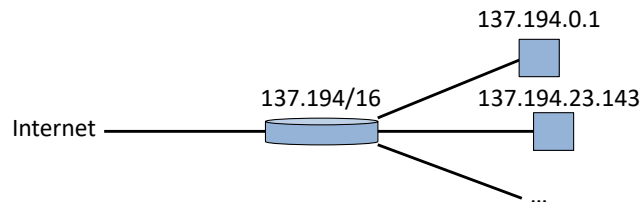
# CIDR : Classless Inter Domain Routing

- **Principe** : On précise la longueur du préfixe avec un « / » après chaque adresse. Cela permet des préfixes de n'importe quelle longueur, et de s'adapter à plusieurs tailles de sous-réseaux.
- Les adresses sont sur  $4 \times 8$  bits = 32 bits, on peut faire des préfixes de taille 1 à 31
- On transcrit les adresses sous forme binaire :
  - X en décimal = abcdefgh en binaire
  - Avec a, b, c, d, e, f, g, et h = 0 ou 1
  - Si  $X = 128*a + 64*b + 32*c + 16*d + 8*e + 4*f + 2*g + 1*h$
- Exemple :
  - 137.194.23.143/20 =  $\underbrace{10001001.11000010.00010111.10001111}_{\text{sous-réseau}}, \underbrace{10001111}_{\text{machine}}$

67

## Sous-réseau

- Ensemble de machines pouvant communiquer directement
- Pour sortir du sous-réseau il faut passer par un routeur



- Deux adresses sont réservées (non assignables à des machines) :
  - L'adresse de sous-réseau (la plus petite disponible) : préfixe + bits restants à 0
  - L'adresse de broadcast (la plus grande disponible) : préfixe + bits restants à 1
  - Exemple : pour 137.194/16
    - 137.194.0.0 est l'adresse du sous-réseau
    - 137.194.255.255 est l'adresse de broadcast sur ce sous-réseau

68

## Masque de sous-réseau

- La longueur du préfixe est donnée en nombre de bits N
- Le masque de sous-réseau correspond à une adresse IP avec N premiers bits à 1 et les 32-N suivants à 0
  - Exemples :
    - /16 = 11111111.11111111.00000000.00000000 = 255.255.0.0
    - /20 = 11111111.11111111.11110000.00000000 = 255.255.240.0
- Adresse de sous réseau = adresse IP (binaire) ET masque (binaire)
  - Exemple :
    - 137.194.23.143 = 10001001.11000010.00010111.10001111
    - Sous réseau de 137.194.23.143/20 :
      - 10001001.11000010.00010111.10001111
      - 11111111.11111111.11110000.00000000
      - = 10001001.11000010.00010000.00000000
      - = 137.194.16.0
- Dans une machine, les masques sont souvent rentrés sous cette forme là

69

# Adresses réservées

- Quand on n'a pas d'adresse IP :
  - 0.0.0.0
- Adresses privées (non routables) :
  - 10/8
  - 172.16/12
  - 192.168/16
- Loopback (joindre sa propre machine) :
  - 127/8
- Documentation (exemples de code) :
  - 198.51.100/24
  - 203.0.113/24
- Multicast :
  - 224/4
- Réservées :
  - 240/4

70

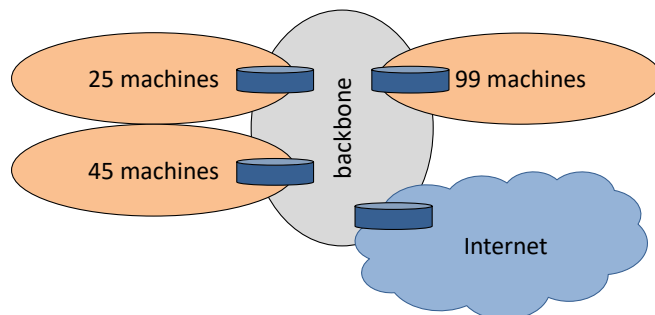
# Plan d'adressage

- Faire un plan d'adressage c'est attribuer une adresse IP à chaque machine :
  - L'administrateur réseau dispose d'une plage d'adresses
  - Il découpe sa plage d'adresses en fonction :
    - Du nombre de sous-réseaux présents (dépend des branchements des routeurs et switches)
    - Du nombre de stations connectées à chaque sous-réseau
  - Les sous-réseaux ne sont pas forcément de même taille (longueurs de masque différentes)
- Attention, il ne faut pas oublier :
  - Les 2 adresses réservées dans chaque sous-réseau
    - /24 contient  $2^8=256$  adresses, c'est donc un sous-réseau qui peut avoir 254 machines
    - /30 contient  $2^2=4$  adresses donc 2 machines
  - D'attribuer des adresses aux routeurs
    - Traditionnellement, pour séparer les stations des routeurs, on attribue dans l'ordre croissant à partir de l'adresse de réseau les adresses aux stations, et dans l'ordre décroissant à partir de l'adresse de broadcast les adresses aux routeurs
    - C'est une pratique et non une obligation
    - Exemple : dans le sous-réseau 137.194.200/24
      - 137.194.200.0 est l'adresse de sous-réseau
      - 137.194.200.1, 137.194.200.2, 137.194.200.3... sont des stations
      - 137.194.200.253, 137.194.200.254 sont des routeurs
      - 137.194.200.255 est l'adresse de broadcast
  - De prévoir un sous-réseau pour relier les sous-réseaux si ceux-ci sont connectés à des routeurs différents (backbone)

71

# Plan d'adressage

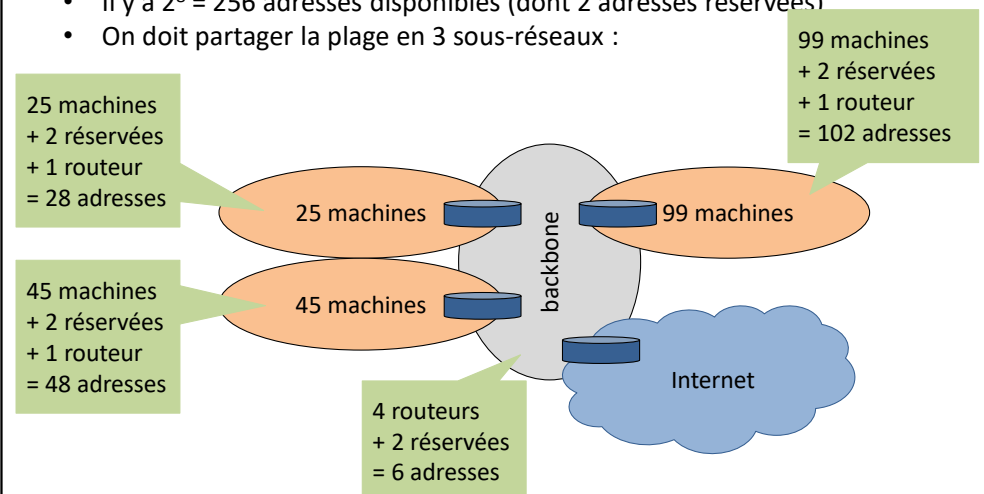
- On considère la plage d'adresses 193.215.124/24
- Il y a  $2^8 = 256$  adresses disponibles (dont 2 adresses réservées)
- On doit partager la plage en 3 sous-réseaux :



72

# Plan d'adressage

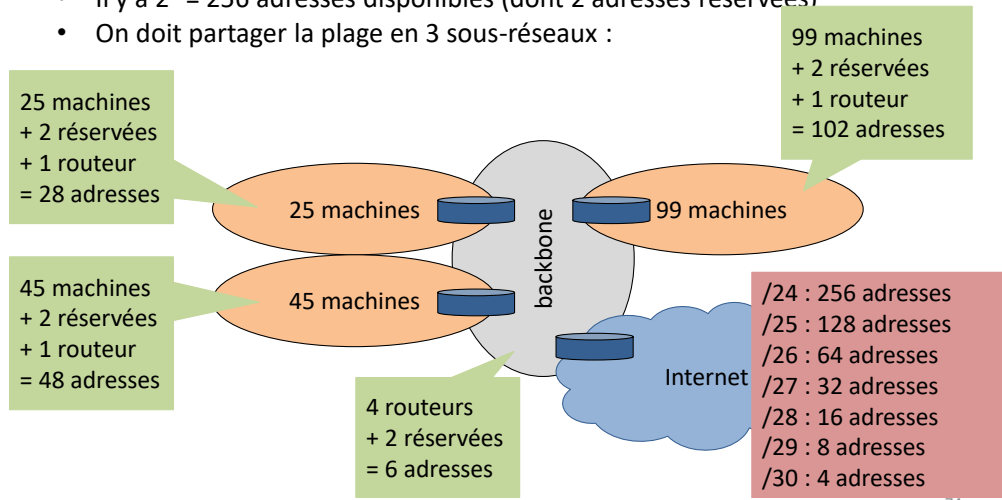
- On considère la plage d'adresses 193.215.124/24
- Il y a  $2^8 = 256$  adresses disponibles (dont 2 adresses réservées)
- On doit partager la plage en 3 sous-réseaux :



73

## Plan d'adressage

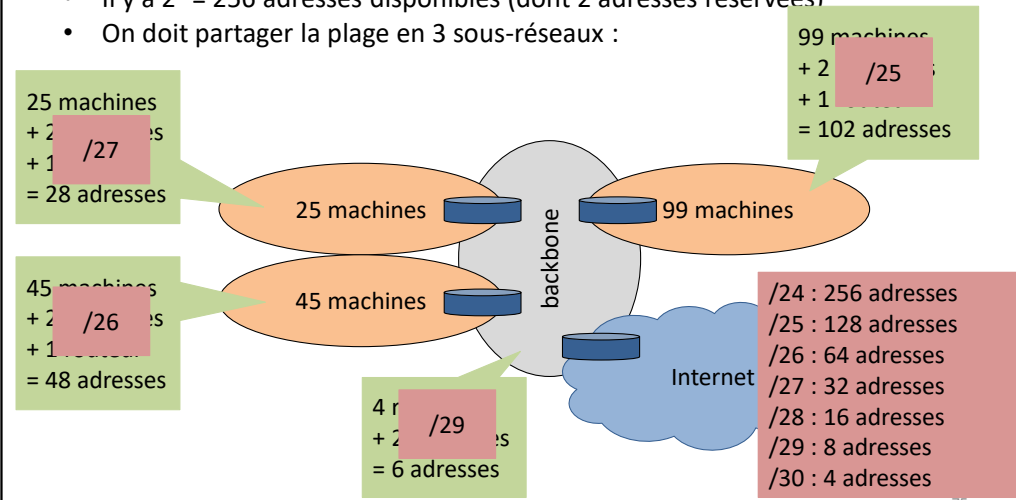
- On considère la plage d'adresses 193.215.124/24
- Il y a  $2^8 = 256$  adresses disponibles (dont 2 adresses réservées)
- On doit partager la plage en 3 sous-réseaux :



74

## Plan d'adressage

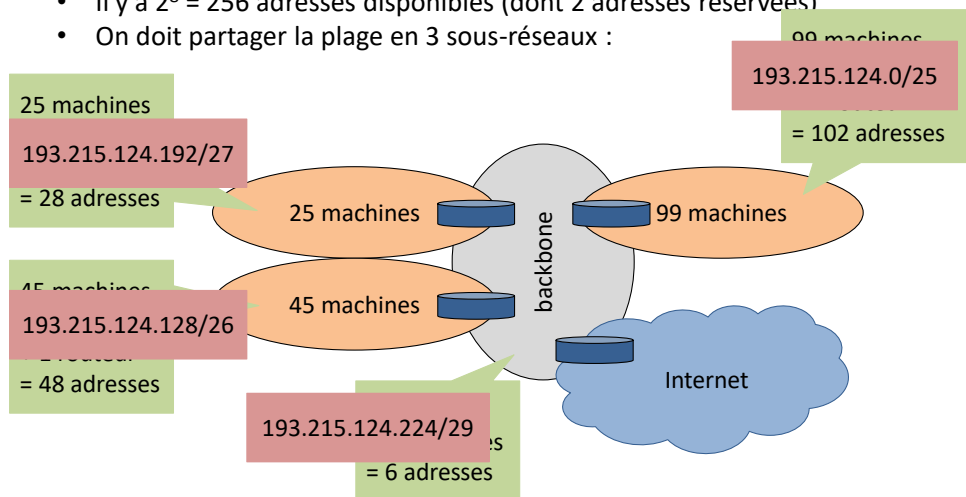
- On considère la plage d'adresses 193.215.124/24
- Il y a  $2^8 = 256$  adresses disponibles (dont 2 adresses réservées)
- On doit partager la plage en 3 sous-réseaux :



75

## Plan d'adressage

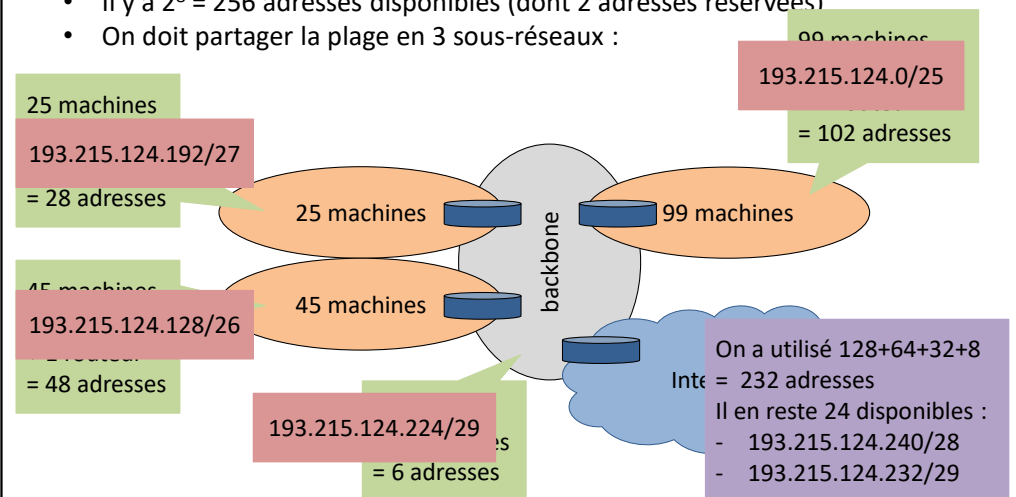
- On considère la plage d'adresses 193.215.124/24
- Il y a  $2^8 = 256$  adresses disponibles (dont 2 adresses réservées)
- On doit partager la plage en 3 sous-réseaux :



76

## Plan d'adressage

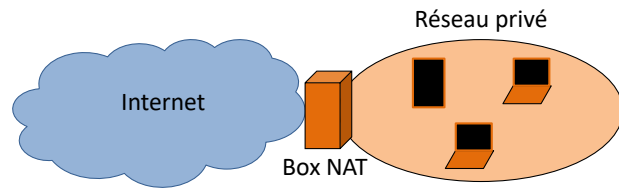
- On considère la plage d'adresses 193.215.124/24
- Il y a  $2^8 = 256$  adresses disponibles (dont 2 adresses réservées)
- On doit partager la plage en 3 sous-réseaux :



77

# Adressage privé et NAT

- Une adresse IP est publique si elle est routable partout sur Internet
- Une adresse privée est non routable en dehors du réseau privé dans laquelle elle a été assignée
- En 2011, l'IANA n'a plus d'adresses publiques IPv4 à donner
- Le NAT (Network Address Translation) permet d'utiliser des adresses privées cachées dans un réseau privé derrière un serveur NAT pour accéder à Internet



- La box a une adresse IP publique routable
- Toutes les stations du réseau privé ont des adresses IP privées

78

# Adressage privé et NAT

- Table de correspondance :

@IP publique + port	@IP privée + port
1.1.1.1 ; 65198	192.168.1.2 ; 55673
1.1.1.1 ; 49173	192.168.1.2 ; 61690
1.1.1.1 ; 58177	192.168.1.3 ; 55673

- Emission :
  - Quand une station émet un paquet, elle remplit l'en-tête avec son adresse IP source (privée) et le numéro de port de la communication
  - Quand la box reçoit le paquet, elle remplace l'adresse IP source par sa propre adresse, et le numéro de port par le correspondant dans la table NAT
- Réception :
  - Quand la réponse arrive en sens inverse, la box remplace l'adresse destination qui contient son adresse IP et le numéro de port de communication par l'adresse privée de la station et le numéro de port correspondant dans la table NAT
  - La station reçoit le paquet avec son adresse IP privée en destination et le numéro de port de la communication

79

# Adressage privé et NAT

- Table de correspondance :

	@IP publique + port	@IP privée + port	
Adresse IP publique de la box	1.1.1.1 ; 65198	192.168.1.2 ; 55673	Une ligne par connexion
	1.1.1.1 ; 49173	192.168.1.2 ; 61690	
	1.1.1.1 ; 58177	192.168.1.3 ; 55673	

- Emission :
  - Quand une station émet un paquet, elle remplit l'en-tête avec son adresse IP source (privée) et le numéro de port de la communication
  - Quand la box reçoit le paquet, elle remplace l'adresse IP source par sa propre adresse, et le numéro de port par le correspondant dans la table NAT
- Réception :
  - Quand la réponse arrive en sens inverse, la box remplace l'adresse destination qui contient son adresse IP et le numéro de port de communication par l'adresse privée de la station et le numéro de port correspondant dans la table NAT
  - La station reçoit le paquet avec son adresse IP privée en destination et le numéro de port de la communication

80

# Adresse IPv6

- Une adresse IPv6 est
  - Sur 128 bits
  - Écrite comme 8 groupes de 4 hexas (quad) séparés par « : »
  - 2<sup>96</sup> fois plus d'adresses que IPv4
  - **Exemple** : FEDC:BA98:7654:3210:FEDC:BA98:7654:3210
- Pour simplifier la lecture on a le droit de couper les 0 dans les cas suivants :
  - 0 en tête de nombre : 0001 -> 1
  - Groupes de 0 remplacés par un seul 0 : 0000 -> 0
  - Groupes de 0 remplacés par rien : :0000 -> :0: -> :: (mais on ne peut le faire qu'une seule fois par adresse)
  - **Exemple** : 2001:0DB8:0000:00A4:0000:0000:0000:CAFE -> 2001:DB8:0:A4::CAFE
- Le préfixe de sous-réseau est toujours de taille 64
  - Plan d'adressage simplifié
  - **Exemple** : 2001:0DB8:0000:00A4:0000:0000:0000:CAFE

préfixe global    sous-réseau    machine

81

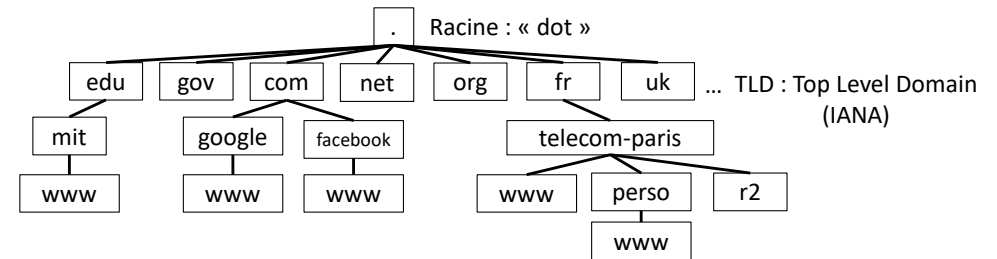
# Adresse IPv6

- 3 types d'adresses :
  - Unicast
    - 1 adresse = 1 interface
  - Multicast
    - 1 adresse = plusieurs interfaces
    - Le broadcast est juste un multicast particulier
    - Adresses en FF00::/8 (de FF00:: à FFFF::)
  - Anycast
    - 1 adresse = plusieurs interfaces
    - On envoie à une seule interface parmi le groupe (n'importe laquelle, par exemple la plus proche)
- Adresses unicast :
  - Global : 2000::/3 (2000:: à 3FFF::)
  - Local : FC00::/7
  - Link local : FE80::/10 (pour le neighbor discovery)
  - Unspecified : :: (0:0:0:0:0:0:0:0)
  - Loopback : ::1 (0:0:0:0:0:0:0:1)

82

# URL

- En pratique, les utilisateurs n'identifient pas des stations par leurs adresses IP
- Sur Internet, les stations ou serveurs sont identifiés par des noms URL (Uniform Resource Locator)
- Le format d'une URL est « protocol://serveur/fichier »
  - Exemple : <http://www.telecom-paris.fr/index.html>
- Une URL valide est un Fully Qualified Domain Name (FQDN) unique



- Règles :
  - 1 nœud a un label (caractères alphanumériques et « - », longueur ≤ 63 caractères)
  - 1 nœud ne peut pas avoir le même label qu'un parent
  - La concaténation des labels de la feuille vers la racine en séparant par des « . » doit être unique (FQDN) (longueur ≤ 255 caractères au total)

83

# DNS

- Le DNS (Domain Name System) permet de faire le lien entre une URL et une adresse IP
- Serveurs DNS :
  - 13 serveurs root numérotés de A à K dans le monde (instances répétées)
  - Serveurs DNS responsables pour chaque TLD, et pour chaque domaine
  - Serveurs DNS locaux qui ont le rôle de « resolver »
  - **Exemple** : le serveur DNS de Télécom Paris est à la fois responsable du domaine telecom-paris.fr, et « resolver » pour les requêtes issues du réseau interne
- Fonctionnement :
  - Un client contacte son serveur DNS local (resolver) avec une URL
  - Le serveur DNS local interroge le serveur DNS responsable du domaine demandé pour obtenir l'adresse IP correspondant
  - Pour retrouver un serveur DNS responsable d'un domaine
    - Les 13 serveurs root sont connus
    - Un serveur root connaît tous les serveurs TLD
    - Un serveur TLD connaît tous les serveurs pour tous ses domaines
  - Chaque serveur maintient un cache pour éviter les demandes répétitives

84

RES101

Internet

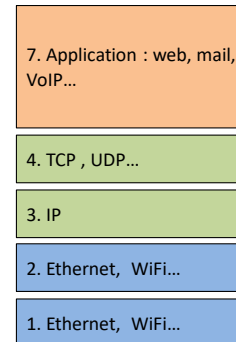
## 3. PROTOCOLES

85

## Plan usager

86

## Pile protocolaire de l'utilisateur

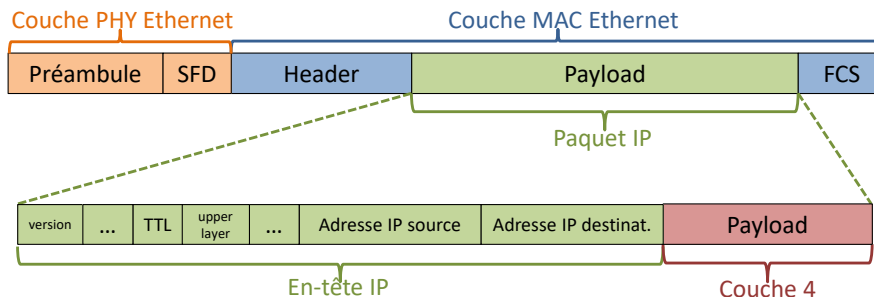


- IP n'est pas uniquement un système d'adressage
- IP = Internet Protocol
- C'est le protocole commun à toutes les communications sur Internet
- C'est un protocole de niveau 3
- Rôles :
  - Adressage global
  - Routage à travers le réseau
  - Découpage des messages venant des couches supérieures en paquets
  - Fragmentation et réassemblage des paquets
    - La taille maximale des paquets MTU (Maximum Transmission Unit) dépend de l'interface utilisé (exemple : pour Ethernet, MTU=1500 octets)
    - En arrivant sur un segment avec une taille limitée, IP fragmente et réassemble à l'issue du segment critique

87

## Paquet IP

- Le paquet IP est la charge (payload) contenue dans une trame Ethernet, ou dans une trame WiFi...

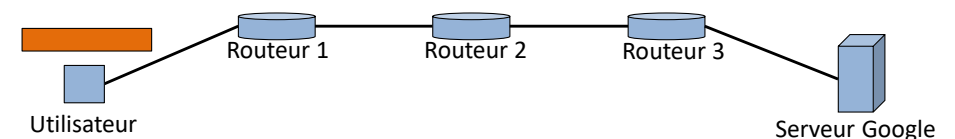


- Le paquet IP a lui aussi un en-tête spécifique IP qui comprend
  - Version du protocole (IPv4 ou IPv6)
  - TTL (Time To Live) (IPv4) ou Hop limit (IPv6) : nombre de sauts maximum du paquet (décrémenté à chaque routeur, entraîne la destruction du paquet si atteint 0)
  - Upper Layer (IPv4) ou Next Header (IPv6) : protocole de couche 4 du payload
  - Adresse IP source
  - Adresse IP destination
  - Champs IPv4 : fragmentation, types de services, options, longueur totale, header checksum
  - Champs IPv6 : QoS, load balancing, longueur du payload

88

## Adressage bout-en-bout

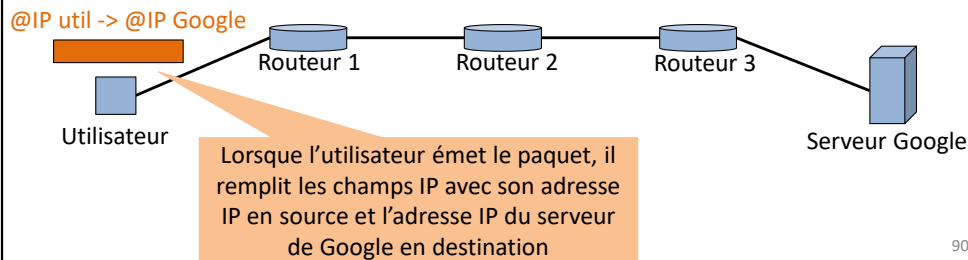
- Un paquet IP circulant sur Ethernet possède donc :
  - Une adresse IP source et une adresse IP destination dans l'en-tête du paquet IP
  - Une adresse MAC source et une adresse MAC destination dans l'en-tête de la trame Ethernet
- Les adresses IP ne sont pas modifiées pendant la durée de vie du message
  - Sauf si le paquet passe par une passerelle NAT
- Les adresses MAC changent à chaque saut IP



89

# Adressage bout-en-bout

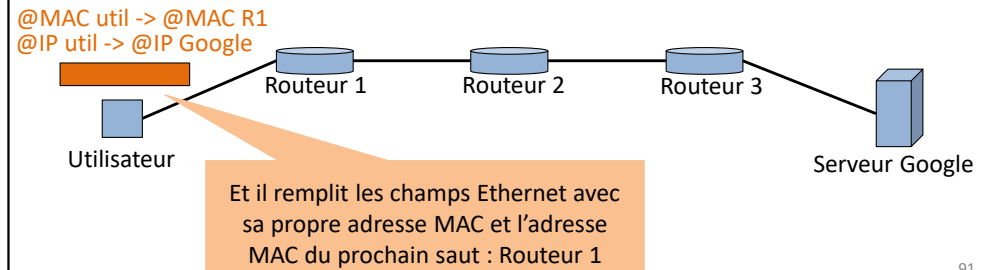
- Un paquet IP circulant sur Ethernet possède donc :
  - Une adresse IP source et une adresse IP destination dans l'en-tête du paquet IP
  - Une adresse MAC source et une adresse MAC destination dans l'en-tête de la trame Ethernet
- Les adresses IP ne sont pas modifiées pendant la durée de vie du message
  - Sauf si le paquet passe par une passerelle NAT
- Les adresses MAC changent à chaque saut IP



90

# Adressage bout-en-bout

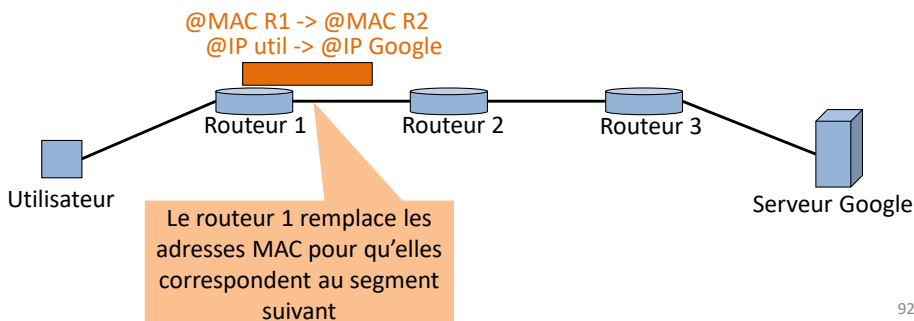
- Un paquet IP circulant sur Ethernet possède donc :
  - Une adresse IP source et une adresse IP destination dans l'en-tête du paquet IP
  - Une adresse MAC source et une adresse MAC destination dans l'en-tête de la trame Ethernet
- Les adresses IP ne sont pas modifiées pendant la durée de vie du message
  - Sauf si le paquet passe par une passerelle NAT
- Les adresses MAC changent à chaque saut IP



91

# Adressage bout-en-bout

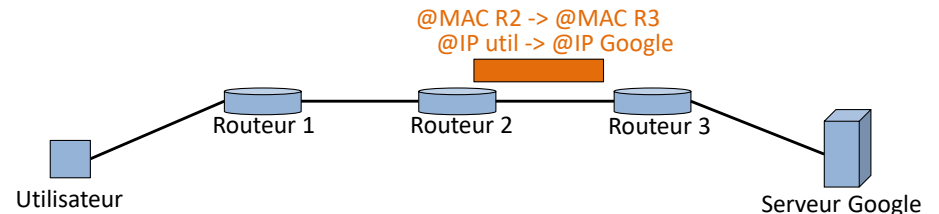
- Un paquet IP circulant sur Ethernet possède donc :
  - Une adresse IP source et une adresse IP destination dans l'en-tête du paquet IP
  - Une adresse MAC source et une adresse MAC destination dans l'en-tête de la trame Ethernet
- Les adresses IP ne sont pas modifiées pendant la durée de vie du message
  - Sauf si le paquet passe par une passerelle NAT
- Les adresses MAC changent à chaque saut IP



92

# Adressage bout-en-bout

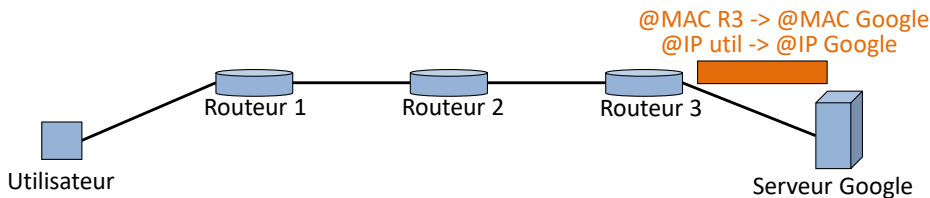
- Un paquet IP circulant sur Ethernet possède donc :
  - Une adresse IP source et une adresse IP destination dans l'en-tête du paquet IP
  - Une adresse MAC source et une adresse MAC destination dans l'en-tête de la trame Ethernet
- Les adresses IP ne sont pas modifiées pendant la durée de vie du message
  - Sauf si le paquet passe par une passerelle NAT
- Les adresses MAC changent à chaque saut IP



93

# Adressage bout-en-bout

- Un paquet IP circulant sur Ethernet possède donc :
  - Une adresse IP source et une adresse IP destination dans l'en-tête du paquet IP
  - Une adresse MAC source et une adresse MAC destination dans l'en-tête de la trame Ethernet
- Les adresses IP ne sont pas modifiées pendant la durée de vie du message
  - Sauf si le paquet passe par une passerelle NAT
- Les adresses MAC changent à chaque saut IP



94

# Table de routage

- Le routage IP se fait de proche en proche
- Un routeur IP cherche le « prochain saut » pour chaque paquet reçu en fonction de l'adresse destination
- Cette information est contenue dans la table de routage
- Fonctionnement :
  - Lecture de l'adresse IP destination
  - Comparaison avec les sous-réseaux connus dans la table de routage
  - Si l'adresse est reconnue, envoi du paquet au prochain saut en passant par l'interface indiquée

Address	Netmask	Next hop	Interface

95

# Table de routage

- Le routage IP se fait de proche en proche
- Un routeur IP cherche le « prochain saut » pour chaque paquet reçu en fonction de l'adresse destination
- Cette information est contenue dans la table de routage
- Fonctionnement :
  - Lecture de l'adresse IP destination
  - Comparaison avec les sous-réseaux connus dans la table de routage
  - Si l'adresse est reconnue, envoi du paquet au prochain saut en passant par l'interface indiquée

Les 2 premières colonnes servent à identifier les sous-réseaux connus : adresse + masque

Address	Netmask	Next hop	Interface

96

# Table de routage

- Le routage IP se fait de proche en proche
- Un routeur IP cherche le « prochain saut » pour chaque paquet reçu en fonction de l'adresse destination
- Cette information est contenue dans la table de routage
- Fonctionnement :
  - Lecture de l'adresse IP destination
  - Comparaison avec les sous-réseaux connus dans la table de routage
  - Si l'adresse est reconnue, envoi du paquet au prochain saut en passant par l'interface indiquée

Cette colonne indique le prochain saut IP. Lorsqu'il est indiqué « local » ou « link » cela signifie que l'on est directement relié au sous-réseau correspondant.

Address	Netmask	Next hop	Interface

97



# Table de routage

- Le routage IP se fait de proche en proche
- Un routeur IP cherche le « prochain saut » pour chaque paquet reçu en fonction de l'adresse destination
- Cette information est contenue dans la table de routage
- Fonctionnement :
  - Lecture de l'adresse IP destination
  - Comparaison avec les sous-réseaux connus dans la table de routage
  - Si l'adresse est reconnue, envoi du paquet par l'interface indiquée

Cette colonne indique l'interface sur laquelle transférer le paquet

Address	Netmask	Next hop	Interface

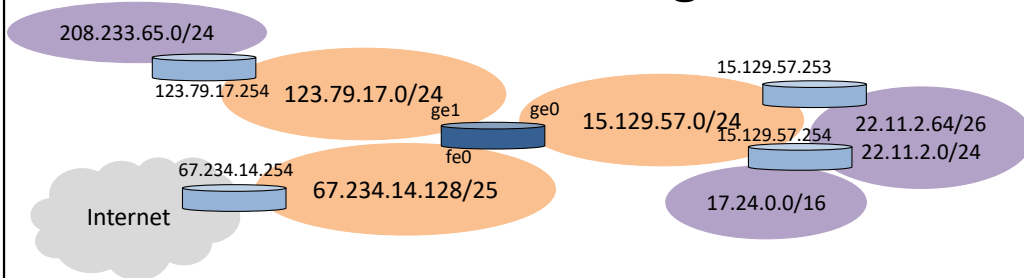
98

# Table de routage

- Local
  - Si la machine est directement reliée au sous-réseau destination
  - Le paquet est envoyé sur le réseau local
  - Le prochain saut est l'adresse IP destination du message
- Règle du plus long préfixe
  - Si plusieurs sous-réseaux correspondent à l'adresse
  - On choisit celui qui a le préfixe le plus long
  - C'est celui qui correspond « le plus »
- Routage par défaut
  - Il y a toujours une ligne qui correspond à toute les adresses possibles
  - Elle a le préfixe 0
  - C'est le sous-réseau 0.0.0.0/0
  - Le prochain saut de l'adresse par défaut est appelé la passerelle par défaut (default gateway)
  - C'est la sortie utilisée quand aucun autre sous-réseau ne convient
  - En pratique, la majorité du routage est effectué par défaut

99

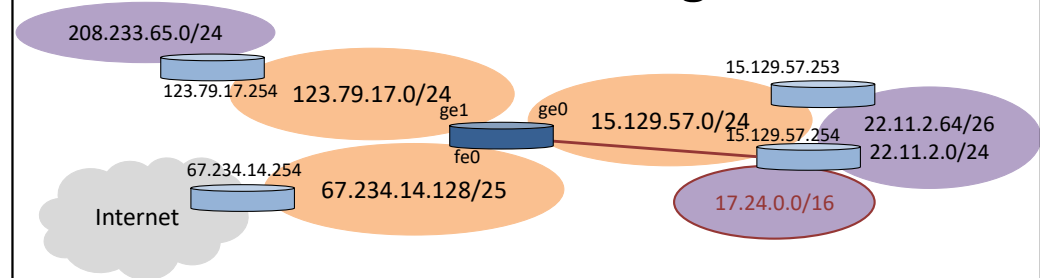
# Table de routage



Address	Netmask	Next hop	Interface
17.24.0.0	255.255.0.0	15.129.57.254	ge0
208.233.65.0	255.255.255.0	123.79.17.254	ge1
22.11.2.0	255.255.255.0	15.129.57.254	ge0
22.11.2.64	255.255.255.192	15.129.57.253	ge0
15.129.57.0	255.255.255.0	Local	ge0
123.79.17.0	255.255.255.0	Local	ge1
67.234.14.128	255.255.255.128	Local	fe0
0.0.0.0	0.0.0.0	67.234.14.254	fe0

100

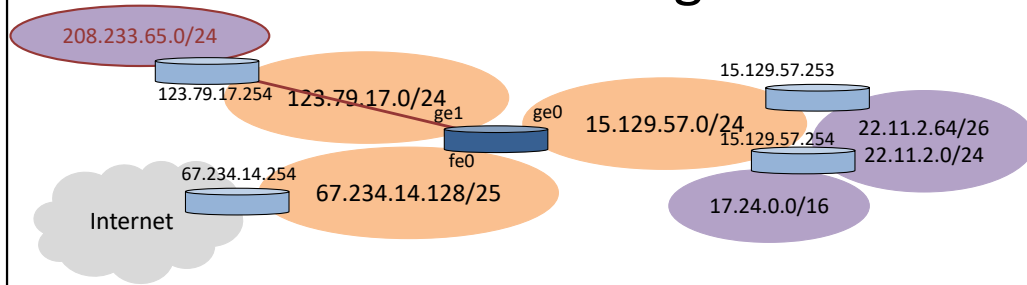
# Table de routage



Address	Netmask	Next hop	Interface
17.24.0.0	255.255.0.0	15.129.57.254	ge0
208.233.65.0	255.255.255.0	123.79.17.254	ge1
22.11.2.0	255.255.255.0	15.129.57.254	ge0
22.11.2.64	255.255.255.192	15.129.57.253	ge0
15.129.57.0	255.255.255.0	Local	ge0
123.79.17.0	255.255.255.0	Local	ge1
67.234.14.128	255.255.255.128	Local	fe0
0.0.0.0	0.0.0.0	67.234.14.254	fe0

101

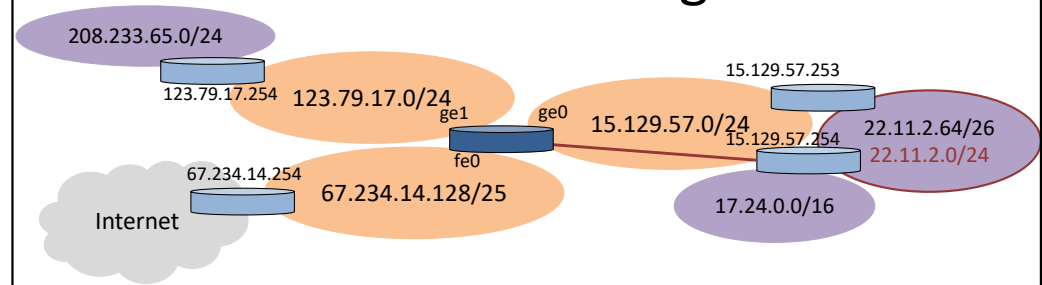
## Table de routage



Address	Netmask	Next hop	Interface
17.24.0.0	255.255.0.0	15.129.57.254	ge0
208.233.65.0	255.255.255.0	123.79.17.254	ge1
22.11.2.0	255.255.255.0	15.129.57.254	ge0
22.11.2.64	255.255.255.192	15.129.57.253	ge0
15.129.57.0	255.255.255.0	Local	ge0
123.79.17.0	255.255.255.0	Local	ge1
67.234.14.128	255.255.255.128	Local	fe0
0.0.0.0	0.0.0.0	67.234.14.254	fe0

102

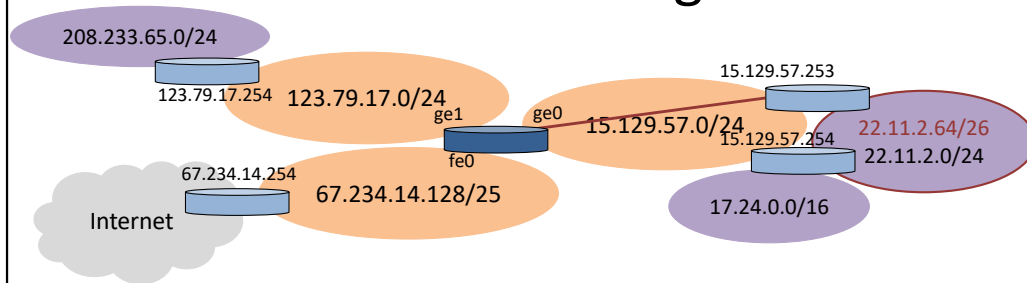
## Table de routage



Address	Netmask	Next hop	Interface
17.24.0.0	255.255.0.0	15.129.57.254	ge0
208.233.65.0	255.255.255.0	123.79.17.254	ge1
22.11.2.0	255.255.255.0	15.129.57.254	ge0
22.11.2.64	255.255.255.192	15.129.57.253	ge0
15.129.57.0	255.255.255.0	Local	ge0
123.79.17.0	255.255.255.0	Local	ge1
67.234.14.128	255.255.255.128	Local	fe0
0.0.0.0	0.0.0.0	67.234.14.254	fe0

103

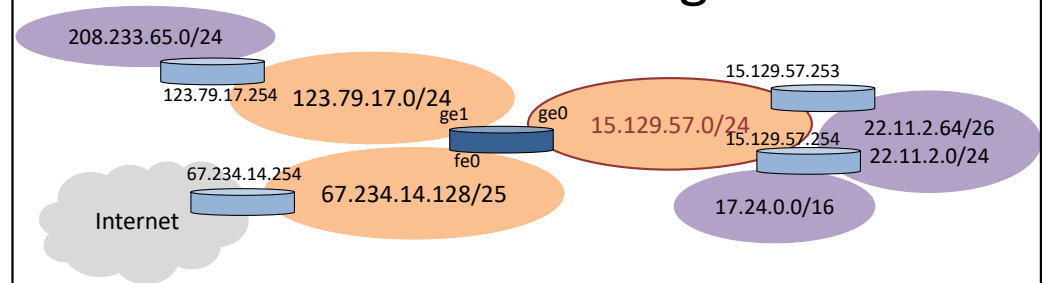
## Table de routage



Address	Netmask	Next hop	Interface
17.24.0.0	255.255.0.0	15.129.57.254	ge0
208.233.65.0	255.255.255.0	123.79.17.254	ge1
22.11.2.0	255.255.255.0	15.129.57.254	ge0
22.11.2.64	255.255.255.192	15.129.57.253	ge0
15.129.57.0	255.255.255.0	Local	ge0
123.79.17.0	255.255.255.0	Local	ge1
67.234.14.128	255.255.255.128	Local	fe0
0.0.0.0	0.0.0.0	67.234.14.254	fe0

104

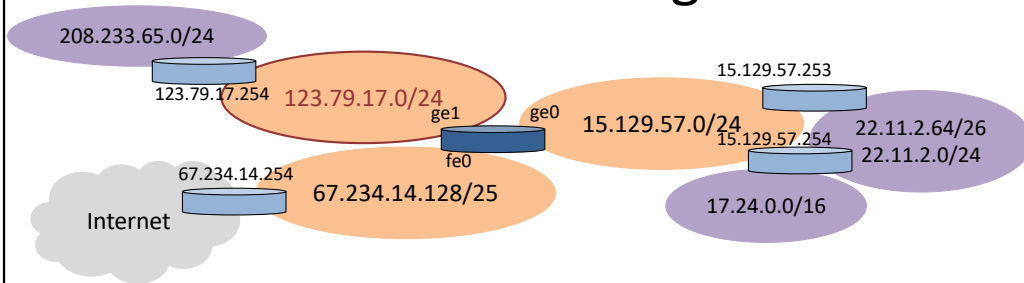
## Table de routage



Address	Netmask	Next hop	Interface
17.24.0.0	255.255.0.0	15.129.57.254	ge0
208.233.65.0	255.255.255.0	123.79.17.254	ge1
22.11.2.0	255.255.255.0	15.129.57.254	ge0
22.11.2.64	255.255.255.192	15.129.57.253	ge0
15.129.57.0	255.255.255.0	Local	ge0
123.79.17.0	255.255.255.0	Local	ge1
67.234.14.128	255.255.255.128	Local	fe0
0.0.0.0	0.0.0.0	67.234.14.254	fe0

105

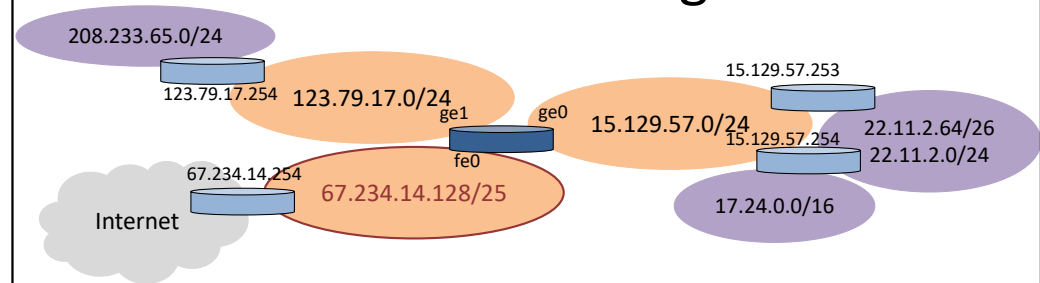
## Table de routage



Address	Netmask	Next hop	Interface
17.24.0.0	255.255.0.0	15.129.57.254	ge0
208.233.65.0	255.255.255.0	123.79.17.254	ge1
22.11.2.0	255.255.255.0	15.129.57.254	ge0
22.11.2.64	255.255.255.192	15.129.57.253	ge0
15.129.57.0	255.255.255.0	Local	ge0
123.79.17.0	255.255.255.0	Local	ge1
67.234.14.128	255.255.255.128	Local	fe0
0.0.0.0	0.0.0.0	67.234.14.254	fe0

106

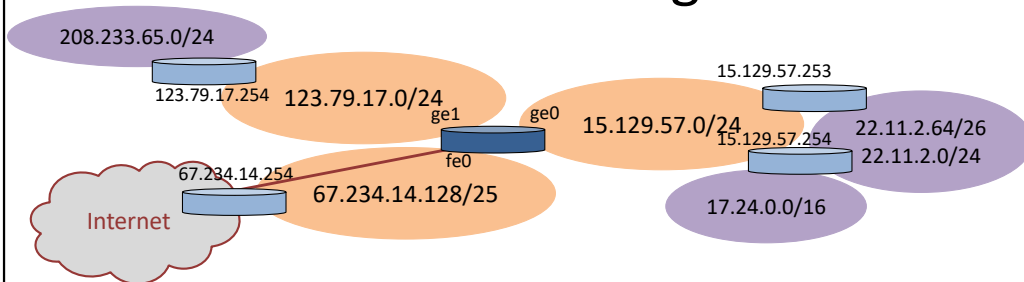
## Table de routage



Address	Netmask	Next hop	Interface
17.24.0.0	255.255.0.0	15.129.57.254	ge0
208.233.65.0	255.255.255.0	123.79.17.254	ge1
22.11.2.0	255.255.255.0	15.129.57.254	ge0
22.11.2.64	255.255.255.192	15.129.57.253	ge0
15.129.57.0	255.255.255.0	Local	ge0
123.79.17.0	255.255.255.0	Local	ge1
67.234.14.128	255.255.255.128	Local	fe0
0.0.0.0	0.0.0.0	67.234.14.254	fe0

107

## Table de routage



Address	Netmask	Next hop	Interface
17.24.0.0	255.255.0.0	15.129.57.254	ge0
208.233.65.0	255.255.255.0	123.79.17.254	ge1
22.11.2.0	255.255.255.0	15.129.57.254	ge0
22.11.2.64	255.255.255.192	15.129.57.253	ge0
15.129.57.0	255.255.255.0	Local	ge0
123.79.17.0	255.255.255.0	Local	ge1
67.234.14.128	255.255.255.128	Local	fe0
0.0.0.0	0.0.0.0	67.234.14.254	fe0

108

## Couche transport

- La couche 4 transport est responsable de la fiabilisation de la liaison bout-en-bout
- Sur Internet, il existe 2 protocoles « principaux » :
  - UDP
    - User Datagram Protocol
    - Sans connexion
    - Non fiable
    - Simple
  - TCP
    - Transmission Control Protocol
    - Orienté connexion
    - Fiable
    - Complexe

109

# Socket

- L'interface entre la couche 4 et les couches supérieures (services) se fait via des sockets identifiées par une adresse IP et un numéro de port
- Une socket est point d'accès aux services de transport en mode client-serveur (accessibles depuis une interface programmation dite API socket en C, Java ou Python par exemple)
  - Socket client qui joint le serveur (fonction connect)
  - Socket serveur qui attend les requêtes entrantes (fonction listen)



110

# Numéros de ports

- Les sockets sont identifiées par une adresse IP et un numéro de port
- La communication entre 2 entités est identifiée par un numéro de port au sein de chaque entité
  - Généralement les numéros sont différents entre les 2 entités
- Il existe 3 catégories de numéros de ports définies par l'IANA :
  - Well known
    - 0 à 1023
    - Exemple : 80 pour un serveur HTTP
  - Registered
    - 1024 à 49151
    - Services propriétaires
  - Dynamic
    - 49152 à 65535
    - Allocation automatique par l'OS
    - Exemple : 55678 pour un client Firefox
- Ce sont ces numéros de ports qui sont utilisés pour le NAT

111

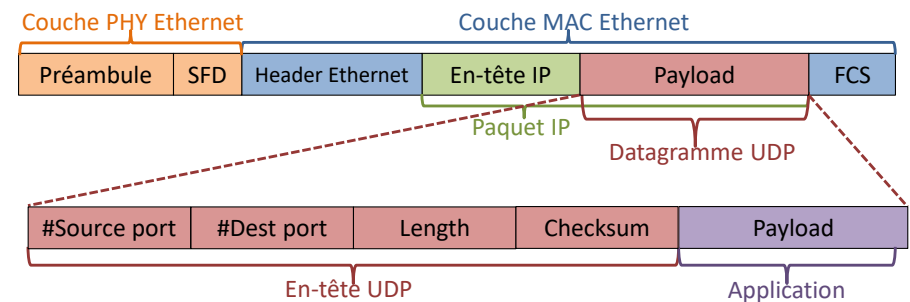
# UDP

- UDP (User Datagram Protocol)
  - Identifié par le numéro 17 dans le champs « upper layer » d'IP
  - Sans connexion :
    - Les datagrammes sont envoyés directement quand les données arrivent de la couche supérieure
    - Le récepteur n'est pas averti au préalable
  - Non fiable :
    - Détection d'erreur par un checksum
    - Aucun mécanisme de reprise sur erreur
- Avantages :
  - Simple : pas de paramètres
  - Rapide : pas de délai d'établissement de connexion
- Utilisé pour :
  - Services simples
  - Services « fire and forget »
  - Services temps réel
- UDP well-known ports :
  - 22 SSH
  - 53 DNS
  - 67-68 serveur-client DHCP

112

# Datagramme UDP

- Un datagramme UDP est la charge (payload) contenue dans un paquet IP :



- L'en-tête UDP est sur 8 octets (4x 2 octets) et comprend :
  - Numéro de port source
  - Numéro de port destination
  - Length : longueur totale du datagramme (en-tête + payload)
  - Checksum : peut être mis à 0 si non utilisé

113

# TCP

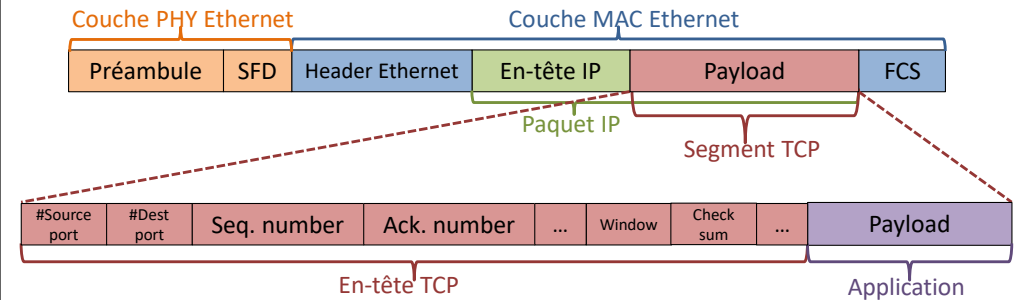
- TCP (Transmission Control Protocol)
  - Identifié par le numéro 6 dans le champs « upper layer » d'IP
  - Orienté connexion
    - Ouverture de connexion avant chaque transfert de données, et fermeture après
    - Le récepteur est impliqué
    - Protocole à états (connexion fermée, en cours d'ouverture, établie, en cours de fermeture)
    - Full duplex
  - Fiable
    - Détection d'erreur (checksum) et de perte (numérotation)
    - Conservation de l'ordre des données
    - Reprise sur erreur et sur perte
    - Contrôle de flux (par rapport au récepteur)
    - Contrôle de congestion (par rapport au lien)
- Flux TCP
  - Vu de l'application, on envoie un flux d'octets
  - Chaque octet (byte) est numéroté
  - Les octets sont transmis sous forme de segments de plusieurs octets par TCP à la couche inférieure (IP)
  - La taille des segments évolue pendant la connexion TCP (contrôle de flux et de congestion)
- TCP well-known ports
 

– 20-21 FTP	– 80 HTTP
– 22 SSH	– 110 POP3
– 23 Telnet	– 143 IMAP
– 25 SMTP	

114

# Segment TCP

- Un segment TCP est la charge (payload) contenue dans un paquet IP :

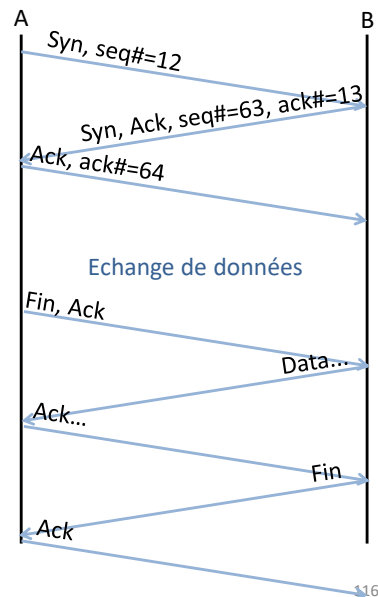


- L'en-tête TCP est sur 20 octets minimum et comprend :
  - Numéro de port source
  - Numéro de port destination
  - Sequence number : numéro du premier octet transporté (équivalent du N(S) dans ARQ)
  - Acknowledgement number : numéro du prochain octet attendu (équivalent du N(R) dans ARQ)
  - Offset : taille de l'en-tête
  - Différents flags (drapeaux) : pour signaler si c'est un message d'ouverture ou de fermeture, un message urgent, unack...
  - Window : taille de la fenêtre de réception (en nombre d'octets acceptés)
  - Checksum : sur l'en-tête et les données
  - Des options

115

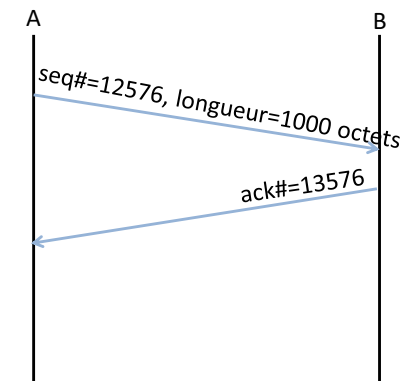
## Ouverture et fermeture de connexion

- Ouverture de connexion
  - 3-way handshake
  - A l'aide du flag « Syn »
  - Synchronisation des sequence number dans les 2 sens
- Fermeture
  - Envoi du flag « Fin » dès qu'on a plus de données à envoyer
  - La connexion est fermée lorsque les 2 parties ont envoyés le flag « Fin » et que ces messages ont été acquittés



## Reprise sur erreur

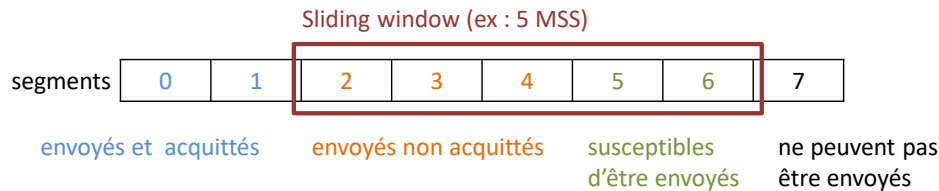
- Tout envoi est acquitté
- Si aucun acquittement n'est reçu au bout d'un certain temps, il y a retransmission
  - RTO : Retransmission Time Out
  - Calculé en fonction du RTT (Round Trip Time) actuel
- Principe de la reprise sur erreur
  - Plusieurs segments peuvent être envoyés et acquittés à la fois
  - En cas d'erreur, on renvoie tout depuis l'erreur



117

# Fenêtre d'émission

- La fenêtre d'émission est glissante (sliding window) : elle avance avec les acquittements reçus

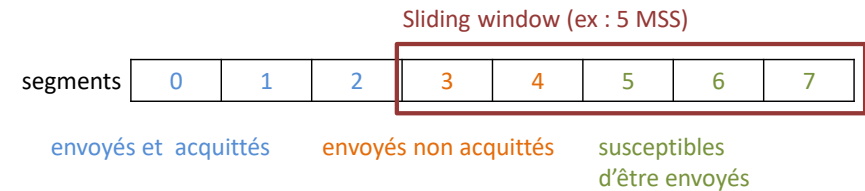


- Sa taille (WND) dépend du contrôle de flux et du contrôle de congestion
  - 1 MSS = Maximum Segment Size = Taille d'un segment = MTU(dépend du lien) – IP header – TCP header
  - La taille de la fenêtre d'émission est donnée en nombre de MSS
  - RWND (Receiver Window)
    - Donnée dans le champ « window » de l'en-tête TCP, mis à jour à chaque envoi
    - But : éviter de noyer le récepteur
  - CWND (Congestion Window)
    - Dépend du nombre de segments bien reçus et perdus, évolue à chaque envoi reçu ou perdu
    - But : éviter d'engorger le lien
  - WND = min (RWND, CWND)

118

# Fenêtre d'émission

- La fenêtre d'émission est glissante (sliding window) : elle avance avec les acquittements reçus



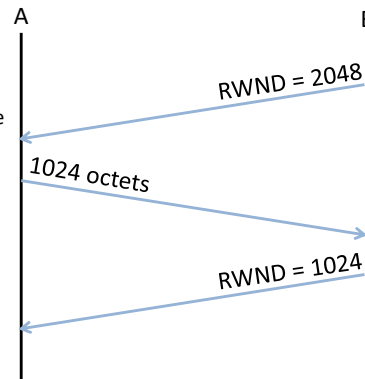
- Sa taille (WND) dépend du contrôle de flux et du contrôle de congestion
  - 1 MSS = Maximum Segment Size = Taille d'un segment = MTU– IP header – TCP header
  - La taille de la fenêtre d'émission est donnée en nombre de MSS
  - RWND (Receiver Window)
    - Donnée dans le champ « window » de l'en-tête TCP, mis à jour à chaque envoi
    - But : éviter de noyer le récepteur
  - CWND (Congestion Window)
    - Dépend du nombre de segments bien reçus et perdus, évolue à chaque envoi reçu ou perdu
    - But : éviter d'engorger le lien
  - WND = min (RWND, CWND)

119

# Fenêtre de réception

- Dans chaque segment TCP, la taille de la fenêtre de réception disponible est indiquée

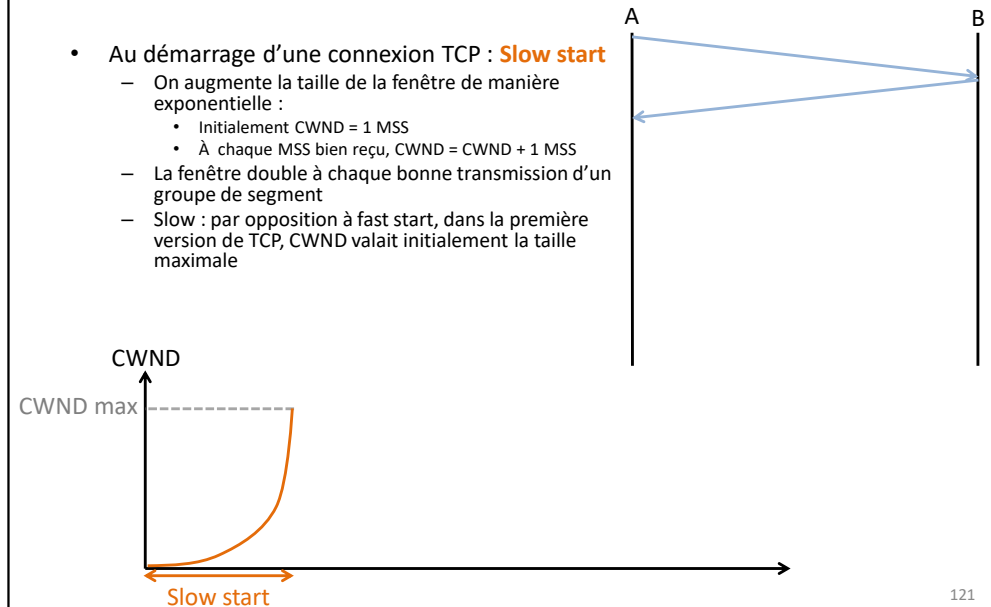
- Lorsque la taille atteint 0
  - On dit que la fenêtre est fermée (closed window)
  - Le récepteur ne peut pas signaler la réouverture de sa fenêtre s'il n'a pas de données à envoyer (un message d'acquittement ne peut pas être envoyé sans avoir reçu de données)
  - L'émetteur continue de tester en permanence la fenêtre de réception avec l'envoi d'un segment à 1 octet
  - Pour ne pas saturer sa fenêtre avec ces envois, le récepteur peut :
    - Attendre l'envoi de MSS segments pour rouvrir sa fenêtre
    - Attendre que sa fenêtre soit à moitié vide pour la rouvrir



120

# Fenêtre de congestion

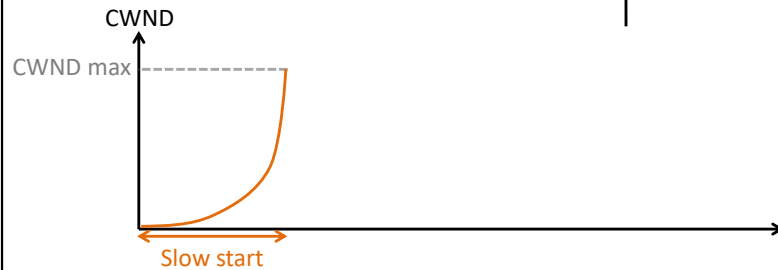
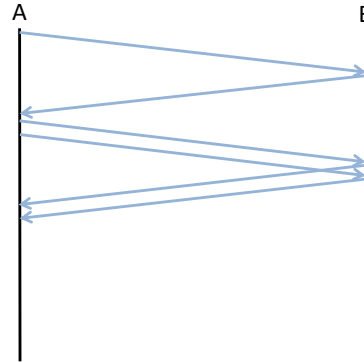
- Au démarrage d'une connexion TCP : **Slow start**
  - On augmente la taille de la fenêtre de manière exponentielle :
    - Initialement CWND = 1 MSS
    - À chaque MSS bien reçu, CWND = CWND + 1 MSS
  - La fenêtre double à chaque bonne transmission d'un groupe de segment
  - Slow : par opposition à fast start, dans la première version de TCP, CWND valait initialement la taille maximale



121

# Fenêtre de congestion

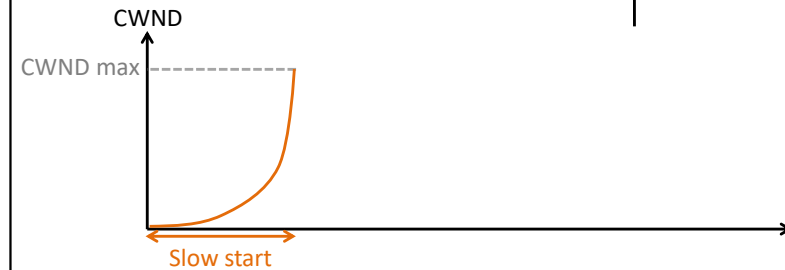
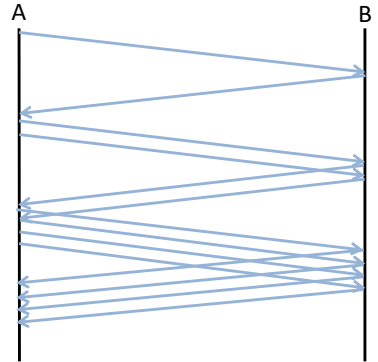
- Au démarrage d'une connexion TCP : **Slow start**
  - On augmente la taille de la fenêtre de manière exponentielle :
    - Initialement  $CWND = 1 \text{ MSS}$
    - À chaque MSS bien reçu,  $CWND = CWND + 1 \text{ MSS}$
  - La fenêtre double à chaque bonne transmission d'un groupe de segment
  - Slow : par opposition à fast start, dans la première version de TCP, CWND valait initialement la taille maximale



122

# Fenêtre de congestion

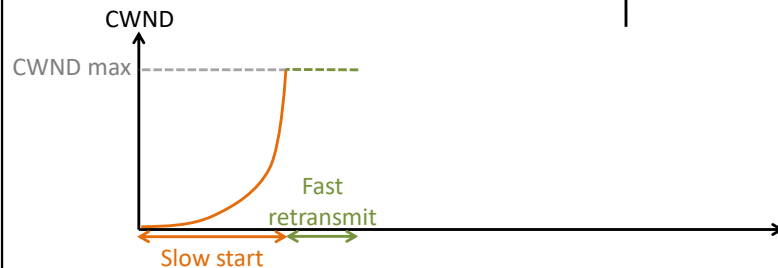
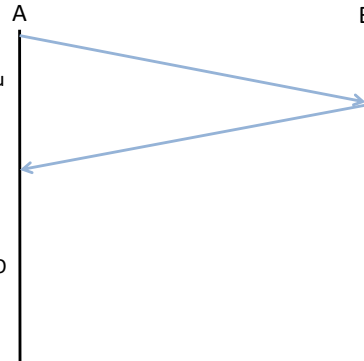
- Au démarrage d'une connexion TCP : **Slow start**
  - On augmente la taille de la fenêtre de manière exponentielle :
    - Initialement  $CWND = 1 \text{ MSS}$
    - À chaque MSS bien reçu,  $CWND = CWND + 1 \text{ MSS}$
  - La fenêtre double à chaque bonne transmission d'un groupe de segment
  - Slow : par opposition à fast start, dans la première version de TCP, CWND valait initialement la taille maximale



123

# Fenêtre de congestion

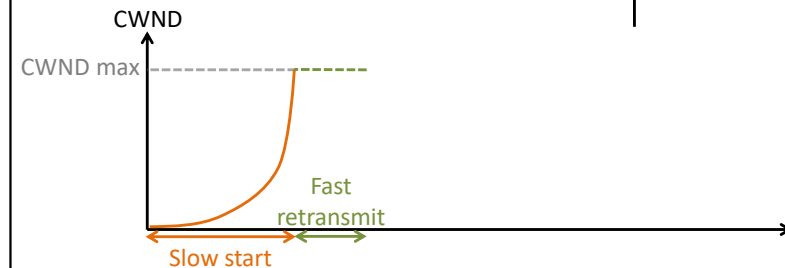
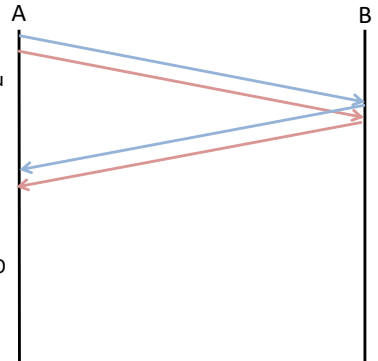
- À la première erreur : **Fast Retransmit**
  - On cherche à détecter les erreurs avant expiration du timer RTO :
    - On signale l'erreur par un acquittement dupliqué
    - À chaque segment reçu, on redemande le segment manquant
    - Au bout de 3 acquittements dupliqués (donc 4 acquittements avec le même numéro), il y a retransmission du segment perdu
  - Fast : par opposition à la première version de TCP où on attendait l'expiration du timer RTO. Si le timer RTO expire, on retourne en slow start.



124

# Fenêtre de congestion

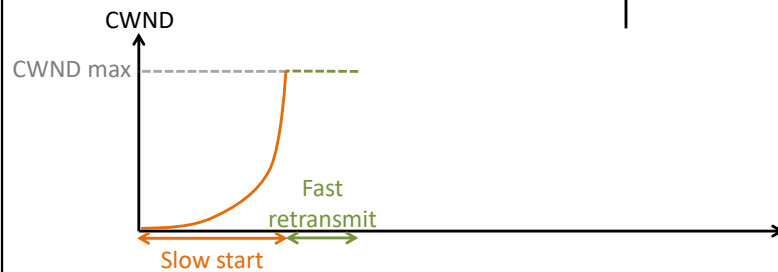
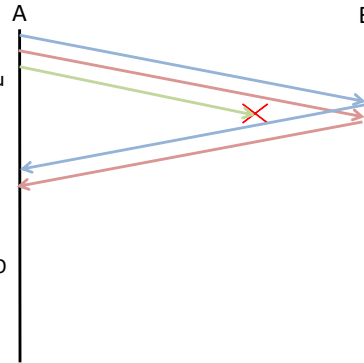
- À la première erreur : **Fast Retransmit**
  - On cherche à détecter les erreurs avant expiration du timer RTO :
    - On signale l'erreur par un acquittement dupliqué
    - À chaque segment reçu, on redemande le segment manquant (Go-back-N)
    - Au bout de 3 acquittements dupliqués (donc 4 acquittements avec le même numéro), il y a retransmission du segment perdu
  - Fast : par opposition à la première version de TCP où on attendait l'expiration du timer RTO. Si le timer RTO expire, on retourne en slow start.



125

# Fenêtre de congestion

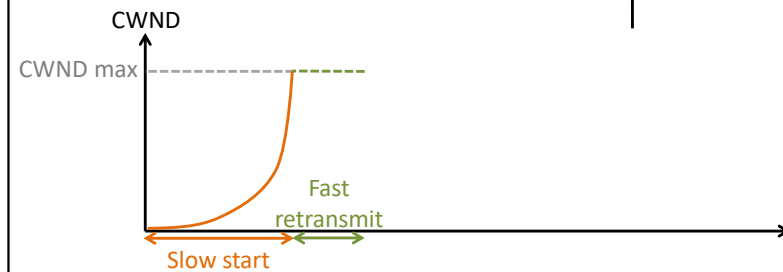
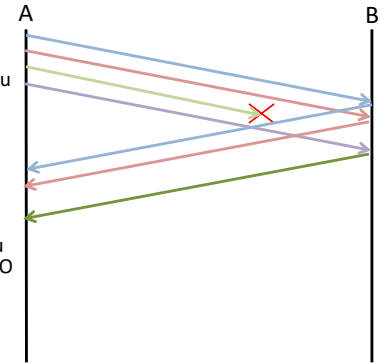
- À la première erreur : **Fast Retransmit**
  - On cherche à détecter les erreurs avant expiration du timer RTO :
    - On signale l'erreur par un acquittement dupliqué
    - À chaque segment reçu, on redemande le segment manquant (Go-back-N)
    - Au bout de 3 acquittements dupliqués (donc 4 acquittements avec le même numéro), il y a retransmission du segment perdu
  - Fast : par opposition à la première version de TCP où on attendait l'expiration du timer RTO. Si le timer RTO expire, on retourne en slow start.



126

# Fenêtre de congestion

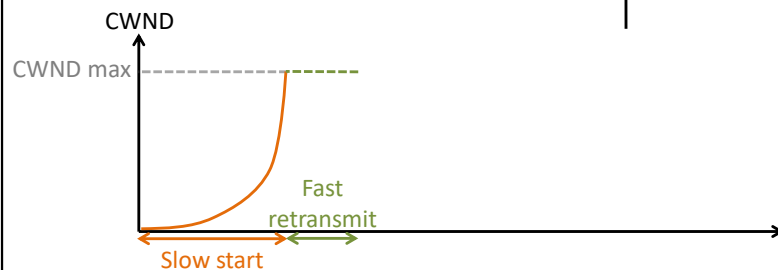
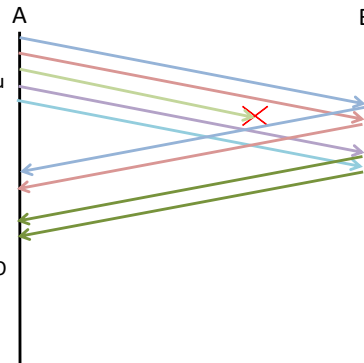
- À la première erreur : **Fast Retransmit**
  - On cherche à détecter les erreurs avant expiration du timer RTO :
    - On signale l'erreur par un acquittement dupliqué
    - À chaque segment reçu, on redemande le segment manquant (Go-back-N)
    - Au bout de 3 acquittements dupliqués (donc 4 acquittements avec le même numéro), il y a retransmission du segment perdu
  - Fast : par opposition à la première version de TCP où on attendait l'expiration du timer RTO. Si le timer RTO expire, on retourne en slow start.



127

# Fenêtre de congestion

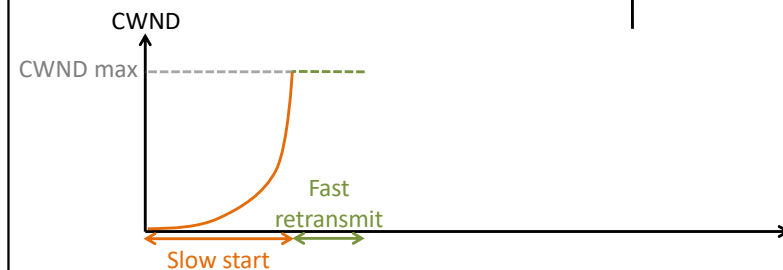
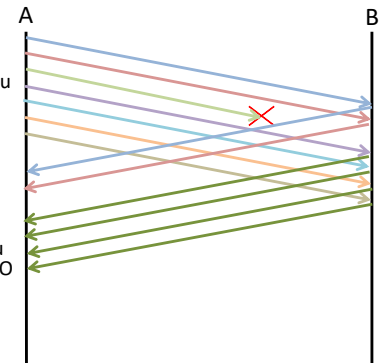
- À la première erreur : **Fast Retransmit**
  - On cherche à détecter les erreurs avant expiration du timer RTO :
    - On signale l'erreur par un acquittement dupliqué
    - À chaque segment reçu, on redemande le segment manquant (Go-back-N)
    - Au bout de 3 acquittements dupliqués (donc 4 acquittements avec le même numéro), il y a retransmission du segment perdu
  - Fast : par opposition à la première version de TCP où on attendait l'expiration du timer RTO. Si le timer RTO expire, on retourne en slow start.



128

# Fenêtre de congestion

- À la première erreur : **Fast Retransmit**
  - On cherche à détecter les erreurs avant expiration du timer RTO :
    - On signale l'erreur par un acquittement dupliqué
    - À chaque segment reçu, on redemande le segment manquant (Go-back-N)
    - Au bout de 3 acquittements dupliqués (donc 4 acquittements avec le même numéro), il y a retransmission du segment perdu
  - Fast : par opposition à la première version de TCP où on attendait l'expiration du timer RTO. Si le timer RTO expire, on retourne en slow start.

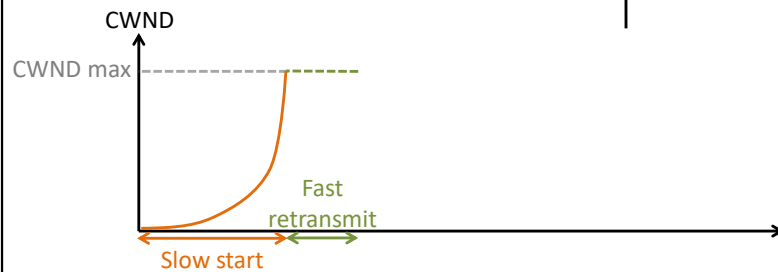
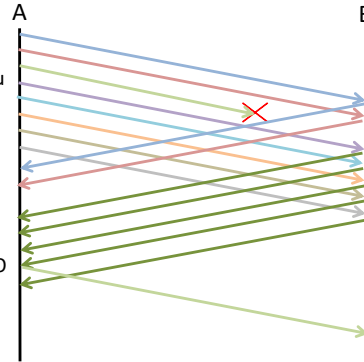


129



# Fenêtre de congestion

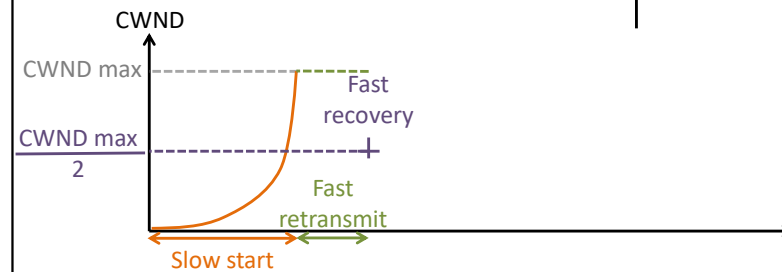
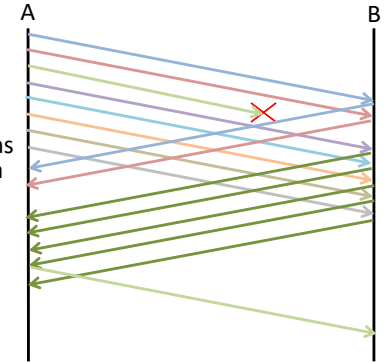
- À la première erreur : **Fast Retransmit**
  - On cherche à détecter les erreurs avant expiration du timer RTO :
    - On signale l'erreur par un acquittement dupliqué
    - À chaque segment reçu, on redemande le segment manquant (Go-back-N)
    - Au bout de 3 acquittements dupliqués (donc 4 acquittements avec le même numéro), il y a retransmission du segment perdu
  - Fast : par opposition à la première version de TCP où on attendait l'expiration du timer RTO. Si le timer RTO expire, on retourne en slow start.



130

# Fenêtre de congestion

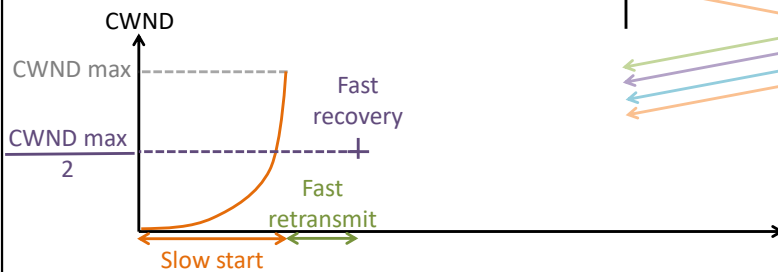
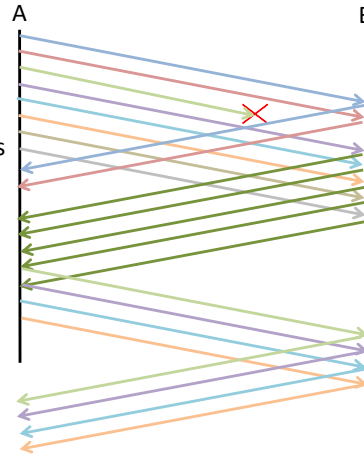
- Une fois l'erreur détectée : **Fast Recovery**
  - On ne repart pas à une fenêtre de taille 1
  - On repart directement à  $CWND_{max} / 2$
  - Fast : par opposition aux anciennes versions de TCP où on repartait en slow start jusqu'à atteindre  $CWND_{max}/2$



131

# Fenêtre de congestion

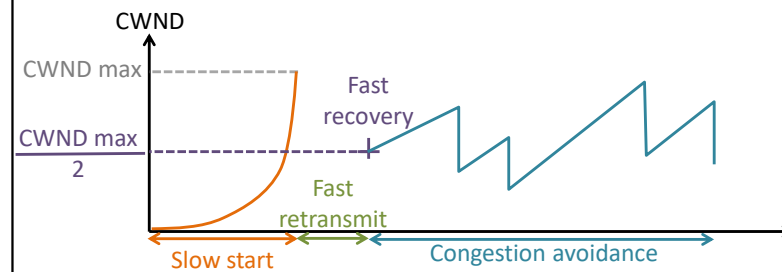
- Une fois l'erreur détectée : **Fast Recovery**
  - On ne repart pas à une fenêtre de taille 1
  - On repart directement à  $CWND_{max} / 2$
  - Fast : par opposition aux anciennes versions de TCP où on repartait en slow start jusqu'à atteindre  $CWND_{max}/2$



132

# Fenêtre de congestion

- Ensuite : **Congestion Avoidance**
  - La taille de la fenêtre continue d'évoluer à chaque segment bien reçu ou perdu selon le mécanisme AIMD
  - On suppose que la fenêtre contient k segments :  $CWND = k \text{ MSS}$
  - AI (Additive Increase)
    - Si les k MSS sont bien reçus
    - Alors on augmente la fenêtre de 1 segment :  $CWND = (k+1) \text{ MSS}$
  - MD (Multiplicative Decrease)
    - Si un au moins des k segments est perdu
    - Alors on divise la fenêtre par 2 :  $CWND = k/2 \text{ MSS}$



133

# Plan contrôle

134

# Plan contrôle

- Dans Internet, il n’y a pas de pile protocolaire unique pour la signalisation
- Il y a des protocoles qui répondent à des besoins spécifiques :
  - Retrouver l’adresse MAC correspondant à une adresse IP
    - ARP
  - Obtenir une adresse IP
    - RARP et DHCP
  - Gérer les erreurs de transmissions IP
    - ICMP
  - Remplir la table de routage
    - Intra AS
      - OSPF
      - RIP
    - Inter AS
      - BGP

135

## ARP

- ARP (Address Resolution Protocol) est le protocole qui permet de demander l’adresse MAC d’une machine connue par son adresse IP
- Problème :
  - Quand un routeur lit sa table de routage, il choisit le prochain saut « next hop »
  - Le prochain saut est identifié par une adresse IP
  - Pour envoyer le paquet IP, il faut l’encapsuler dans une trame Ethernet avec la bonne adresse MAC
  - Il faut donc obtenir l’adresse MAC correspondant à l’adresse IP du « next hop »
- Fonctionnement :
  - ARP est encapsulé directement dans Ethernet
  - Requête ARP : « Who has @IP ? » envoyée en broadcast MAC
  - Réponse ARP : « @MAC correspondant à @IP demandé » envoyé en unicast par le propriétaire de l’adresse IP
  - Cache ARP : chaque machine maintient une table avec les correspondances adresses MAC – adresses IP pour éviter les demandes répétitives

Adresse IP	Adresse MAC
192.168.1.1	b8:26:6c:fb:33:ba
192.168.1.13	6c:40:80:a2:88:dd

- Remplacé par ND (Neighbor Discovery) dans IPv6

136

## RARP

- RARP (Reverse ARP) est le protocole qui permet d’obtenir une adresse IP automatiquement à partir de son adresse MAC
- Problème :
  - Une station rejoint un sous-réseau (branchement Ethernet uniquement)
  - Pour pouvoir communiquer sur le réseau il lui faut une adresse IP correspondant à son sous-réseau
- Fonctionnement :
  - RARP est comme ARP encapsulé dans Ethernet
  - La station nouvellement connectée envoie une requête RARP en broadcast Ethernet
  - Le serveur RARP présent sur le réseau répond en unicast Ethernet lui attribuant une adresse IP
- Limitations :
  - Le masque est déduit de l’adresse IP avec le ClassFull Addressing
  - La réponse ne comprend pas les autres paramètres du réseau : routeur par défaut, serveur DNS...
  - RARP est obsolète

137

# DHCP

- DHCP (Dynamic Host Configuration Protocol) est le protocole qui permet d'obtenir une adresse IP automatiquement (sans configuration manuelle) quand on se connecte à un réseau
- Problème :
  - Une station rejoint un sous-réseau (branchement Ethernet, WiFi)
  - Pour pouvoir communiquer sur le réseau il lui faut une adresse IP correspondant à son sous-réseau
- Fonctionnement :
  - DHCP est encapsulé dans UDP sur IP sur Ethernet ou WiFi
  - La station nouvellement connectée envoie un message « DHCP Discover » en utilisant le port source 68 et le port destination 67 (well-known ports) et avec adresse IP source l'adresse par défaut 0.0.0.0, et en adresse IP destination l'adresse de broadcast 255.255.255.255
  - Le serveur DHCP présent sur le réseau répond en utilisant les mêmes numéros de ports (inversés), son adresse IP en adresse source et l'adresse de broadcast 255.255.255.255 en adresse destination
  - La réponse comprend :
    - L'adresse IP affectée à la machine
    - Le masque de sous-réseau
    - Le routeur par défaut
    - Le ou les serveurs DNS
    - La durée de validité (lease time)

138

# ICMP

- ICMP (Internet Control Message Protocol) est le protocole de signalisation d'IP
- ICMP permet de signaler une erreur dans la transmission d'un paquet IP

- Fonctionnement :

- ICMP est encapsulé dans IP avec le champs « upper layer » = 1
- ICMPv6 existe pour IPv6
- Un message ICMP comporte :
  - Un champ « Type »
  - Un champ « Code »
  - Les 8 premiers octets du message IP provoquant l'erreur le cas échéant
- Le type et le code indiquent le type d'erreur (voir exemples ci-contre)

Type	Code	Description
3	0	Destination network unreachable
3	1	Destination host unreachable
3	2	Destination protocol unreachable
3	3	Destination port unreachable
3	6	Destination network unknown
3	7	Destination host unknown
0	0	Echo reply (ping)
8	0	Echo request (ping)
11	0	TTL expired

139

# Ping et Traceroute

- Ping
  - Dans un terminal « ping google.fr »
  - Permet de vérifier la joignabilité d'une machine identifiée par son adresse IP ou son URL
  - On envoie un « echo request »
  - La machine répond avec un « echo reply » contenant le temps de réponse
  - Remarque : il faut préciser le nombre d'envois (« ping -c2 google.fr » pour 2 envois par exemple) ou interrompre sinon cela ne s'arrête pas
- Traceroute
  - Dans un terminal « traceroute google.fr »
  - Donne les sauts IP pour joindre une destination identifiée par son adresse IP ou son URL
  - Principe : envoi d'un message avec TTL=1 pour déclencher un message d'erreur ICMP, puis TTL=2, TTL=3,...
  - Remarques :
    - Certains sauts ne répondent pas (configuration du routeur)
    - Certains sauts peuvent être différents en raison du partage de charge (load balancing)

140

# Table de routage

- Les tables de routage sont utilisées pour transférer les paquets IP dans la bonne direction de proche en proche
- On peut remplir une table de routage de manière :
  - Statique : routage par défaut configuré en DHCP
  - Dynamique : routage s'adapte aux changements du réseau
- Plusieurs protocoles de routage dynamique existent. Ils fonctionnent selon les mêmes principes :
  - Partage de l'information entre routeurs
  - En utilisant l'agrégation d'adresses (sans partager son plan d'adressage interne)
  - Convergence quand tous les routeurs ont les mêmes informations
- Un protocole de routage ne gère pas :
  - L'adressage
  - Le choix de la route par défaut
  - Les choix politiques

141

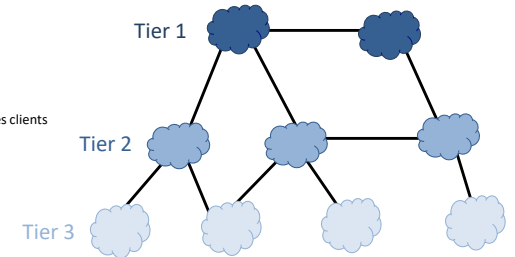
# AS

- Le routage est différent selon si on est à l'intérieur d'un AS ou entre 2 AS
- Routage inter-AS
  - Règle commune
  - BGP : Border Gateway Protocol
- Routage intra-AS
  - Règle interne choisie par l'AS
  - IGP : Interior Gateway Protocol
  - Il y a 2 types de protocoles IGP :
    - Distance Vector
    - Link State
- La table de routage d'un routeur est l'union des 2 tables issues des protocoles de routage inter et intra-AS

142

# BGP

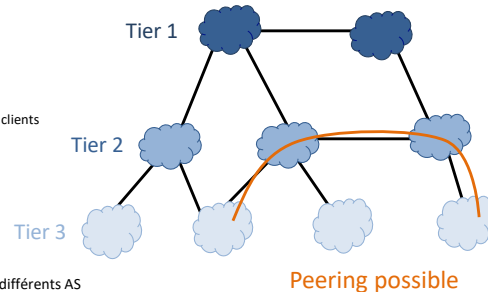
- Chaque AS annonce le meilleur chemin vers tous les sous-réseaux qu'il connaît
  - Choix d'affichages
  - « Meilleur » selon les critères de l'AS
- Relations entre AS
  - Relation client/fournisseur
    - Utilisation du lien payant pour le client
  - Relation pair à pair
    - Utilisation du lien gratuite si trafic bilatéral
    - Uniquement pour le trafic de ou vers ses propres clients
- Choix du meilleur chemin
  - Le moins cher
  - Le plus court en nombre d'AS
  - Le plus court en interne
- Protocole
  - eBGP : échanges BGP entre routeurs appartenant à différents AS
  - iBGP : échanges entre border routeurs d'un même AS (généralement en full mesh)
- Remarque : le paramétrage BGP est sensible car il a un impact global
  - Exemple : Pakistan Telecom a annoncé par erreur un plus court chemin vers Youtube. En 1 minute, les majorité des AS routaient leur trafic Youtube vers Pakistan Telecom. La résolution a pris une trentaine de minutes (annonce du bon chemin + convergence).



143

# BGP

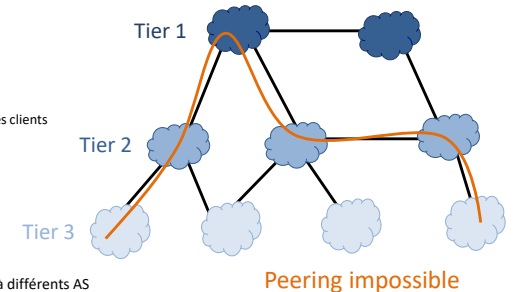
- Chaque AS annonce le meilleur chemin vers tous les sous-réseaux qu'il connaît
  - Choix d'affichages
  - « Meilleur » selon les critères de l'AS
- Relations entre AS
  - Relation client/fournisseur
    - Utilisation du lien payant pour le client
  - Relation pair à pair
    - Utilisation du lien gratuite si trafic bilatéral
    - Uniquement pour le trafic de ou vers ses propres clients
- Choix du meilleur chemin
  - Le moins cher
  - Le plus court en nombre d'AS
  - Le plus court en interne
- Protocole
  - eBGP : échanges BGP entre routeurs appartenant à différents AS
  - iBGP : échanges entre border routeurs d'un même AS (généralement en full mesh)
- Remarque : le paramétrage BGP est sensible car il a un impact global
  - Exemple : Pakistan Telecom a annoncé par erreur un plus court chemin vers Youtube. En 1 minute, les majorité des AS routaient leur trafic Youtube vers Pakistan Telecom. La résolution a pris une trentaine de minutes (annonce du bon chemin + convergence).



144

# BGP

- Chaque AS annonce le meilleur chemin vers tous les sous-réseaux qu'il connaît
  - Choix d'affichages
  - « Meilleur » selon les critères de l'AS
- Relations entre AS
  - Relation client/fournisseur
    - Utilisation du lien payant pour le client
  - Relation pair à pair
    - Utilisation du lien gratuite si trafic bilatéral
    - Uniquement pour le trafic de ou vers ses propres clients
- Choix du meilleur chemin
  - Le moins cher
  - Le plus court en nombre d'AS
  - Le plus court en interne
- Protocole
  - eBGP : échanges BGP entre routeurs appartenant à différents AS
  - iBGP : échanges entre border routeurs d'un même AS (généralement en full mesh)
- Remarque : le paramétrage BGP est sensible car il a un impact global
  - Exemple : Pakistan Telecom a annoncé par erreur un plus court chemin vers Youtube. En 1 minute, les majorité des AS routaient leur trafic Youtube vers Pakistan Telecom. La résolution a pris une trentaine de minutes (annonce du bon chemin + convergence).



145

# IGP – Distance Vector

- Protocoles Distance Vector
  - Basés sur une connaissance de la topologie locale
  - Un routeur annonce seulement ses meilleurs chemins (et pas tous ses chemins)
- RIP (Routing Information Protocol)
  - Basé sur l'algorithme de Bellman
    - Chaque nœud annonce uniquement ses meilleurs chemins, donc il n'y a pas de boucles
    - Chaque nœud calcule le min des distances sur ses antécédents
  - Fonctionnement :
    - Un routeur reçoit les plus courts chemins calculés par tous ses voisins (antécédents)
    - Il calcule ses propres plus courts chemins à partir de ces informations
    - Chaque routeur annonce ses plus courts chemins périodiquement
  - Attention : création de fausses boucles infinies en cas de perte de lien
    - Un lien tombe
    - Un routeur situé plus loin annonce un plus court chemin vers le lien tombé, les autres mettent à jour en passant par lui, puis lui incrémente en fonction de ses antécédents...
    - Solutions :
      - Définition d'un infini à 16
      - Split horizon : on n'annonce pas une route à celui qui nous la fournit
  - Avantage : simple, peu d'usage mémoire
  - Limitations : lent à converger, pour les petits réseaux ( $\leq 16$  liens), difficulté à repérer les boucles

146

# IGP – Link State

- Protocoles Link States
  - Basés sur une connaissance de la topologie complète du réseau (à l'intérieur de l'AS)
  - Un routeur annonce tous ses chemins
- OSPF (Open Shortest Path First)
  - Basé sur l'algorithme de Dijkstra
    - Tous les nœuds annoncent tous leurs chemins
    - Chaque nœud connaît tout le graphe et calcule l'arborescence des plus courts chemins depuis lui-même
  - Fonctionnement :
    - Un routeur reçoit tous les chemins de tout le monde
    - Il connaît tous les liens existant dans le réseau
    - Tous les routeurs ont le graphe en entier et appliquent l'algorithme de Dijkstra
    - Chaque routeur annonce ses tous ses chemins périodiquement
  - Avantages : rapide à converger, passe à l'échelle, pas de problèmes pour détecter des boucles ou en cas de perte de lien
  - Limitations : complexe à implémenter, lourd en mémoire

147

RES101  
Internet

## 4. SERVICES

148

# Application

- Une application au sens réseau c'est :
  - Une architecture
    - Plusieurs clients, un ou plusieurs serveurs, des pairs...
  - Un protocole
    - Couche 7 du modèle OSI
  - Un processus qui s'exécute
    - Agent, daemon
  - Des sockets
    - Pour communiquer avec les couches inférieures

149

# Architecture

- **Clients / serveur**

- Le serveur envoie le contenu à chaque client individuellement

- **CDN (Content Distributed Network)**

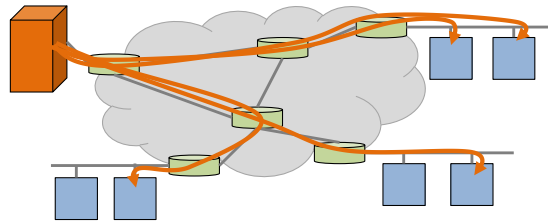
- Le serveur envoie le contenu à plusieurs nœuds intermédiaires
- Les clients se connectent à ces nœuds intermédiaires

- **Multicast IP**

- Le serveur utilise le multicast IP pour limiter le nombre de paquets envoyés pour un même contenu tant que la route est commune
- Le nombre de paquets différents augmente quand on se rapproche des clients et qu'il n'y a plus de route commune

- **P2P (Peer to peer)**

- Le serveur envoie le contenu à certains clients
- Les clients ayant le contenu deviennent serveur pour d'autres clients



150

# Architecture

- **Clients / serveur**

- Le serveur envoie le contenu à chaque client individuellement

- **CDN (Content Distributed Network)**

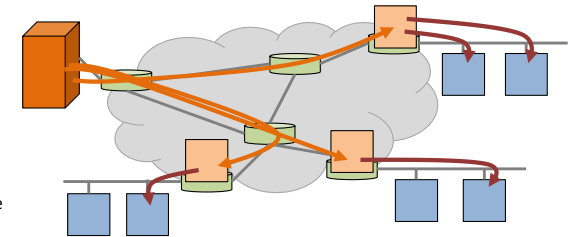
- Le serveur envoie le contenu à plusieurs nœuds intermédiaires
- Les clients se connectent à ces nœuds intermédiaires

- **Multicast IP**

- Le serveur utilise le multicast IP pour limiter le nombre de paquets envoyés pour un même contenu tant que la route est commune
- Le nombre de paquets différents augmente quand on se rapproche des clients et qu'il n'y a plus de route commune

- **P2P (Peer to peer)**

- Le serveur envoie le contenu à certains clients
- Les clients ayant le contenu deviennent serveur pour d'autres clients



151

# Architecture

- **Clients / serveur**

- Le serveur envoie le contenu à chaque client individuellement

- **CDN (Content Distributed Network)**

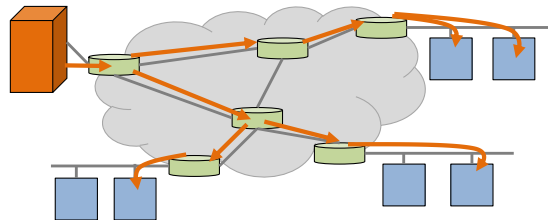
- Le serveur envoie le contenu à plusieurs nœuds intermédiaires
- Les clients se connectent à ces nœuds intermédiaires

- **Multicast IP**

- Le serveur utilise le multicast IP pour limiter le nombre de paquets envoyés pour un même contenu tant que la route est commune
- Le nombre de paquets différents augmente quand on se rapproche des clients et qu'il n'y a plus de route commune

- **P2P (Peer to peer)**

- Le serveur envoie le contenu à certains clients
- Les clients ayant le contenu deviennent serveur pour d'autres clients



152

# Architecture

- **Clients / serveur**

- Le serveur envoie le contenu à chaque client individuellement

- **CDN (Content Distributed Network)**

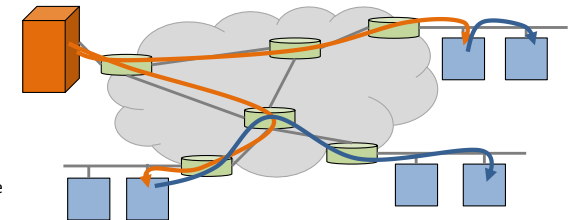
- Le serveur envoie le contenu à plusieurs nœuds intermédiaires
- Les clients se connectent à ces nœuds intermédiaires

- **Multicast IP**

- Le serveur utilise le multicast IP pour limiter le nombre de paquets envoyés pour un même contenu tant que la route est commune
- Le nombre de paquets différents augmente quand on se rapproche des clients et qu'il n'y a plus de route commune

- **P2P (Peer to peer)**

- Le serveur envoie le contenu à certains clients
- Les clients ayant le contenu deviennent serveur pour d'autres clients



153

# Architecture

- Paradigme lien logique/lien « physique »
  - Le lien logique est au niveau de l'application (couche 7 OSI)
  - Le lien bout-en-bout est au niveau IP (couche 3 OSI)
- Architecture network friendly
  - Une architecture est dite network friendly si elle implémente un protocole de niveau 4
  - L'application implémente alors un protocole de gestion de la transmission bout-en-bout
  - Cela permet de gérer les problèmes de contrôle d'erreur, de flux et de congestion bout-en-bout
- Architecture network aware
  - Une architecture est dite network aware si elle implémente un protocole de niveau 3
  - L'application implémente alors un protocole de routage bout-en-bout
  - Cela permet de contrôler les liens utilisés, par exemple de rester au sein d'un même ISP (Internet Service Provider, FAI Fournisseur d'Accès Internet en français)

154

# Protocole

- Règles de communication entre les processus
  - Type de message
  - Syntaxe
  - Format des messages
- Protocoles publics
  - IETF (HTTP, SMTP...)
  - Libre (BitTorrent...)
- Protocoles propriétaires
  - Skype
  - iMessage
  - Snapchat...

155

# Processus

- Processus Agent
  - Logiciel côté client
  - Gère l'IHM (Interface Homme-Machine)
  - Gère l'interface avec la couche inférieure (socket)
    - Ouverture de connexion niveau 4 si besoin
    - Port dynamique
- Processus Daemon
  - Logiciel côté serveur
  - Tourne en background en permanence
  - Interface avec la couche inférieure
    - Écoute les demandes d'ouverture de connexion si besoin
    - Port réservé

156

# Sockets

- La socket est l'interface avec la couche 4
  - Choix du protocole de couche 4
  - Identifiée par une adresse IP et un numéro de port
- Le choix du protocole de couche 4 est basé sur la QoS/QoE désirée
  - QoS : Quality of Service
    - Débit
    - Probabilité de perte
    - Délai
    - Gigue
  - QoE : Quality of Experience
    - Délai avant l'affichage, avant la synchronisation
    - Fiabilité (erreurs)
    - Qualité de l'image ou de la vidéo
  - Exemple :
    - Données : besoin de fiabilité => TCP
    - Voix : besoin de délai le plus faible possible => UDP

157

# Web

158

# Web

- Architecture
  - Client / serveur
  - Client : navigateur, crawler, spider
  - Serveur : serveur web (Apache), proxy
- Protocole
  - HTTP : HyperText Transfer Protocol
  - HTTP/1.0 : 1995
  - HTTP/2.0 : 2012
  - Message de type requête /réponse
- Socket
  - Fonctionne sur TCP
  - Port serveur 80

159

## HTTP/1.0 et HTTP/1.1

- 2 types de messages
  - Requêtes
    - GET : download
    - POST : upload
    - HEAD : download de l'en-tête uniquement
    - PUT : upload vers une URL donnée
    - DELETE : efface
    - + suivi de la liste des champs d'en-tête
  - Réponses
    - 1xx : information (exemple : 100 = pris en charge)
    - 2xx : succès (exemple : 200 = OK)
    - 3xx : redirection (exemples : 301 = moved permanently, 302 = moved temporarily)
    - 4xx : erreur client (exemples : 403 = forbidden, 404 = not found)
    - 5xx : erreur serveur (exemples : 503 = try again later)
- Format ASCII

160

## HTTP/1.0 et HTTP/1.1

Exemple de **requête** et **réponse** avec lignes d'en-tête :

```
GET /index.html HTTP/1.0
Host: www.telecom-paris.fr
Connection: close
User-agent: Mozilla
Accept: text/html, image/gif, image/jpeg
Accept-language: fr
```

```
HTTP/1.0 200 OK
Connection: close
Date: Thu, 06 Aug 2002 11:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-modified: Mon, 22 Apr 2001
Content-length: 621
Content-type: text/html
```

[DATA]

161



# HTTP

- HTTP/1.0
  - Type requête/réponse
  - Connexions TCP éphémères (1 connexion par objet)
  - Connexions possibles en parallèles
- HTTP/1.1
  - Type requête/réponse
  - Ajout des requête PUT et DELETE
  - Connexion TCP persistante (pour chercher tous les objets)
- HTTP/2.0
  - Streams (pas simple requête/réponse)
  - Priorités (ordre des réponses du serveur)
  - Compression des en-têtes répétitifs
  - Push des éléments avant les requêtes
  - Chiffrement SSL

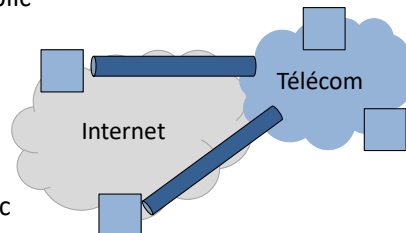
162

# VPN

163

# VPN

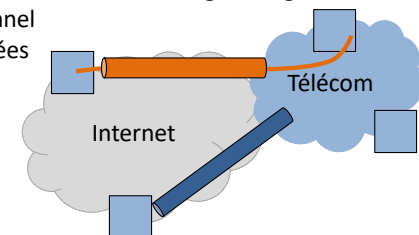
- Virtual Private Network
  - Système qui permet d'encapsuler et transmettre des informations d'un réseau privé en utilisant un réseau public
  - Accès virtuel à un réseau privé
- Usages
  - Connexion à distance (télétravail...)
  - Relier différents sites d'une entité
  - Protéger ses échanges du réseau public
- Ce que fait un réseau VPN :
  - Permet la confidentialité des informations échangées entre les machines distantes et le réseau privé
  - Permet l'accès à des ressources réservées au réseau privé
  - Permet d'avoir une adresse IP correspondant au réseau privé



164

# Tunnel

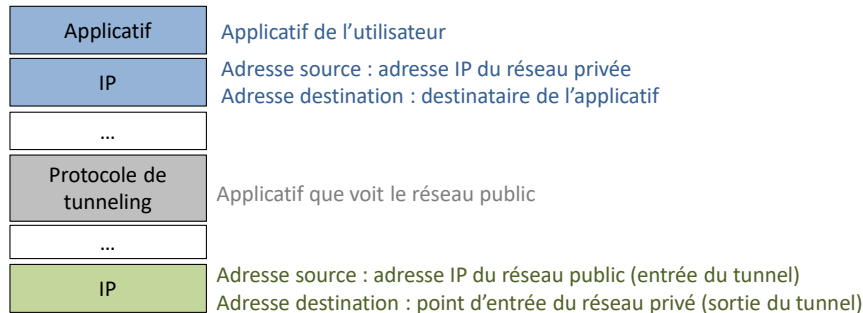
- Toutes les données échangées entre la station distante et le réseau privé passent à travers un tunnel sur le réseau public
  - Le tunnel est créé lors de la connexion au réseau VPN après authentification
  - Pour créer le tunnel, il y a nécessairement échange de signalisation
  - Plan contrôle : gestion du tunnel
  - Plan usager : données envoyées dans le tunnel
- L'usage du tunnel permet :
  - Le chiffrement bout en bout
  - Le routage (choix du chemin) une seule fois pour toute la communication
  - L'intégrité des données transportées à l'intérieur du tunnel
- Du point de vue du réseau public, seul le tunnel est visible (mais pas ce qu'il y a dedans)



165

# Tunnel

- Un tunnel permet donc de
  - Encapsuler du trafic de niveau 2 (PPP) ou de niveau 3 (IP)
  - Au niveau 3 (dans IP), 4 (dans TCP ou UDP) ou supérieur (dans HTTPS)
- La pile protocolaire d'un utilisateur comprend alors 2 couches IP
  - 1 couche IP dans le plan d'adressage du réseau privée
  - 1 couche IP dans le plan d'adressage du réseau sur lequel le paquet transite
  - C'est comme une enveloppe enfermée dans une autre enveloppe



166

# Protocoles

Il existe différents protocoles qui peuvent être utilisés pour créer un VPN (le VPN est alors une nouvelle connexion de l'ordinateur), dont :

- PPTP (Point-to-Point Tunneling Protocol)
  - Encapsule des trames de niveau 2 (PPP) dans TCP
  - Dépend de PPP pour l'authentification et le chiffrement
  - Développé par Microsoft, présent sur Windows
  - Obsolète
- L2TP (Layer 2 Tunneling Protocol)
  - Encapsulation de trames de niveau 2 dans UDP/IP
  - Authentification et chiffrement possible pour les trames encapsulées ou pour le tunnel sur IPSEC
  - Combinaison de L2F (Cisco) et PPTP
- IPsec (Internet Protocol security)
  - Encapsulation d'un paquet IP dans un autre paquet IP
  - Authentification et chiffrement sur IP
- SSL/TLS (Secure Sockets Layer/Transport Layer Security)
  - Protocole d'authentification, chiffrement, et protection d'intégrité
  - Utilisé dans HTTPS
- SSH (Secure Shell)
  - Protocole de communication chiffrée
  - Permet d'ouvrir un terminal sur un ordinateur distant
- MPLS (Multiprotocol Label Switching)
  - Réseau à commutation de labels (étiquettes)

167

# OpenVPN

- Logiciel permettant de créer et utiliser un réseau VPN (le VPN est alors une application)
  - Client et serveur
  - Logiciel libre
- Utilise
  - La bibliothèque OpenSSL
  - Le protocole SSL/TLS
- VPN de Télécom Paris

168

IC203

## TRAVAUX DIRIGES N° 1

### 2. Exemple de scénarios

#### 2.1 Introduction des variables V(S) et V(R)

La recommandation X25 (dans la partie qui spécifie le protocole LAPB) définit deux variables internes V(S) et V(R) dans chacun des deux équipements impliqués dans le dialogue. V(S) et V(R) peuvent être considérées comme des variables entières définies dans le logiciel gérant le protocole et évoluant au cours du temps suivant les différents événements qui peuvent arriver.

Elles sont définies comme suit :

- V(S) est le numéro de séquence de la prochaine trame devant être émise,
- V(R) est le numéro de la prochaine trame attendue.

La définition de telles variables permet de décrire de façon simple les processus d'émission et de réception.

#### 2.2 Hypothèses des scénarios

Au début des scénarios considérés, A et B sont au repos après avoir engagé avec succès une procédure de connexion. Aucune temporisation n'est en cours.

Lorsqu'un équipement reçoit une trame correcte et qu'il n'a aucune trame d'information à envoyer, il acquitte la trame reçue immédiatement par une trame de supervision.

On suppose que les équipements (A et B) ont une fenêtre d'anticipation maximale de 4 trames.

Dans les 3 scénarios suivants, il est demandé de compléter les dialogues (si nécessaire), d'indiquer les types de trames, leurs numéros et de donner la valeur des compteurs V(S) et V(R) pour chacun des équipements.

#### 2.3 Dialogue duplex sans erreur

On considère un dialogue duplex sans aucune erreur de transmission dans le scénario 1.  
a. Compléter le schéma illustrant le dialogue en supposant que A désire terminer la connexion lorsque l'ensemble des trames a été acquitté. Représenter le nombre de trames émises non acquittées à chaque instant.

b. On suppose maintenant que les équipements ont une fenêtre d'anticipation de taille 3 et qu'aucune temporisation n'arrive à échéance. Comment le dialogue est-il modifié ? Même question si la fenêtre n'est plus que de taille 2. On se limitera au début du dialogue.

#### 2.4 Dialogue avec erreur sur une trame

Dans les scénarios 2 A désire transmettre 3 trames d'information à B. Ce dernier n'a rien à transmettre et se contente de réagir à ce qu'il reçoit conformément au protocole.

Dans le scénario 2, la première trame est bien reçue par B et l'acquiescement de B est également bien reçu. La deuxième trame émise par A est brouillée sur la ligne ; elle est reçue