Le Réseau

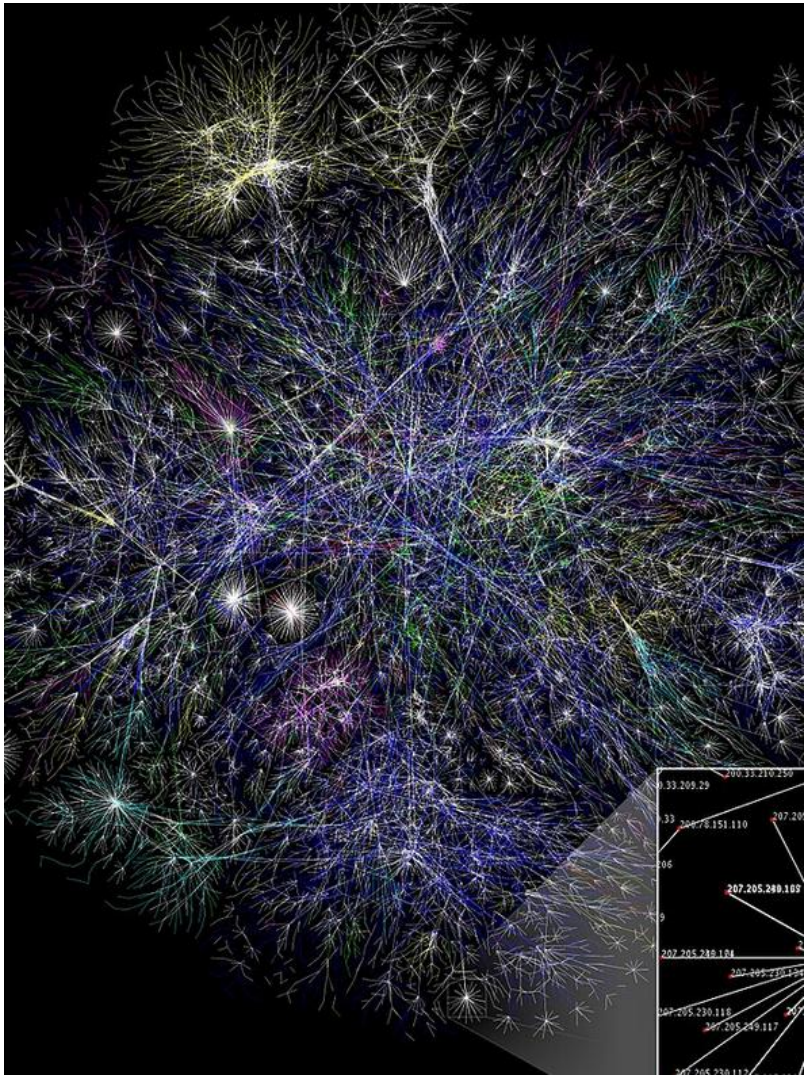
Rappel sur les couches hautes

Les couches basses

Internet

P. Laurençot

Internet (1)



- Liaison En 2005

Internet (2)

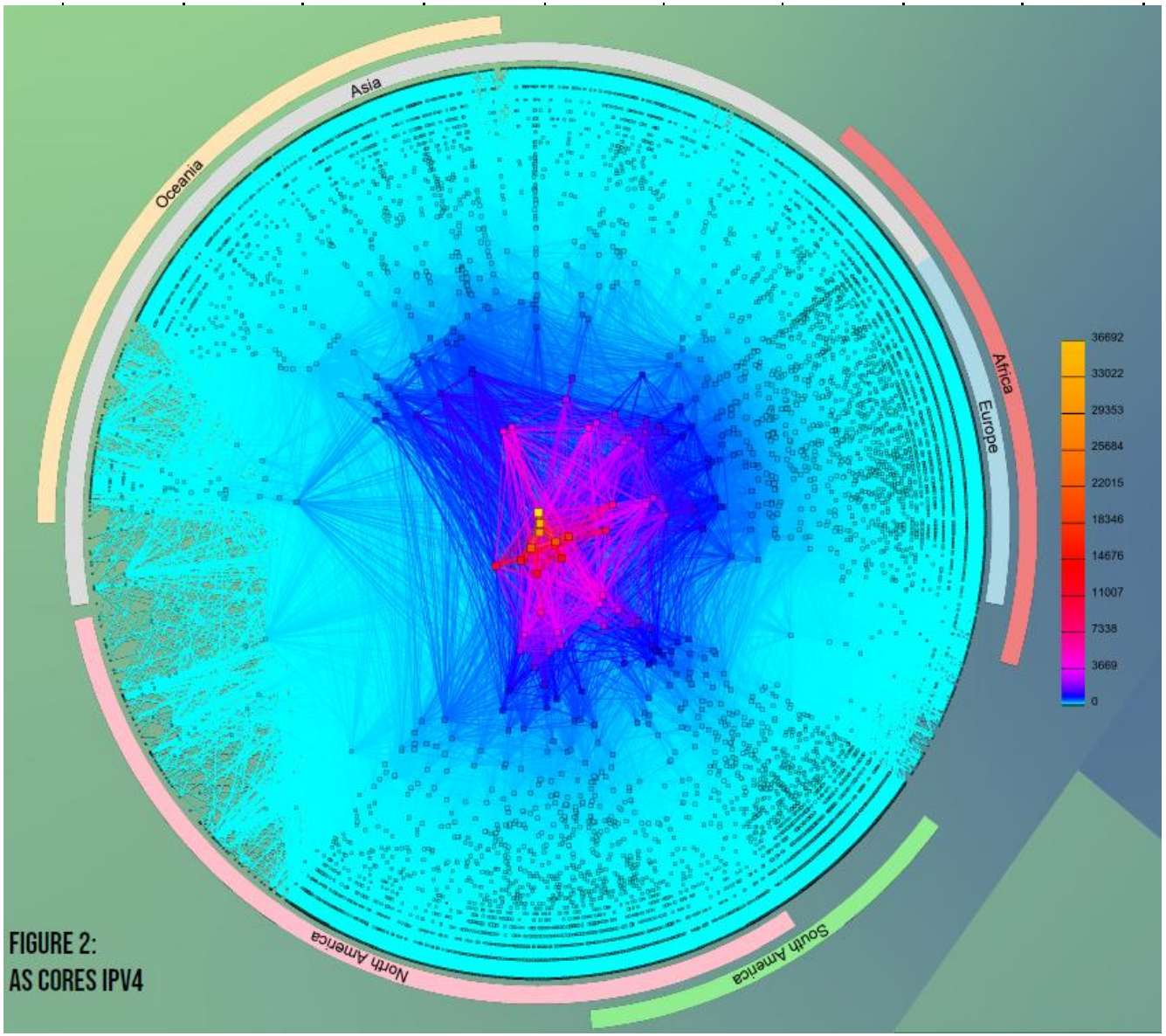


FIGURE 2:
AS CORES IPV4

Caida's ipv4
As Core

Quelques rappels

Les couches de l'OSI (1)

7	Application *Application*	échanges de données d'application (selon l'application)
6	Présentation *Presentation*	mise en forme des données pour la transmission
5	Session *Session*	synchronisation de la communication entre processus
4	Transport *Transport*	transfert de blocs d'octets entre processus
3	Réseau *Network*	transfert de blocs d'octets entre systèmes distants (pas forcément sur le même réseau physique)
2	Liaison de données *Data Link*	transfert fiable de blocs d'octets entre systèmes situé sur le même réseau physique
1	Physique *Physical*	transfert physique de bits entre systèmes raccordés au même médium

Les couches de l'OSI (1 bis)

7	Application *Application*	échanges de données d'application (selon l'application)
6	Présentation *Presentation*	mise en forme des données pour la transmission
5	Session *Session*	synchronisation de la communication entre processus
4	Transport *Transport*	transfert de blocs d'octets entre processus
3	Réseau *Network*	transfert de blocs d'octets entre systèmes distants (pas forcément sur le même réseau physique)
LLC	Contrôle de lien logique *Logical Link Control*	Couche de transition entre la couche MAC et la couche 3
MAC	Contrôle d'accès au médium *Medium Access Control*	transfert de bits entre systèmes raccordés au même support physique , avec contrôle du droit d'émission
1	Physique *Physical*	transfert physique de bits entre systèmes raccordés au même médium

Les couches de TCP/IP

Application	échanges de données d'application (selon l'application) mise en forme de données échangées synchronisation de processus distants
Transport	transfert de blocs d'octets entre processus
Internet	transfert de blocs d'octets entre systèmes distants (pas forcément raccordés au même médium)

Accès Réseau	transfert fiable de blocs d'octets entre systèmes raccordés sur le même réseau et transmission physique

La couche 1 de l'OSI

Couche Physique

- Description des moyens physiques de transmission (spécification des connecteurs, des tensions,…)
- Transmission de blocs de bits codés en signaux physiques sur un médium
- Notion de bande passante → débit (b/s)

débit idéal > débit réel

car
- topologie du réseau
- nb d'utilisateurs sur le réseau
- équipements d'inter-réseau
- conditions d'alimentation, …

- *Equipement utilisé* : Répéteur, concentrateur actif ou passif (HUB), modem (téléphonique, DSL, câble), cartes réseaux

La couche 2 de l'OSI

Couche Liaison de Données

- Transfert de blocs d'octets entre 2 systèmes raccordés **au même réseau,** dans des **trames**

- Transfert dans le bon ordre et sans multiplication
- Détection/ Correction d'erreurs
- Régulation de flux

- *Equipement utilisé* : Commutateur (Switch)

  Carte NIC - Network Interface Card – (carte réseau)

  - choix selon :
    - type de média connecté (cuivre, fibre optique, sans fil)
    - protocole utilisé
    - type de bus interne au système (ISA, PCI,PCI express…)

La couche 3 et 4 de l'OSI

Couche Réseau

- Transfert de blocs **d'octets entre réseaux différents**, par l'intermédiaire de routeurs, dans des **paquets**

- Calcul de route(s)
- Contrôle de flux, gestion des ressources du réseau
- Problème d'interconnexion des réseaux
- Appellation (adressage logique)

- *Equipement utilisé* : routeur

Couche Transport

- Transmission de blocs d'octets entre processus, dans des **segments**
- Contrôle des pertes et duplications de segments, ainsi que de l'ordre de livraison des blocs d'octets

Les couches 5, 6 et 7 de l'OSI

Couche Session

- Organisation des échanges (qui parle en premier ?)
- Définition de points de reprise

Couche Présentation

- Mise en forme des données d'application, pour tenir compte des différences de codage des systèmes
- Chiffrement (si nécessaire)

Couche Application

- Services aux utilisateurs : Web, messagerie, jeux, …
- Supervision de processus industriels
- Supervision du réseau, des ordinateurs, ..
- Télé-travail,….

La couche applicative

Généralité

Notion client/serveur

- Serveur : en attente de connexion → un port ouvert
- Client : veut une information détenue par le serveur
  - une requête à effectuer avec la bonne syntaxe

Client

- processus créé lorsque le besoin apparaît
  - par une personne
  - au démarrage du système
- l'entité-client doit savoir à quelle entité-serveur elle doit/peut s'adresser
  - au lancement du processus : fichier de configuration, argument
  - par saisie pendant l'exécution

Généralité - Côté serveur

- Les entités serveur
  - fonctionnement permanent (démons)
    - processus créé au démarrage du système
  - processus lancé lorsqu'une demande arrive
    - par un processus "écouteur"
- Méthodes de prise en charge d'une demande de Service
  - L'entité serveur peut traiter elle-même la demande
    - Mise en attente des autres demandes jusqu'à la fin du traitement
  - L'entité serveur peut déléguer à un processus fils (ou un thread) le traitement de la demande
    - Contrôle du nombre de fils (ou de thread) ->nombre maximum donné, pré-création …
    - Risque de surcharge de la mémoire

Applicatifs

- Quels applicatifs connaissez-vous et à quoi servent-ils ?

HTTP, HTTPS, DHCP, SMTP, POP, IMAP, DNS,FTP, SSH, SFTP, FTPS, NTP, RTP, …

La couche transport

TCP / UDP

TCP
Service en mode connecté

- jusqu'à 65534 points d'accès par système
- détection et tentative de correction des pertes et duplications
- signalement des erreurs
- livraison dans l'ordre d'émission
- contrôle de flux
- Service fiable

UDP
Service en mode non connecté

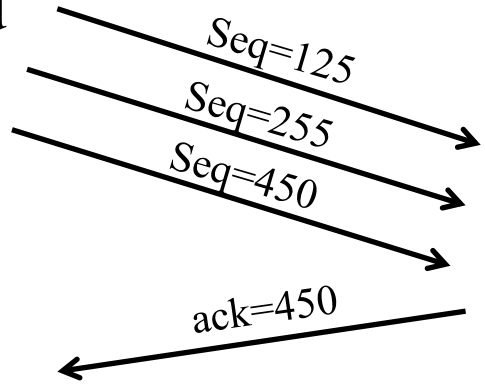
- jusqu'à 65534 points d'accès par système
- un ou plusieurs destinataires
- Service "non fiable"

Fiabilité de TCP (1)

- Numérotation lors de l'expédition
  - les octets de l'utilisateur sont transmis par bloc dans des PDU-TCP
  - l'entité-TCP calcule un numéro de séquence qui correspond "aux nombres d'octets envoyés" et le place dans la PDU envoyé, cela correspond au numéro seq : séquence
  - initialisation de seq : à l'établissement de connexion
  - Calcul lors de la phase d'initialisation d'un nombre correspondant "aux nombres d'octets reçus" (aussi envoyé dans la PDU), c'est le numéro ack : acquittement
- Vérification lors de la réception
  - n° reçu > n° attendu → une perte
  - n° reçu < n° attendu → une duplication
  - n° reçu = n° attendu → OK
- Correction
  - *duplication* : l'entité-TCP ignore la PDU-TCP en doublon
  - *perte* : retransmission de la PDU contenant le bloc
    - on renvoie la partie perdue

Fiabilité de TCP (2)

- Différence paquet perdu /paquet en retard
  - TCP peut acquitter plusieurs segments d'un coup
    (gain en bande passante)
  - Mise en place de **timers** (temporisation)
    - A chaque émission, création d'un timer
      - Si ack avant fin timer ⇒ OK
        sinon segment perdu



- Calcul du timer
  - Si timer trop petit → beaucoup de retransmission
  - Si timer trop grand → attente longue pour détecter une perte
  - Re-Calcul du timer de temps en temps
    - basé sur le temps pour un échantillon, le délai moyen, la variance, etc…
      (timer >> RTT)

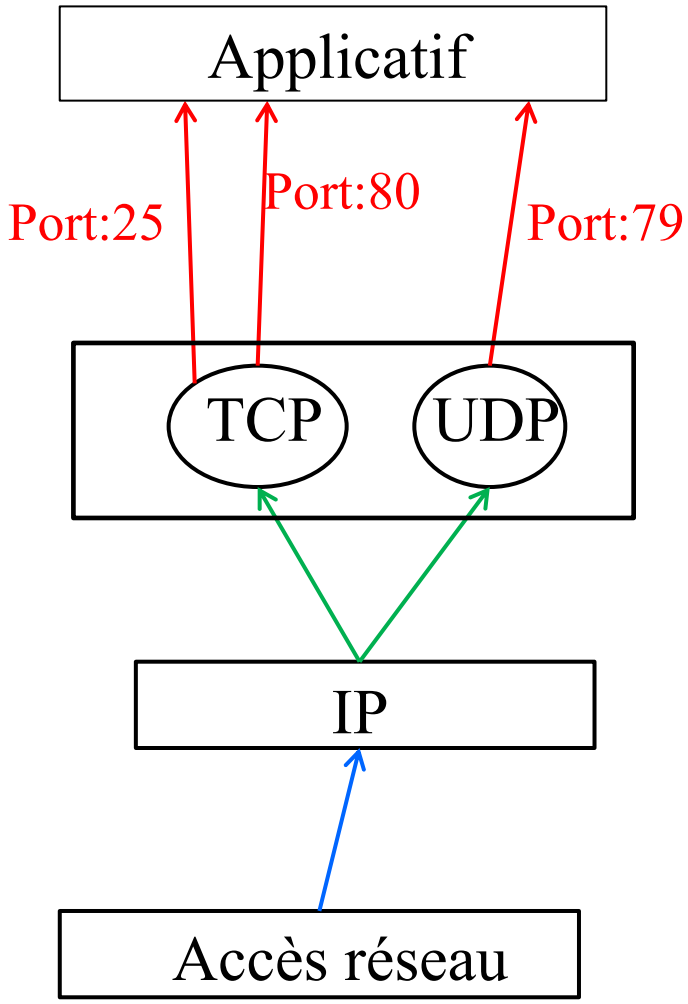
Protocole TCP - contrôle de flux

- Une entité-TCP place les octets reçus pour l'utilisateur dans des files d'attente où il peut en prendre livraison
  - place dans la file d'attente = capacité de réception
  - file d'attente pleine
    ⇨ impossible d'ajouter de nouveaux octets reçus
    et consommation inutile de ressources de (re)transmission

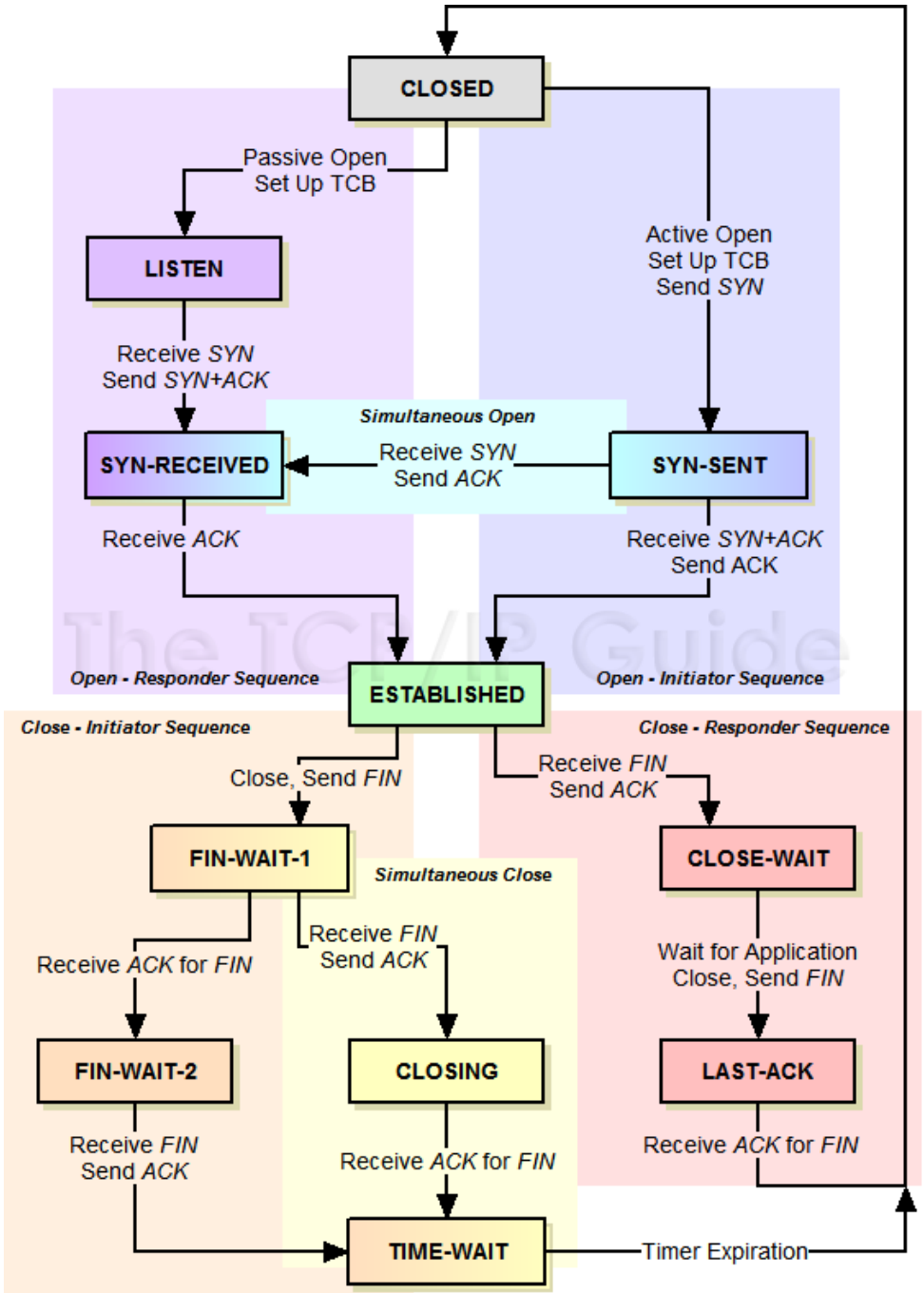
- Le contrôle de flux permet d'éviter à l'entité-TCP expéditrice d'envoyer plus que l'entité-TCP réceptrice peut mémoriser dans les files d'attente
  - chaque entité-TCP place dans chaque PDU qu'elle envoie le nb d'octets libres dans la file d'attente (champ win – tcp window size)

Connexion TCP/IP

- Identifier un processus
  - 3 informations obligatoires :
    - L'adresse IP
    - Le protocole utilisé
    - Le numéro de port



L'accès réseau :
- Non filaire : Wifi
- Filaire : ethernet



Etat d'une connexion TCP

netstat -an

État : listen(ing)
established

Contrôle de congestion

- But : utiliser toute la bande passante du câble

- Au départ, valeur bande passante est inconnue

  -> différents algorithmes existent (New Reno, vegas, Westwood+..)

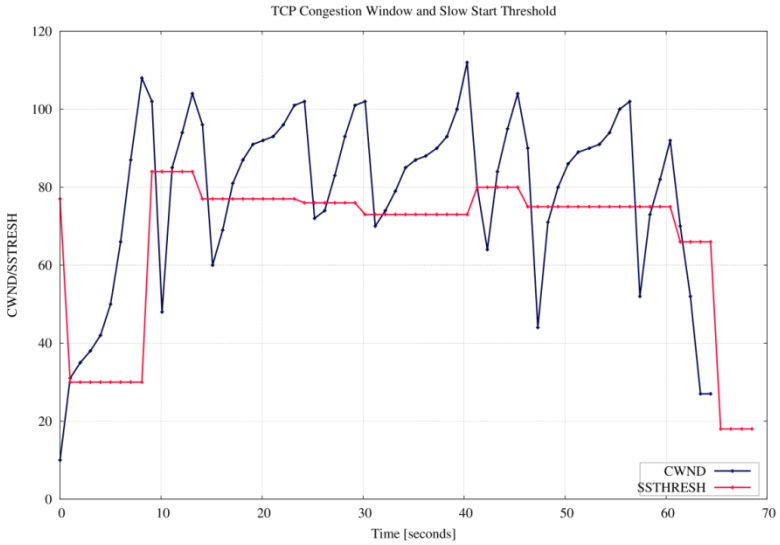
- **Méthode**

  - on envoie de plus en plus de segments, sans attendre d'acquittement
  - si perte segment, détection via les valeurs de seq/ack

    -> on diminue le nombre de segments sans acquittement

    (soit de moitié, soit d'un pourcentage)

  puis on ré-augmente progressivement



La Couche Réseau (OSI)
ou
Internet (TCP/IP)

Couche 3 - Généralité

- L'unité d'information est le **paquet**

- Les fonctionnalités principales de la couche 3
  - *Mode de connexion*
    - 2 modes existent : connecté (ATM, X25,…), non connecté (IP,….)
  - *Désignation des systèmes* (adressage)
    - Unicité des adresses si possible
  - ***Calcul des routes***
    - Trouver un chemin entre la source et la destination
    - Création des tables de routage

IPv4 (1)

- Caractéristiques
  - Destinataire unique ou multiple
  - Mode non connecté

Adresse IP pour chaque carte réseau, pas par hôte

➢ Adresse IPv4 : 32 bits –> 4*8 bits

Pour simplifier :

4 nombres : 0-255.0-255.0-255.0-255

exemple : 193.55.95.26, 43.10.15.2, 192.156.20.254

Max = 2^32= 4,2 Milliards ….

IPv4 (2)

- **@IP : Partie réseau + numéro unique sur le réseau**

Unique au niveau mondial

- *Partie réseau* : donnée par des instances internationales (RIR – Regional Internet Registry)
- *Partie hôte :* donnée par les responsables du site (unicité)

- A partir d'une @IP, où se trouve la partie réseau ?
  - **Utilisation du masque**
  - **Le masque est donné avec l'@IP**
    - Faire un "et" bit à bit pour trouver la partie réseau

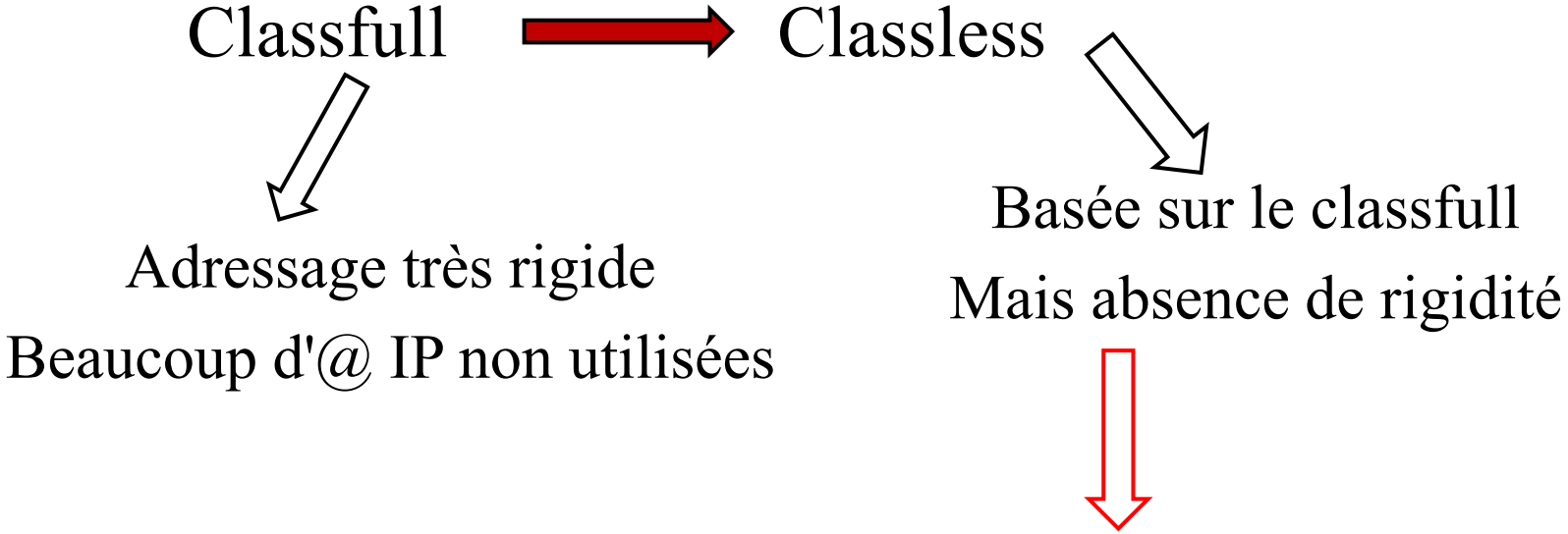
Exemple : @IP : 193.55.95.32 masque : 255.255.255.0

réseau : 193.55.95.0

IPv4 (3)

- Pour chaque réseau, 2 adresses réservées :
  - Adresse réseau : tous les bits hôtes à 0,
  - Adresse broadcast : tous les bits hôtes à 1

- Evolution de l'adressage réseau



IPv4 (4)

Par défaut,…

- Si 1er octet entre 0 et 127, masque 255.0.0.0 (classe A)
  - Seul 1er octet réseau et 3 octets pour la partie hôte
- Si 1er octet entre 128 et 191, masque 255.255.0.0 (classe B)
  - Les 2 premiers octets pour le réseau et 2 octets pour la partie hôte
- Si 1er octet entre 192 et 223, masque 255.255.255.0 (classe C)
  - Les 3 premiers octets pour le réseau et 1 octet pour la partie hôte

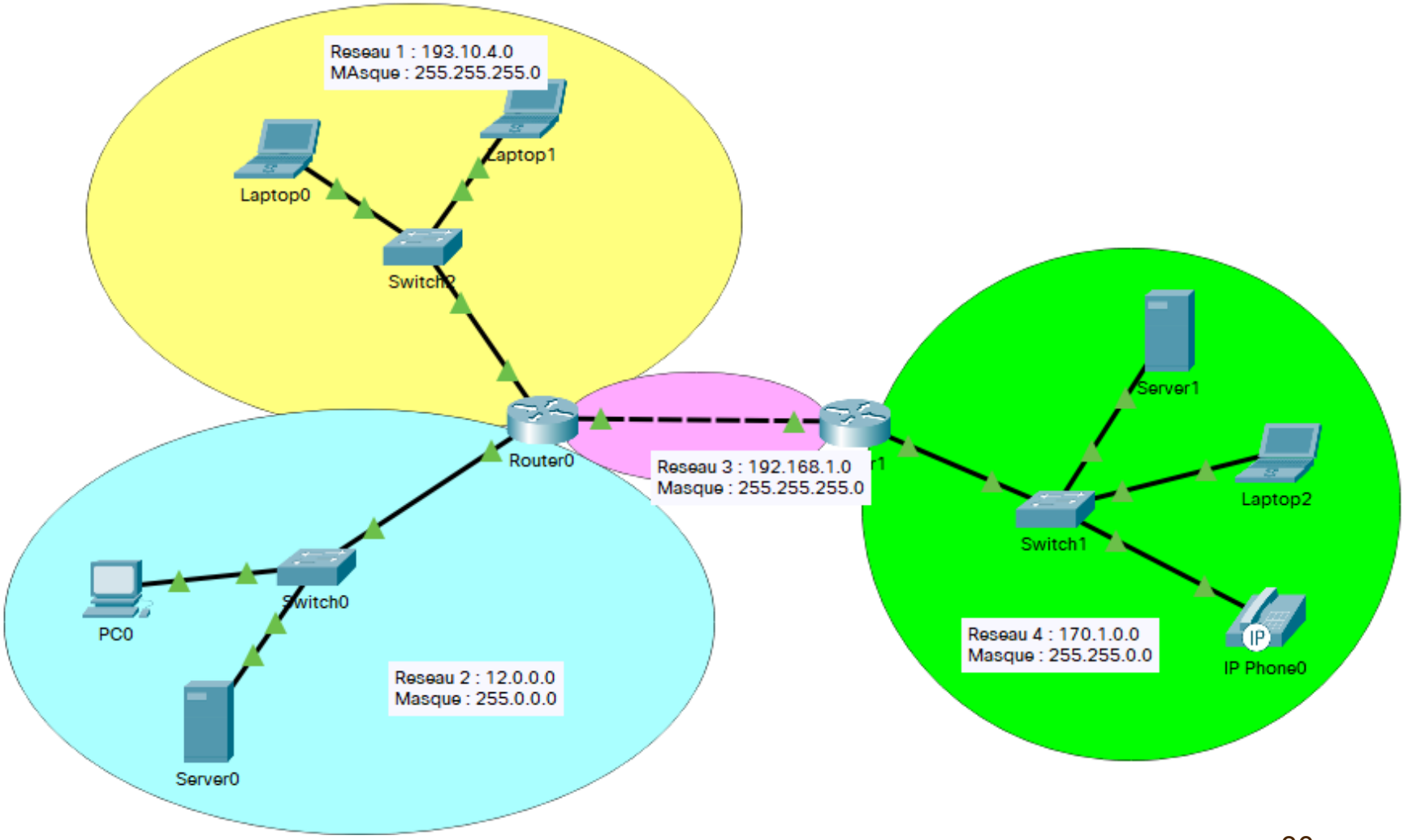
Une classe C ne peut contenir que 254 machines.

Exemple : @réseau : 200.10.15.0 @machines : 200.10.15.1

@broadcast : 200.10.15.255 ….. 200.10.15.254

Architecture

- **Une adresse réseau différente par réseau**



IPv4 (5)

- Les sous-réseaux (ou réseaux…)
  - Objectif : prendre un réseau de classe A, B ou C et le sous-divisé en plus petits …
  - Différentes étapes :
    - Trouvez la classe du réseau initial
    - Créer des sous-réseaux en jouant sur le masque
      - **Puissance de 2**
      - A chaque sous-réseaux, réservation de 2 adresses
        - Adresse réseau
        - Adresse broadcast

Masque : 11111111 . 11111111 . 11111 000 . 00000000

Réseau Sous-réseau Hôte

Les sous-réseaux (1)

Calcul du masque du sous-réseau

- augmentation du masque du réseau de 1 bit par puissance de 2

Exemple pour un classe C:

Nb de sous-réseaux	Nb d'hôtes	Masque
2	126	255.255.255.128
4	62	255.255.255.192
8	30	255.255.255.224

Adresses IP (exemple)

masque réseau	Nb sous réseaux	Réseau	Diffusion	minIP	MaxIP	Nb hotes
128	**2**	**0**	**127**	**1**	**126**	**126**
		128	**255**	**129**	**254**	**126**
192	**4**	**0**	**63**	**1**	**62**	**62**
		64	**127**	**65**	**126**	**62**
		128	**191**	**129**	**190**	**62**
		192	**255**	**193**	**254**	**62**

Les sous-réseaux (2)

Calculer un sous-réseau : masque

- trouver la puissance de 2 **supérieure ou égale** au besoin
- calcul du masque du sous-réseau

Ex : classe B, 20 sous-réseaux → 32 = 2^5

255. 255 . | xxxxx | xxx . xxxxxxxx

d'où 255.255. 248 . 0

Trouvez les sous-réseaux

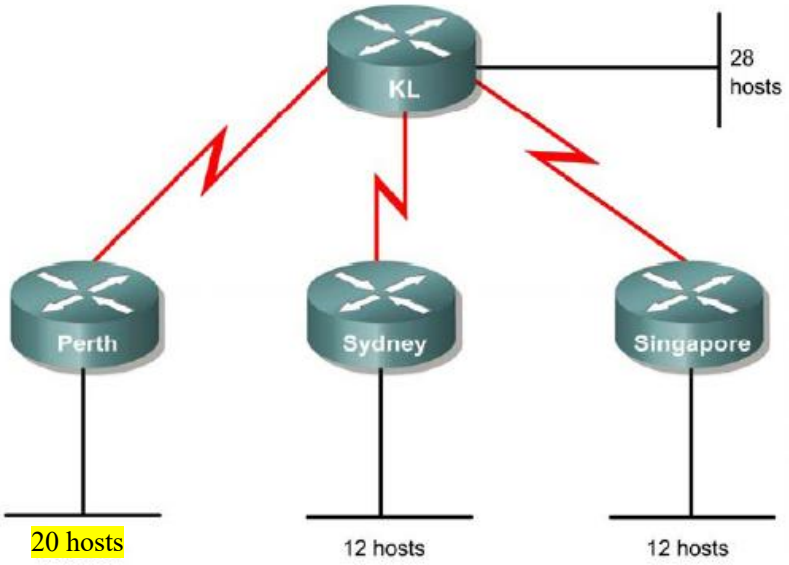
- Prendre l'adresse réseau ex : 156.10.0.0
- Faire varier les bits utilisés pour le sous-réseau pour obtenir les adresses des sous-réseaux

Ex : 156.10. 00000 000.0 → 156.10.0.0
156.10. 00001 000.0 → 156.10.8.0
156.10. 00010 000.0 → 156.10.16.0 etc…

VLSM (1)

- Variable Length Subnet Mask
  - Minimiser encore plus la perte des @IP
  - Sous-diviser un sous-réseau…
  - OSPF, IS-IS, EIGRP, RIPv2 supportent le VLSM

- Exemple



Réseau : 200.1.1.0/24

7 réseaux différents:

200.1.1.
{0,32,64,96,…}/27

30 machines max -> OK

VLSM (2)

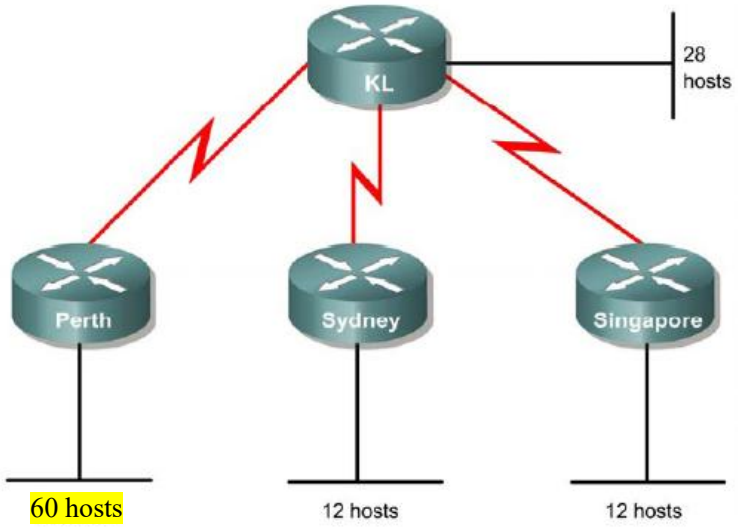
- Problème

➔ 7 réseaux

2 possibilités :

8 sous-réseaux, mais 32 machines max

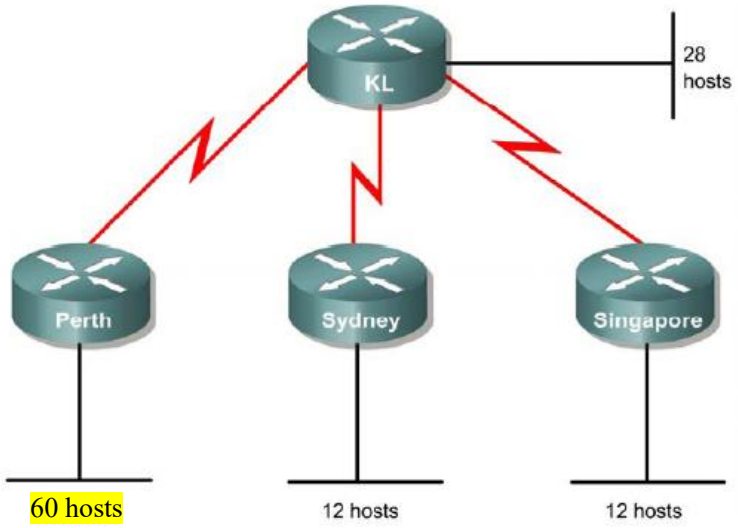
4 sous-réseaux, avec 64 machines max



- On sous-divise des réseaux
  - **On trie les réseaux du plus grand vers le plus petit**
    60, 28, 12, 12, 2, 2, 2
  - 200.1.1.0/24
    - 200.1.1.0/26 ← pour réseau Perth
    - 200.1.1.64/26 libre → à diviser pour les autres réseaux
    - 200.1.1.128/26 libre
    - 200.1.1.192/26 libre

VLSM (3)

- 200.1.1.0/24
  - 200.1.1.0/26 ←Perth
  - 200.1.1.64/26 libre
  - 200.1.1.128/26 libre
  - 200.1.1.192/26 libre



- 200.1.1.64/26
  - 200.1.1.64/27 ← KL
  - 200.1.1.96/27 libre
  - 200.1.1.128/26 libre

- 200.1.1.96/27
  - 200.1.1.96/28 ← Sydney
  - 200.1.1.112/28 ← Singapore

- 200.1.1.128/26
  - 200.1.1.128/30 200.1.1.132/30 200.1.1.136/30

Résumé de route (1)

- Résumé ou agrégation de route
  - Inverse du sous-réseau
  - Faire des supers masques (CIDR)

- Exemple
  - 172.16.64.0 = **10101100.00010000.01000**00.00000000
  - 172.16.65.0 = **10101100.00010000.01000**01.00000000
  - 172.16.66.0 = **10101100.00010000.01000**10.00000000
  - 172.16.67.0 = **10101100.00010000.01000**11.00000000
  - Bits communs : **10101100.00010000.010000**xx

On annonce alors le réseau : 172.16.64.0/22 ou 255.255.252.0

Résumé de route (2)

- Exercice
  - Quel résumé de route faire pour :
    - 172.16.74.0
    - 172.16.75.0
    - 172.16.76.0
    - 172.16.77.0

IPv4 (6)

- Adresses IP privées (RFC 1918)
  - Non unique, mais gratuite
  - Ne permettent pas d'aller sur internet
  - @réseau : 10.0.0.0 /8
    masque 255.0.0.0
  - @réseau : 172.16.0.0- 172.31.0.0 /16
    masque 255.255.0.0
  - @réseau : 192.168.0.0 – 192.168.255.0 /24
    masque 255.255.255.0

Problème adresses IPv4

Raisons:

- Croissance rapide d'internet ➡ manque d'adresses IPv4 (seulement sur 4 octets)
- Tables de routage gigantesques…
- Des fonctionnalités manquantes à mettre en œuvre

Migration vers IPv6:

- **Adresse sur 16 octets (128 bits, 8 * 16 bits)**
- Fonctionnalités nouvelles :
  - sécurité (chiffrement des paquets, authentification,…)
  - source routing
  - Gestion du temps réel
  - autoconfiguration (amélioration de DHCP, passage par ICMPv6)

IPv6 (1)

- Les différentes sortes d'adresses
  - **Unicast**
    -> pour un destinataire unique
  - **Multicast**
    -> pour désigner un groupe d'interfaces, donc un groupe de machines
  - **Anycast**
    -> pour désigner une interface, appartenant à un groupe de machines
    -> théoriquement, la machine la plus proche doit recevoir le paquet
  - **Broadcast**
    -> n'existe plus.

Adressage IPv6 (1)

- Utilisation de la notation hexadécimale, en regroupant les chiffres par 4 et en les séparant par ‘:’
  ex : FEDC:E323:A65A:95F5:63D4:08BB:76F5:A234

- Disparition des sous-réseaux, apparition de la taille du préfixe ‘/’
  (nb de bits appartenant au préfixe réseau)
  ex : 2F45:EE34:C23E::/48

- Elimination des 0 présents dans l’adresse:
  ex : 2001:0:0:0:0:342D:342F:FF45

2001::342D:342F:FF45

Adressage IPv6 (3)

Adresse Unicast

- 64 bits pour le réseau
  - 48 bits pour le préfixe global de routage (topologie publique)

    → préfixe 2000::/3 théoriquement, 3 premiers bits à : 001

    → 45 bits suivants dépendent de l'IANA, et donc RIPE-NCC
  - 16 bits pour le réseau d'entreprise (topologie du site)
- 64 bits pour l'hôte
  - 64 bits -> interface (unicast -> utilisation de icmpv6)

L'adresse commence en général par 2001:…. et se termine par l'adresse MAC modifiée:

3 octets (constructeur) + FFFE (16 bits) + 3 octets (unique MAC)

Adressage IPv6 (4)

Adresse Multicast

- indicateur (8 b) + drapeau (4 b) + visibilité (4 b) + group Id (112 b)
  - Indicateur : FF (1 octet avec que des 1)
  - Drapeau : 000 puis 0 pour un groupe permanent,
    1 sinon
  - Visibilité : sur internet, local à l'entreprise…
    - Node – Local (même médium) → visibilité =1
    - Link – Local (même domaine broadcast) → visibilité =2
    - Site – Local (même site) → visibilité =5
    - Organization – Local (même entreprise) → visibilité =8
    - Global → visibilité =E
- Group id : 1 = machines, 2 = routeurs, 4 = OSPF routers, ….
- Exemple : *FF01::1 -> multicast pour node local et machines (id =1)*
  FF05::2 -> multicast pour site-local et routers (id =2)

Adresse lien local IPv6

- Adresse valide seulement sur un lien (elle ne traverse pas les routeurs)
- Fonctionnement identique à 169.254.0.0/16 en Ipv4
- Adresse donnée automatiquement à chaque système ce qui permet de découvrir les voisins.
  - Pour chaque interface, une adresse lien local.
- Utilisation du préfixe : FE80::/10 (voir /64 généralement)
- 64 bits pour l'interface en eui-64

Exemple : FE80::20C:29FF:FEF1:B52F / 64