

Bases des réseaux IP

Interconnexion de réseaux

Benoit Lourdelet
Cisco Systems
blourdel@cisco.com

Interconnexion de réseaux

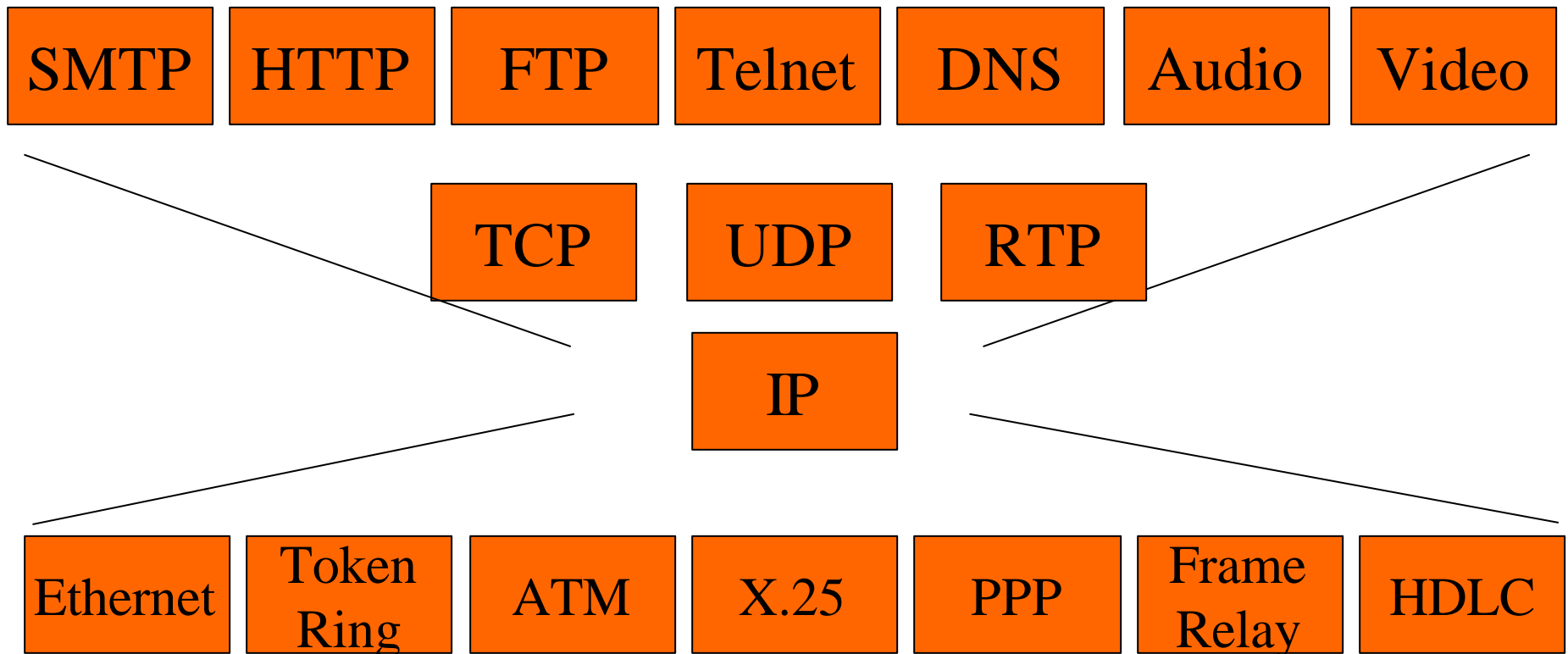
- ⌘ Plusieurs petits réseaux
- ⌘ Supports différents:
 - ☒ ethernet, lignes spécialisées, RTC, ATM, Frame Relay, FDDI
- ⌘ Chaque support à ses propres règles d'adressages et ses protocoles
- ⌘ Comment les relier entre eux et fournir une vue unifiée du tout ?

Unification par la couche réseau

- ⌘ Définir un protocole qui fonctionne de la même manière quel que soit le support sous-jacent
- ⌘ Appelons celle-ci la couche réseau
- ⌘ Les routeurs IP fonctionnent à la couche réseau
- ⌘ Des méthodes sont définies pour utiliser:
 - ⊞ IP sur ethernet
 - ⊞ IP sur ATM
 - ⊞ IP sur FDDI
 - ⊞ IP sur ligne série (PPP)
 - ⊞ IP au dessus de tout... (RFC 1149)

Le modèle en sablier

⌘ Couches réseau



Trame, Datagramme, Segment, Paquet

- ⌘ Différents noms pour les paquets à différents niveaux
 - ☑ Ethernet (lien physique): trame
 - ☑ IP (couche réseau): datagramme
 - ☑ TCP (transport): segment
- ⌘ La terminologie n'est pas strictement respectée
 - ☑ On utilise souvent le terme "paquet" à tous les niveaux

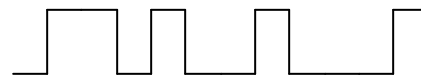
Fonctions des couches dans la pile de protocoles

7	Application		<i>Mail, Web, etc.</i>
6	Presentation		
5	Session		
4	Transport	TCP/UDP	<i>Disponibilite de bout en bout</i>
3	Network	IP	<i>Forwarding (best-effort)</i>
2	Data Link		<i>tramage</i>
1	Physical		<i>signal</i>

Couche 1: couche physique

Couche physique

- ⌘ Transmets des bits en utilisant du courant électrique, lumière, ondes radio, etc...
- ⌘ Pas de concept d'octets ou de trames.
- ⌘ Les bits sont définis par des niveaux de tension, ou autre propriété physique similaire:



1101001000

Couche 2: données

- ⌘ Regroupe les bits en trames, achemine les trames entre les stations sur un même lien
- ⌘ Une trame a un début, une fin et une taille
 - ☒ utilisation de certaines valeurs pour délimiter début/fin
- ⌘ Dispose souvent d'une adresse source et destination sur le lien (ex: adresse MAC ethernet)
- ⌘ Certains types de liens détectent les trames corrompues
- ⌘ Certains types de liens relayent les trames corrompues (pas ethernet)

Couche 3: réseau (Ex: IP)

- ⌘ Un espace d'adressage pour tout le réseau d'interconnexion
- ⌘ Ajoute une couche d'adressage supplémentaire
 - ☒ ex: adresse IP != adresse MAC
 - ☒ nous avons donc besoin d'un mécanisme de correspondance entre adresses de niveau différents
- ⌘ Non fiable (meilleur effort)
 - ☒ si les paquets sont perdus, la couche réseau ne s'en soucie pas
 - ☒ c'est le rôle des couches supérieures de retransmettre les paquets perdus

⌘ "robuste mais non fiable"

Couche 3: réseau (IP -- suite)

- ⌘ Retransmet les paquets saut par saut (hop by hop forwarding)
 - ☒ les paquets de la couche réseau sont encapsulés dans une trame de la couche données
 - ☒ différentes formes d'encapsulation sur différents type de liens
 - ☒ possibilité d'envoyer sur un lien, retransmettre sur un autre
 - ☒ Il peut y avoir de nombreux sauts entre la source et la destination

Couche 3: réseau (IP -- suite)

⌘ Prend des décisions de routage

- ☑ comment envoyer un paquet plus près de sa destination finale ?
- ☑ les tables de routage et de retransmission représentent la "connaissance" de la topologie du réseau
- ☑ les routeurs peuvent discuter entre eux pour échanger des informations sur la topologie du réseau

Couches 5,6,7

⌘5: Session

☑ pas utilisée dans le modèle TCP/IP

⌘6: Présentation

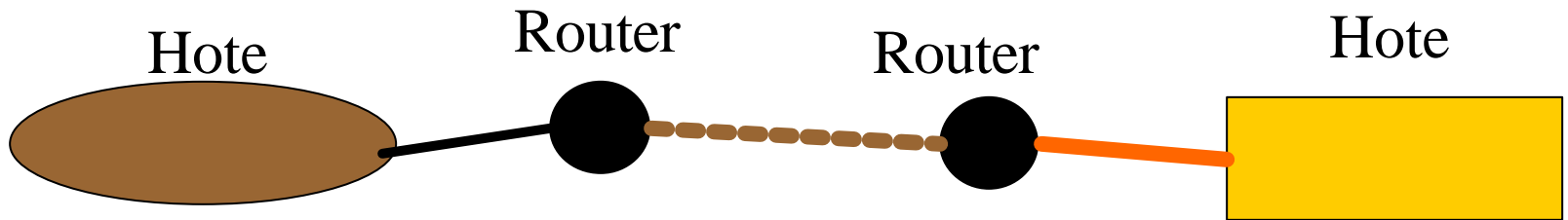
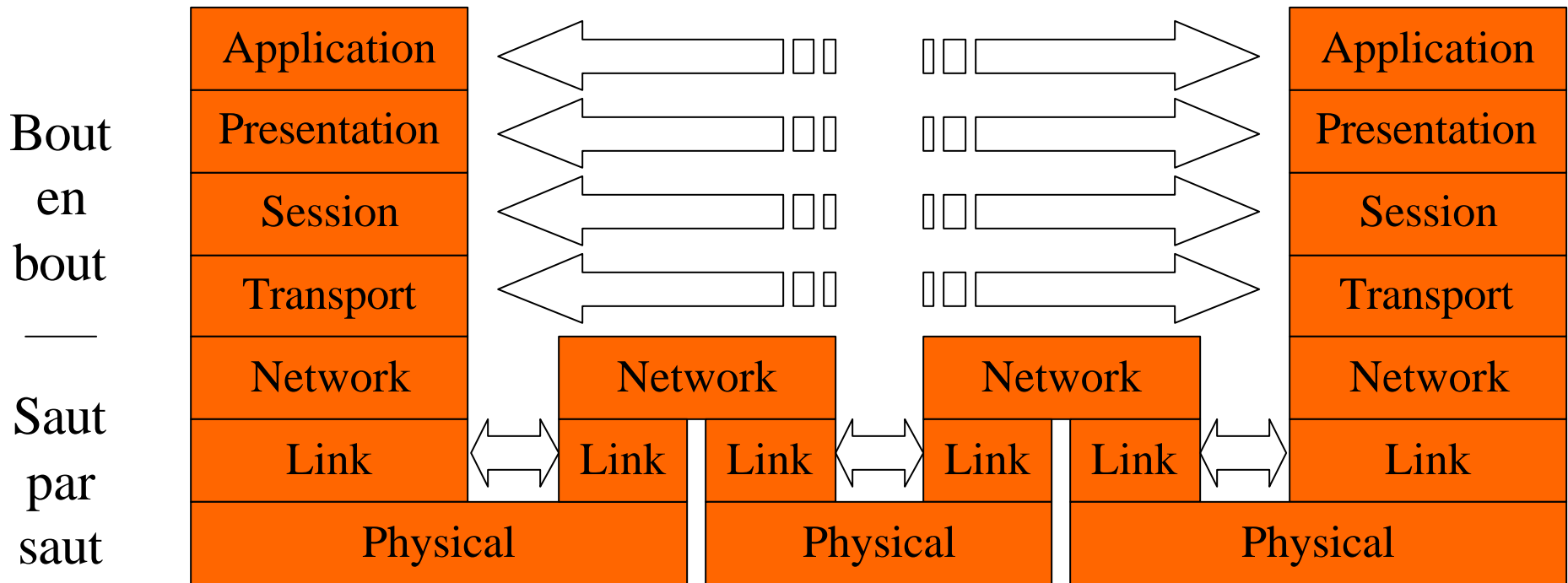
☑ pas utilisée dans le modèle TCP/IP

⌘7: Application

☑ Utilise les couches sous-jacentes

☒ ex: SMTP (mail), HTTP (web), Telnet, FTP, DNS

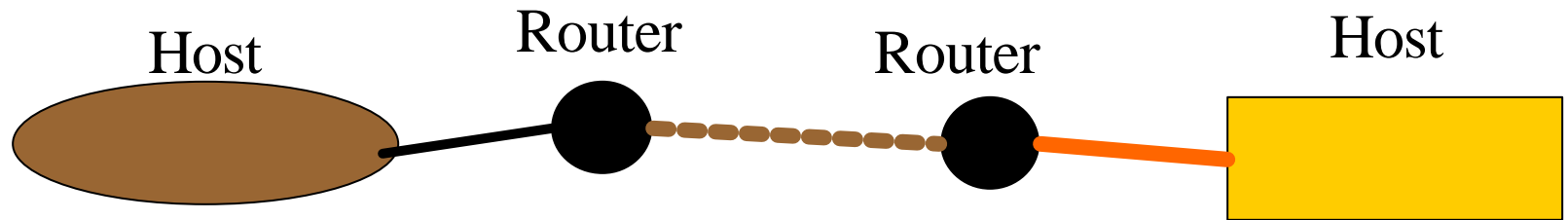
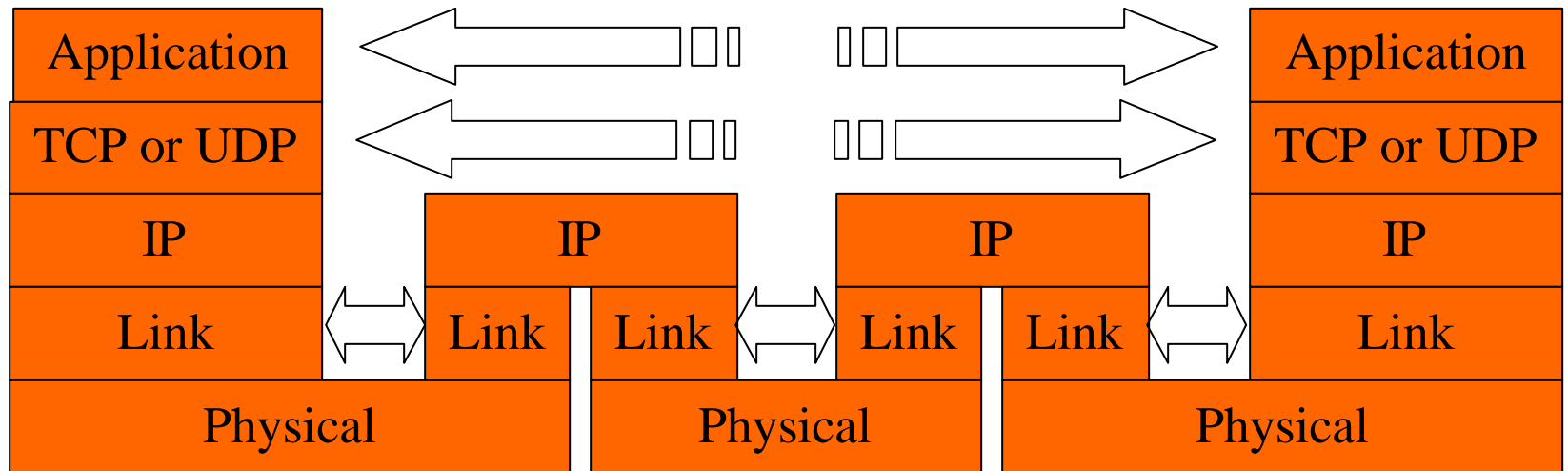
Interaction des couches (1)



Interaction des couches (2)

Pas de couche session ou presentation dans le modele TCP/IP

Bout
en
bout
—
Saut
par
saut

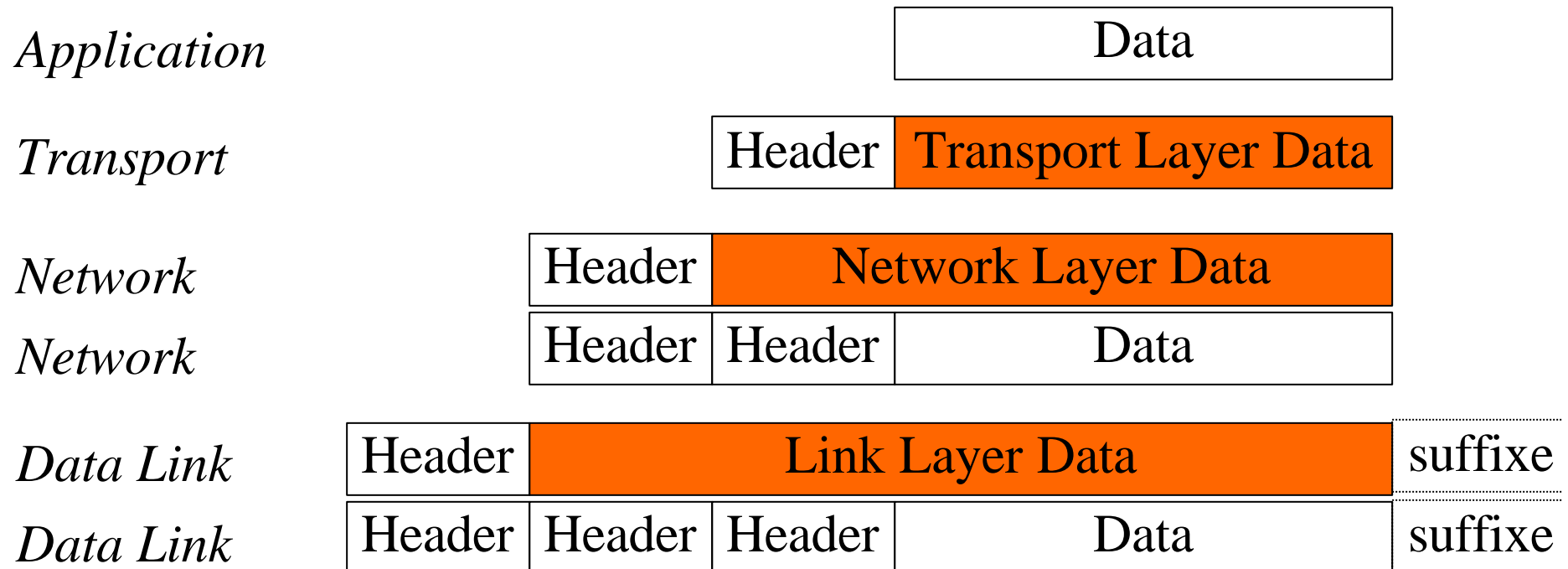


Interaction des couches (3)

- ⌘ La couche application est bout-en-bout
- ⌘ La couche transport est bout-en-bout
 - ⊞ Encapsulation/décapsulation par la couche réseau aux extrémités
- ⌘ Couche réseau s'étend à tout le réseau d'interconnexion
 - ⊞ Encapsulation/décapsulation par la couche réseau à chaque saut
- ⌘ Les couches physiques et de données peuvent être différentes à chaque saut:
 - ⊞ (ethernet, token ring, ppp, fibre...)

Encapsulation

- ⌘ Les couches inférieures ajoutent des en-têtes (et quelquefois des suffixes) aux données des couches supérieures

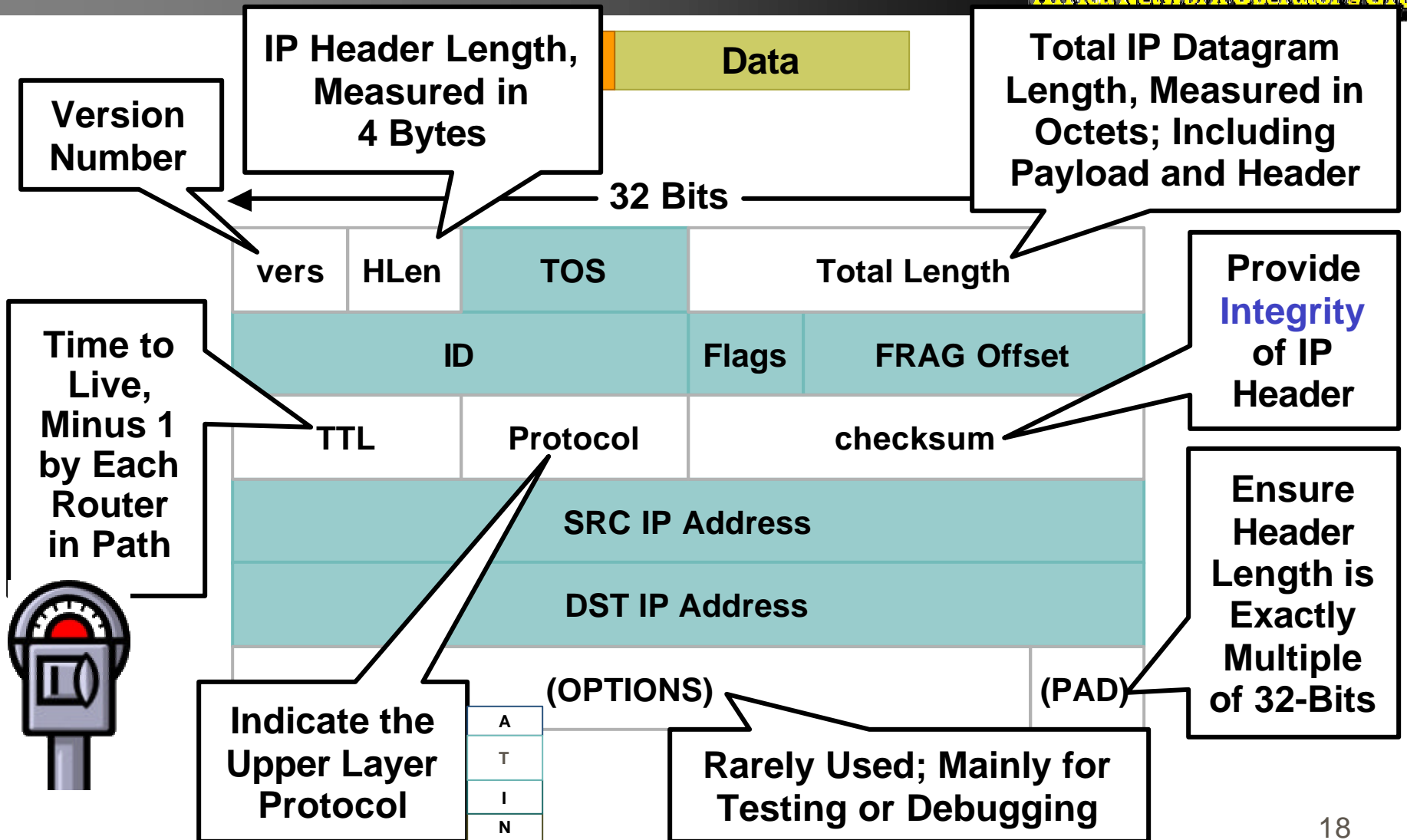


Couche 2: Trame Ethernet

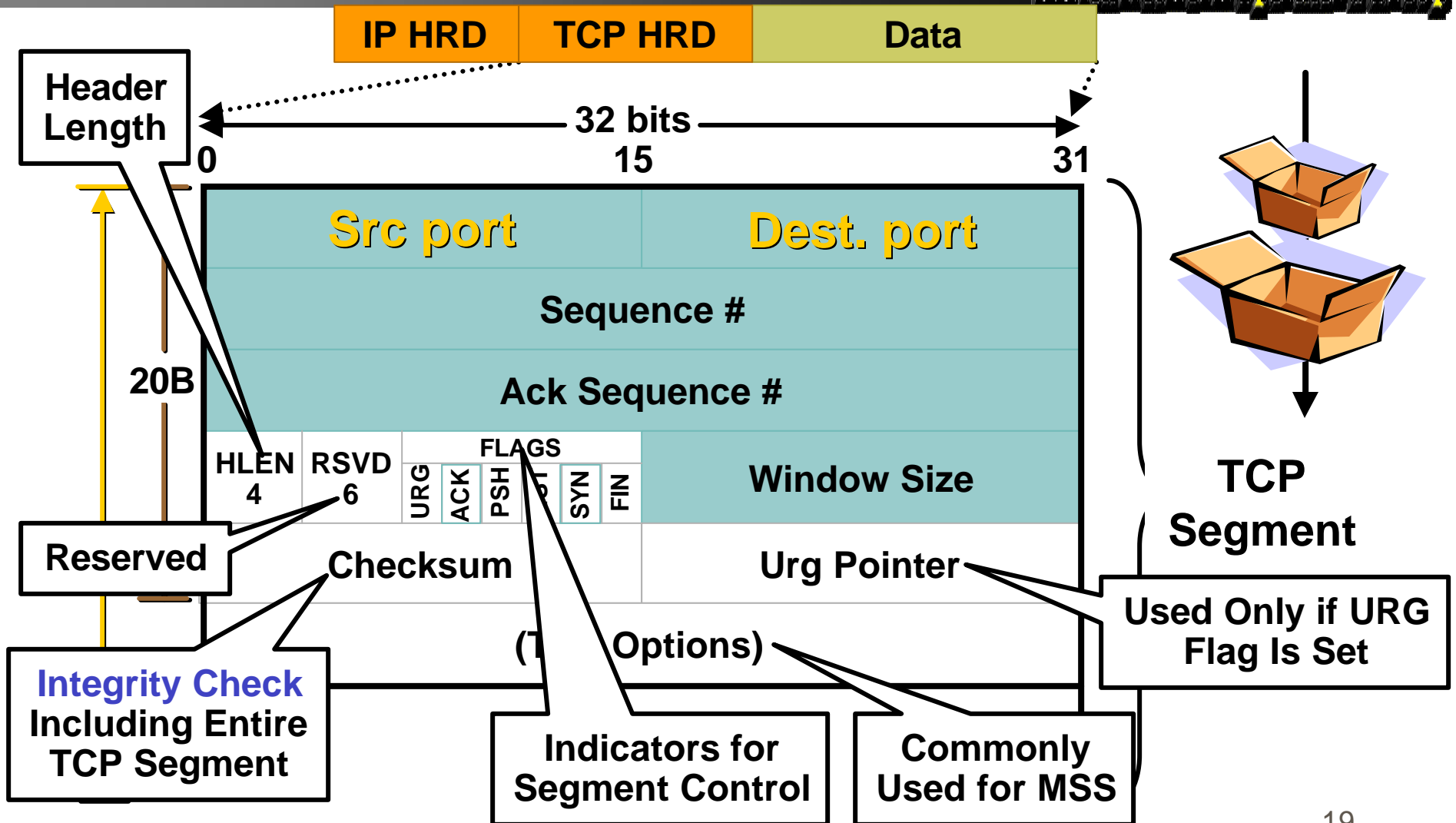
Preamble	Dest	Source	Type	Data	CRC
----------	------	--------	------	------	-----

- ☒ Destination et source sont des adresses MAC sur 48 bits (6 octets)
- ☒ Le type de valeur 0x0800 indique que la partie données de la trame données contient un datagramme IP. Le type 0x0806 indique ARP.

Couche 3: datagramme IP



Couche 4: segment TCP



But d'une adresse IP

⌘ Identification unique

- ☑ Source (quelquefois utilisé dans le filtrage)
- ☑ Destination (pour que le réseau sache où envoyer les données)

⌘ Format réseau indépendant IP au dessus de tout

Structure de base d'une adresse IP

⌘ Nombre sur 32 bits (4 octets, donc 0-255, 0-255, 0-255, 0-255)

Representation décimale

133	27	162	125
-----	----	-----	-----

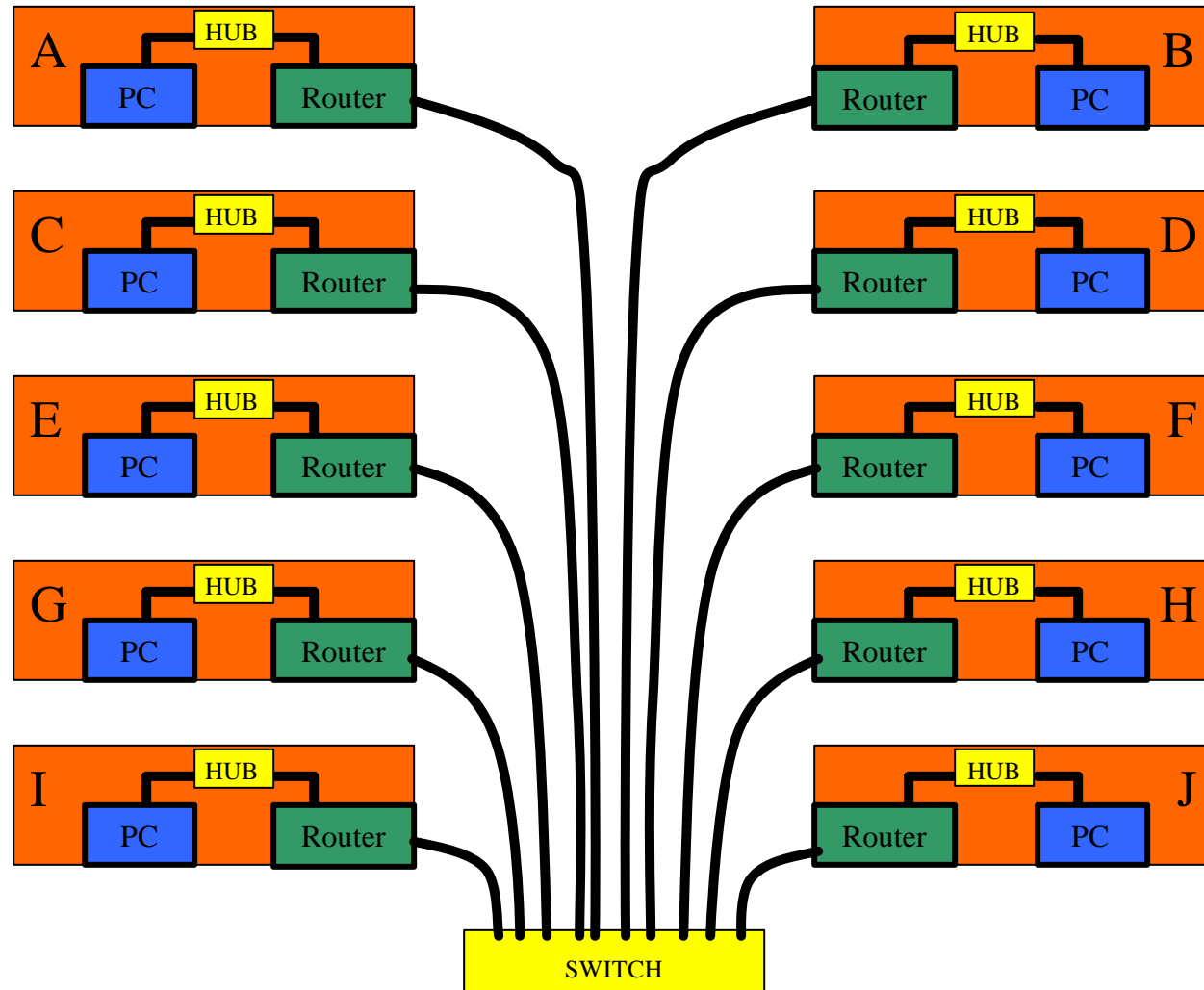
Representation binaire

10000101	00011011	10100010	01111101
----------	----------	----------	----------

Representation hexadécimale

85	1B	A2	7D
----	----	----	----

Exercice adressage



Exercice adressage

- ⌘ Construire une adresse IP pour relier votre PC au backbone
- ⌘ 81.199.111.x
- ⌘ (x = 1 pour la 1^{ère} table, 2 pour la 2^{ème}, et ainsi de suite)
- ⌘ Ecrire cette adresse sous forme décimale ainsi que binaire

Adressage dans les réseaux d'interconnexions (Internets) (1)

- ⌘ Plus d'un réseau physique
- ⌘ Différentes localités
- ⌘ Plus grand nombre de machines
- ⌘ Besoin de structure dans les adresses IP
 - ☐ partie réseau indique de quel réseau on parle dans le réseau d'interconnexion
 - ☐ partie hôte identifie la machine sur ce réseau

Adressage dans les réseaux (2)

⌘ Division hiérarchique des adresses IP

⊞ Partie réseau (préfixe)

⊞ indique quel réseau physique

⊞ Partie hôte (adresse de la machine)

⊞ indique quelle machine sur ce réseau

1111 1111	1111 1111	1111 1111	11	00 0000
1100 1101	0010 0101	1100 0111	10	00 0000

⊞ La frontière peut se situer n'importe où

⊞ TRES souvent elle n'est PAS un multiple de 8 bits!

Masques réseau (netmask)

- ⌘ Indique quels bits sont utilisés pour décrire la partie réseau
- ⌘ Plusieurs représentations:
 - ☑ notation décimale à point: 255.255.248.0
 - ☑ binaire: 11111111 11111111 11100000 00000000
 - ☑ hexadécimale: 0xFFFFE000
 - ☑ nombre de bits: / 19
- ⌘ L'opération binaire AND de l'adresse IP 32 bits avec le netmask donne la partie réseau de l'adresse

Exemple de préfixes

⌘ 137.158.128.0/17 (reseau 255.255.128.0)

1111 1111	1111 1111	1	000 0000	0000 0000
1000 1001	1001 1110	1	000 0000	0000 0000

⌘ 198.134.0.0/16 (reseau 255.255.0.0)

1111 1111	1111 1111		0000 0000	0000 0000
1100 0110	1000 0110		0000 0000	0000 0000

⌘ 205.37.193.128/26 (reseau 255.255.255.192)

1111 1111	1111 1111	1111 1111	11	00 0000
1100 1101	0010 0101	1100 0111	10	00 0000

Adresses particulières

- ⌘ Tout à 0 dans la partie hôte: l'adresse du réseau lui même
 - ☒ ex: 193.56.58.0/24 -> réseau 193.56.58.0
 - ☒ ex: 80.248.72.128/25 -> réseau 80.248.72.128
- ⌘ Tout à 1 dans la partie hôte: diffusion sur ce réseau (broadcast)
 - ☒ ex: 130.226.255.255 (130.226.0.0/16)
 - ☒ ex: 134.132.100.255 (134.132.100.0/24)
 - ☒ ex: 190.0.127.255 (190.0.0.0/17)
- ⌘ 127.0.0.0/8: boucle locale (loopback -> 127.0.0.1)
- ⌘ 0.0.0.0: utilisation spéciale (DHCP, ...)

Exercices d'adressage

- ⌘ Sachant qu'il y a 11 routeurs sur le réseau dorsal (backbone) de la salle de classe:
 - ☑ Quel est le nombre minimum de bits nécessaire dans la partie hôte pour pouvoir allouer une adresse IP unique à chaque routeur ?
 - ☑ Quel est la longueur du préfixe correspondante ? / X
 - ☑ Quel est le netmask correspondant (en décimal) ? 255.X.Y.Z
 - ☑ Combien de machines au total peut-on adresser avec ce netmask ?

Exercices binaires

⌘ En décimal (base 10), le nombre 403 signifie:

☒ $(4 \cdot 10^2) + (0 \cdot 10^1) + (3 \cdot 10^0)$

☒ $\Leftrightarrow (4 \cdot 100) + (0 \cdot 10) + (3 \cdot 1)$

☒ $\Leftrightarrow 400 + 0 + 3$

⌘ Pareillement, en binaire (base 2), le nombre 1011 signifie:

☒ $1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$

☒ $\Leftrightarrow 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1$

☒ $\Leftrightarrow 8 + 0 + 2 + 1$

☒ $\Leftrightarrow 11$ en décimal

Regroupement de nombre décimaux

- ⌘ Supposons que nous avons une liste de nombres décimaux de 4 chiffres allant de 0000 à 9999.
- ⌘ On veut faire un groupe de 10^2 (100) nombres
- ⌘ On pourrait utiliser 00xx (0000 à 0099) ou 31xx (3100 à 3199), ou encore 99xx (9900 à 9999), etc...
- ⌘ On ne peut pas utiliser (0124 à 0223) ou (3101 à 3200), parce qu'ils ne forment pas de groupe homogènes

Regroupement de nombres binaires

- ⌘ Supposons que l'on ait une liste de nombres binaires de 4 chiffres allant de 0000 à 1111.
- ⌘ On veut former des groupes de 2^2 (4) nombres.
- ⌘ On pourrait utiliser 00xx (0000 à 0011), ou 01xx (0100 à 0111) ou 10xx (1000 à 1011) ou encore 11xx (1100 à 1111)
- ⌘ On ne peut pas utiliser (0101 à 1000) ou (1001 à 1100), parce qu'ils ne forment pas de groupes homogènes

Regroupement de nombres décimaux

- ⌘ Soit une liste de nombres décimaux à 4 chiffres (0000 à 9999) $\rightarrow 10^4 = 10000$ nombres en tout.
- ⌘ On peut avoir:
 - ⊞ 10^1 (10) groupes de 10^3 (1000)
 - ⊞ 10^2 (100) groupes de 10^2 (100)
 - ⊞ 10^3 (1000) groupes de 10^1 (10)
 - ⊞ 10^4 (10000) groupes de 1
- ⌘ Tout grand groupe peut être sous divisé en groupes plus petits, et ceci de manière récursive.

Regroupement de nombres binaires

- ⌘ Soit une liste de nombres binaires à bits (0000 à 1111) $\rightarrow 2^4 = 16$ nombres en tout.
- ⌘ On peut avoir:
 - ☒ 2^1 (2) groupes de 2^3 (8)
 - ☒ 2^2 (4) groupes de 2^2 (4)
 - ☒ 2^3 (8) groupes de 2^1 (2)
 - ☒ 2^4 (16) groupes de 1
- ⌘ Tout grand groupe peut être sous divisé en groupes plus petits, et ceci de manière récursive

Regroupement de nombres binaires

- ⌘ Soit un grand nombre sur 32-bits (000...000 à 111...111)
- ⌘ On peut avoir:
 - ☑ 2^0 (1) groupe de 2^{32} nombres
 - ☑ 2^8 (256) groupes de 2^{24} nombres
 - ☑ 2^{25} groupes de 2^7 nombres
- ⌘ Prenons un groupe de 2^7 (128) nombres
 - ☑ ex: 1101000110100011011010010xxxxxxxx
 - ☑ On peut le diviser en 2^1 (2) groupes de 2^6 (64)
 - ☑ On peut le diviser en 2^3 (8) groupes de 2^4 (16)
 - ☑ etc...

Niveaux de hiérarchie

- ⌘ Ne pas oublier de la division d'une adresse IP en une partie réseau et une partie hôte
- ⌘ De manière similaire, on peut regrouper plusieurs réseaux dans un plus grand bloc, ou diviser un grand bloc en blocs plus petits (supernetting et subnetting)
 - ☑ nombre arbitraire de niveaux et de hiérarchies
 - ☑ les blocs n'ont pas besoin d'être de la même taille (VLSM)
- ⌘ Les vieilles implémentations sont souvent moins flexibles

Ancienne classification des adresses IP (1)

- ⌘ Différentes classes utilisées pour représenter différentes tailles de réseaux (petit, moyen, grand)
- ⌘ Classe A (grand)
 - ☒ 8 bits de réseau, 24 bits hôtes (/8 => 255.0.0.0)
 - ☒ Premier octet dans l'intervalle 0-127
- ⌘ Classe B (moyen)
 - ☒ 16 bits de réseau, 16 bits hôtes (/16 => 255.255.0.0)
 - ☒ Premier octet dans l'intervalle 128-191
- ⌘ Classe C (petit)
 - ☒ 24 bits de réseau, 8 bits hôtes (/24 = 255.255.255.0)
 - ☒ Premier octet dans l'intervalle 192-223

Ancienne classification des adresses IP (2)

⌘ Il suffit de regarder l'adresse pour connaître la classe

☒ Classe A: 0.0.0.0 à 127.255.255.255

☒ binaire 0xxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

☒ Classe B: 128.0.0.0 à 191.255.255.255

☒ binaire 10xxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

☒ Classe C: 192.0.0.0 à 223.255.255.255

☒ binaire 110xxxxx xxxxxxxx xxxxxxxx xxxxxxxx

☒ Classe D (multicast): 224.0.0.0 à 239.255.255.255

☒ binaire 1110xxxx xxxxxxxx xxxxxxxx xxxxxxxx

☒ Classe E (réservé) : 240.0.0.0 à 255.255.255.255

Netmasks implicites des classes d'adresses

- ⌘ Un réseau "à classe" (classful) à netmask / longueur de préfixe implicite:
 - ☑ Classe A: préfixe de longueur 8, netmask 255.0.0.0
 - ☑ Classe B: préfixe de longueur 16, netmask 255.255.0.0
 - ☑ Classe C: préfixe de longueur 24, netmask 255.255.255.0
- ⌘ Les équipements de routage anciens suivent souvent le netmask implicite
- ⌘ Les équipements de routage modernes utilisent toujours un netmask /longueur de préfixe explicite

Découpage en sous-réseau (subnetting) des réseaux à classe

- ⌘ Les anciens équipements de routage permettaient à un réseau à classe (classful) d'être divisé en sous-réseaux
 - ☒ Tous les sous-réseaux (du même réseau à classe) devaient être de même taille et avoir le même netmask
- ⌘ Les sous-réseaux ne pouvaient pas être sous-divisés eux-mêmes
- ⌘ Aucune de ces restrictions ne s'applique aux systèmes modernes

Groupage traditionnel (supernetting)

- ⌘ Certains anciens systèmes autorisaient le supernetting, c'est-à-dire la création de groupes de réseaux adjacents, vus comme un seul réseau
 - ☒ Par exemple: combinaison de 2 réseaux de Classe C (ayant des numéros consécutifs) peuvent être groupé en un supernet de masque
255.255.254.0 (/23)
- ⌘ Un système moderne utilise un principe plus générale sans classes (classless)

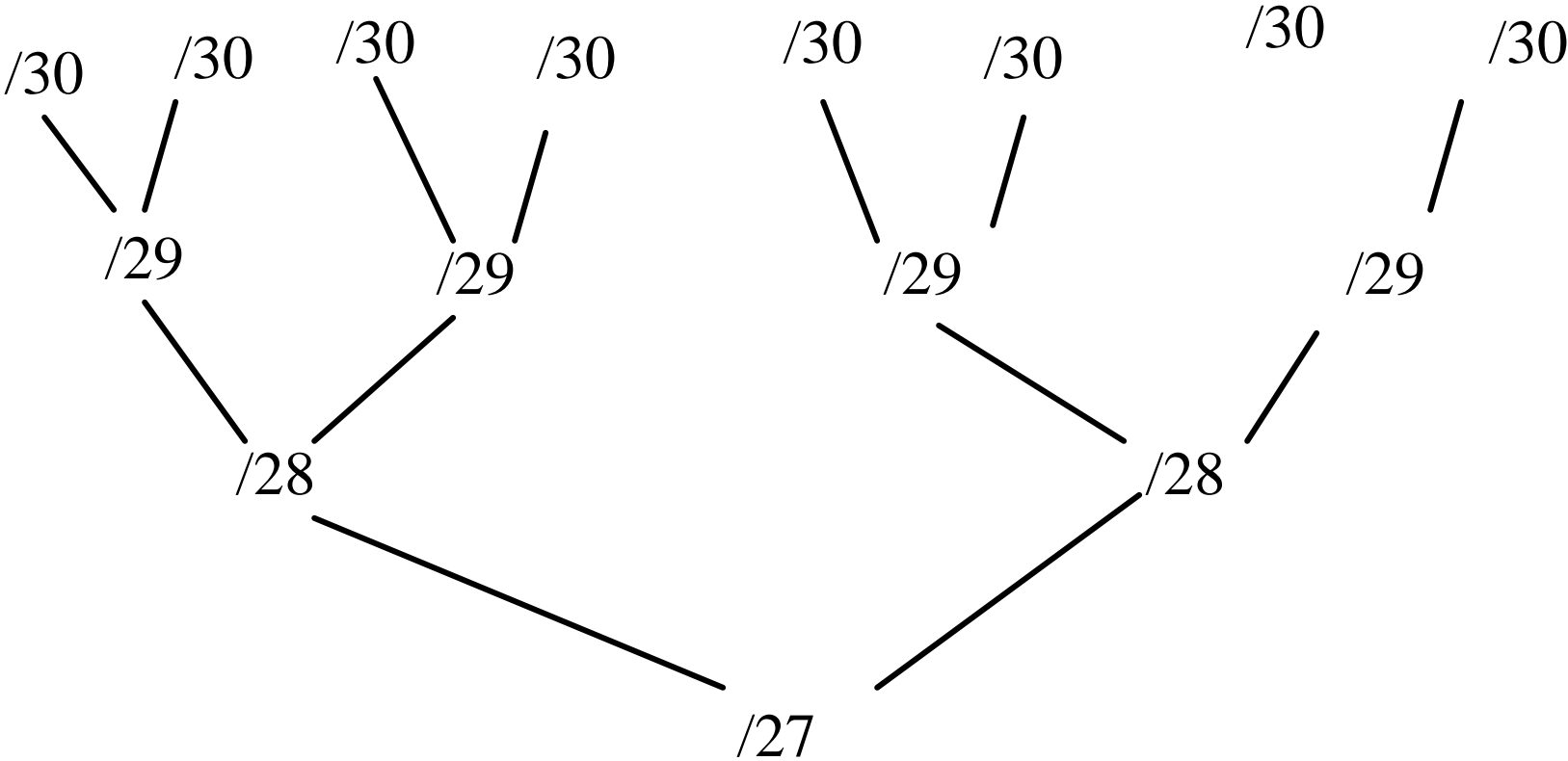
Adressage sans classes (classless)

- ⌘ Oubliez les classes A, B et C
- ⌘ Le routage dans l'Internet et la gestion d'adresse se font aujourd'hui sans classes
- ⌘ CIDR = Classless Inter Domain Routing
- ⌘ (routage interdomaine sans classe)
 - ☒ le routage ne déduit pas qu'un réseau dont le nombre est de classe A, B ou C à une longueur de préfixe de /8, /16 ou /24
- ⌘ VLSM = Variable Length Subnet Masks (masque de sous-réseau à longueur variable)
 - ☒ le routage ne conclut pas que tous les sous-réseaux sont de même taille

Exemple d'adressage sans classe

- ⌘ Un grand ISP obtient un gros bloc d'adresses
 - ☒ ex: un préfixe /16, c'est à dire 65536 adresses
- ⌘ Il alloue des blocs plus petits aux clients, par exemple:
 - ☒ un préfixe /22 (1024 adresses) à un client
 - ☒ et un préfixe /28 (16 adresses) à un autre client
- ⌘ Une organisation qui obtient un /22 de son ISP peut à son tour le sous-diviser en blocs plus petits, par exemple:
 - ☒ un préfixe /26 (64 adresses) pour un département,
 - ☒ un préfixe /27 (32 adresses) pour un autre département

Hiérarchie des préfixe CIDR



Exercice d'adressage sans classe

- ⌘ Prenons un bloc d'adresses 81.199.111.0/24
- ⌘ Allouez 8 blocs /29 et un bloc /28
- ⌘ Quelle est l'adresse IP de chaque bloc ?
(adresse réseau)
 - ☑ en notation longueur de préfixe (A.B.C.D/X)
 - ☑ avec netmask en décimal
 - ☑ Les plages d'adresses associées
- ⌘ Quelle est le plus grand bloc qui soit encore disponible ?
- ⌘ Quels autres blocs reste-t-il ?

Routeur IP

- ⌘ Un équipement ayant plus d'une interface au niveau de la couche de données
- ⌘ Plusieurs adresses IP (dans des sous-réseaux différents) sur des interfaces différentes
- ⌘ Reçoit des paquets sur une interface, et le retransmet (forwarding), souvent sur une autre interface, pour les rapprocher de leur destination
- ⌘ Maintient une table de retransmission (!= routage, bien qu'elles soient souvent confondues)

IP -une décision pour chaque paquet

- ⌘ Un paquet est reçu sur une interface
- ⌘ Vérification pour voir si l'adresse de destination est le routeur lui-même
- ⌘ Décrément du TTL (Time to Live), destruction du paquet si le TTL atteint 0
- ⌘ Recherche de la destination dans la table de retransmission
- ⌘ La destination peut être sur un lien directement connecté , ou via un autre routeur

Le forwarding saut par saut

- ⌘ Chaque routeur essaye de transmettre le paquet un saut plus loin, vers la destination
- ⌘ Chaque routeur prend une décision indépendante, basé sur sa table de retransmission
- ⌘ Des routeurs distinct auront des tables de routage différentes
- ⌘ Les routeurs discutent entre eux en utilisant des protocoles des routage, pour mettre à jour la table de routage et de retransmission

Structure de la table de retransmission (forwarding)

- ⌘ La table ne contient pas toutes les adresses IP de l'Internet
- ⌘ A la place, la table de forwarding contient des préfixes (numéros de réseau)
- ⌘ "Si les /n premiers bits correspondent à cette entrée, envoyer le datagramme dans cette direction"
- ⌘ Si plus d'un préfixe correspond, le préfixe le plus long "gagne" (route la plus spécifique)
- ⌘ 0.0.0.0/0 est une "route par défaut" -- elle correspond à tout, mais uniquement s'il n'existe pas d'autre préfixe correspondant.

Types de liens

- ⌘ Différentes stratégies d'encapsulation et de d'acheminement des paquets
- ⌘ IP sur les différents type de liens
- ⌘ Point- à -Point - ex: PPP
- ⌘ Diffusion (broadcast) - ex: Ethernet
- ⌘ Non-broadcast, multi-access - ex: Frame Relay, ATM

Liens point- à -point

- ⌘ Deux hôtes sont reliés par un lien point- à -point
 - ☒ les données envoyées par l'un sont reçues par l'autre
- ⌘ L'émetteur prend un datagramme IP, l'encapsule d'une manière ou d'une autre (PPP, HDLC, SLIP, ...) et le transmet
- ⌘ Le destinataire enlève l'encapsulation de la couche de données
- ⌘ Vérification de l'intégrité , jette les mauvais paquets, et traite les paquets corrects

Lien à diffusion (broadcast) - 1

⌘ De nombreux hôtes connectés à un média de diffusion

☑ les données envoyées par un hôte peuvent être reçues par tous les autres

☑ ex: radio, ethernet

Lien à diffusion (broadcast) - 2

- ⌘ Limite les interférences en empêchant les transmissions simultanées (CSMA/CD ou CSMA/CA)
- ⌘ Addressages individuel des hôtes
 - ☑ les machines savent ainsi quels paquet leurs sont destinés, et donc traiter, et lesquels ignorer
 - ☑ l'adressage de la couche données est très différente de celui de la couche réseau
- ⌘ Correspondance entre les adresses de couche réseau et celles de la couche de données (ex: ARP)

Liens NMBA (Non-broadcast, multi-access)

- ⌘ ex: X.25, Frame Relay, SMDS
- ⌘ De nombreux hôtes
- ⌘ Chaque hôte dispose d'une adresse unique sur la couche de données
- ⌘ Chaque hôte peut potentiellement envoyer un paquet à tout autre machine
- ⌘ Chaque paquet est reçu par un seul hôte
- ⌘ Dans certains cas, la diffusion (broadcast) est également disponible

Rappels essentiels sur l'ethernet

- ⌘ Ethernet est un média de diffusion
- ⌘ Structure d'une trame ethernet

Preamble	Dest	Source	Type	Data	CRC
----------	------	--------	------	------	-----

- ⌘ Un datagramme IP constitue la totalité de la partie données d'une trame Ethernet
- ⌘ Mécanisme d'acheminement (CSMA/CD)
 - ☑ On attend et on réessaie plus tard en cas de collision

Correspondance IP/Ethernet

⌘ Adresse Internet

- ☑ unique au monde (sauf pour les réseaux privés RFC 1918)
- ☑ indépendante du réseau physique

⌘ Adresse Ethernet

- ⌘ unique au monde (sauf erreurs)
 - ☑ ethernet seulement

- ⌘ Besoin d'effectuer une correspondance de la couche supérieure vers la couche inférieure (c'est-à-dire IP vers Ethernet), en utilisant ARP

Questions & réponses