Adressage IP 3ème partie

Hainaut Patrick 2024

But

- Nous terminons par cette troisième partie le tour du protocole IPv4 en y voyant certains aspects avancés

- Quelques répétitions seront faites par rapport aux parties Theo05a et Theo05b, pour constituer un ensemble cohérent

Introduction

Systèmes de numération

- « Façon d'énoncer ou d'écrire les nombres »
- Séries hiérarchisées de symboles
- Plusieurs numérations :
  - Arabe : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50, 100, 1000
  - Romaine : I, II, IV, V, VI, VII, IX, X, L, C, M
- Plusieurs bases par numération
  - Nombre de symboles différents utilisés
  - Exemple: notre système décimal est de base 10

Introduction
Système décimal

- Base 10
  - dix chiffres
  - 0 1 2 3 4 5 6 7 8 9

- Nombres : composition de chiffres

- Exemple :
  - 1995_{10} (mille neuf cent nonante cinq)
  - $= 1*10^3 + 9*10^2 + 9*10^1 + 5*10^0$
  - $= 1*1000 + 9*100 + 9*10 + 5*1$
  - Plus les chiffres sont à gauche, plus leurs poids est élevé

Introduction

Système décimal

- Exemple : 1995_{10}
- Le rang :
  - 1 : rang des milliers
  - 5 : rang des unités
- Le poids :
  - 1 : chiffre de poids fort
  - 5 : chiffre de poids faible

Introduction
Système binaire

- Base 2
  - chiffres 0 et 1
  - Le courant passe ou pas
  - Ca permet de définir deux états disctincs
  - Toute la logique informatique (et électronique digitale) est basée sur ce principe

- Exemple :
  - 1011_2 = onze
  - $= 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0$
  - $= 8 + 2 + 1$
  - $= 11_{10}$

Introduction
Système binaire

- Bit = « Binary digit »
  = chiffre binaire

- MSB = « Most significant bit »
  = bit de poids fort

- LSB = « Least significant bit »
  = bit de poids faible

Introduction

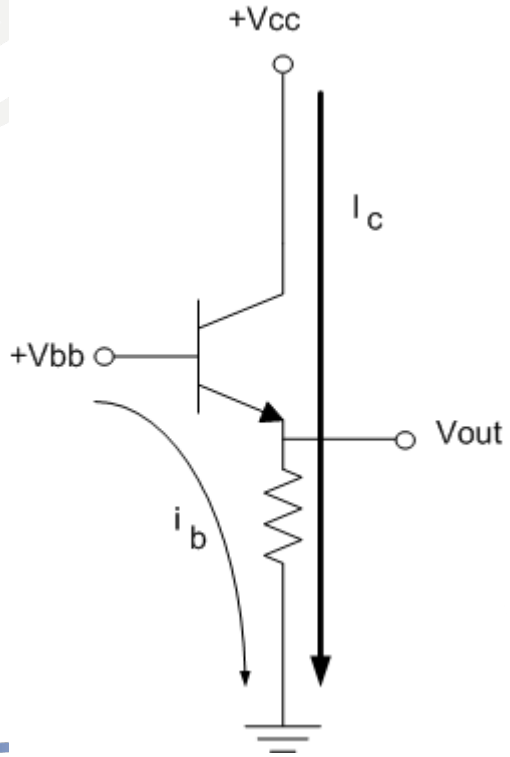
Pourquoi employer le système binaire ?

- Les µ-Processeurs actuels sont composés de près d'un milliard de transistors en commutation

- Chaque transistor se comporte comme un interrupteur

- Si Vbb = 0 volts, Vout = 0 volts
  => 0 logique

- Si Vbb = +Vbb, Vout +/-= +Vcc
  => 1 logique



Introduction

Pourquoi employer le système binaire ?

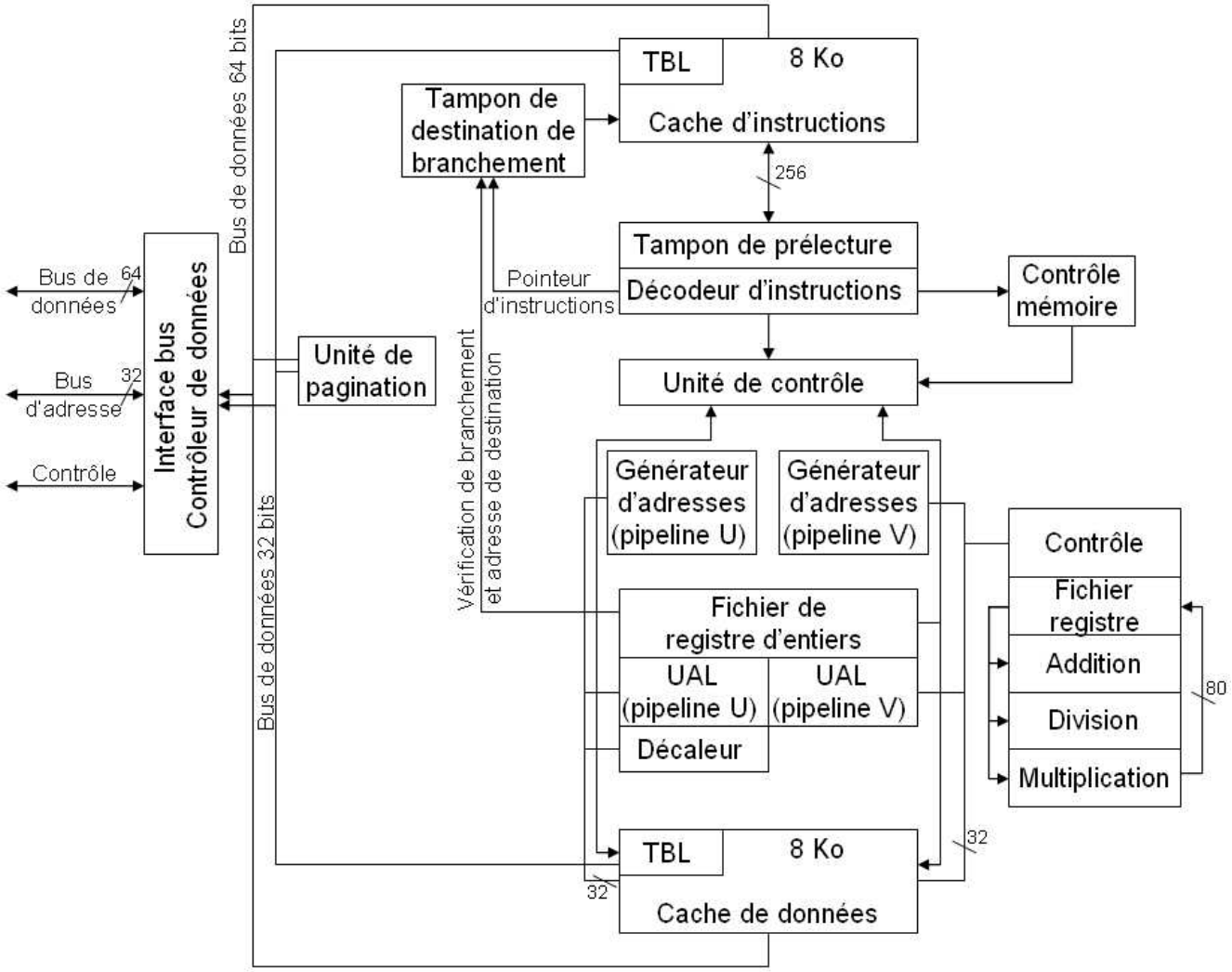
- Chaque transistor contrôle donc 1 bit
- On groupe les transistors de façon à former des groupes de bits.
- Ces groupements binaires permettent de représenter des informations
  - Exemple: le caractère « A » est codé 65 en ASCII, ce qui correspond à 01000001 en binaire
- Remarque: en informatique et électronique binaire, on compte à partir de 0 et pas à partir de 1 comme en décimal !!!
  Donc 3 en décimal représente 4 états binaires (00, 01, 10 et 11)

Introduction

Architecture interne d'un P4

Les transistors sont groupés de façon à former des groupes logiques

Les informations qui circulent sont du binaire



Introduction
octets et mots binaires

- Si les bits sont groupés:
  - par 8, cela forme un octet (byte en anglais)
  - par 16, cela forme un mot (word en anglais)

Introduction
octets et mots binaires

- Table des codes ASCII sur 7 bits:

Version Internationale

				B6	0	0	0	0	1	1	1	1
				B5	0	0	1	1	0	0	1	1
				B4	0	1	0	1	0	1	0	1
B3	B2	B1	B0	Exemple : E = 100 0101$_{(2)}$ = 69$_{(10)}$. Appuyez sur "ALT", saisissez 69 et relâchez "ALT". Convaincu !								
0	0	0	0		NUL	DLE	SP	0	@	P	`	p
0	0	0	1		SOH	DC1	!	1	A	Q	a	q
0	0	1	0		STX	DC2	~	2	B	R	b	r
0	0	1	1		ETX	DC3	#	3	C	S	c	s
0	1	0	0		EOT	DC4	$	4	D	T	d	t
0	1	0	1		ENQ	NAK	%	5	E	U	e	u
0	1	1	0		ACK	SYN	&	6	F	V	f	v
0	1	1	1		BEL	ETB	'	7	G	W	g	w
1	0	0	0		BS	CAN	(	8	H	X	h	x
1	0	0	1		HT	EM	)	9	I	Y	i	y
1	0	1	0		LF	SUB	*	:	J	Z	j	z
1	0	1	1		VT	ESC	+	;	K	[	k	{
1	1	0	0		FF	FS	,	<	L	\	l	|
1	1	0	1		CR	GS	-	=	M	]	m	}
1	1	1	0		SO	RS	.	>	N	^	n	~
1	1	1	1		SI	US	/	?	O	_	o	DEL

- On retrouve le caractère « A » qui se code en binaire:
  1000101 sur 7 bits,
  01000101 sur 8 bits,
  ce qui donne, en décimal:
  64+4+1=65
- Dans cette table originale, il n'y a pas de caractères accentués, absents de la grammaire anglaise
- Ils seront ajoutés dans une version à 8 bits du code ASCII

Introduction
octets et mots binaires

- Sur 16 bits, on trouvera la table Unicode qui tient compte de tous les caractères issus des différents langages.

Introduction

Conversion binaire - décimal

- Soit un octet: 10011101
  - A chaque chiffre binaire correspond un poids comme vu précédemment.
  - Cela donne:

128	64	32	16	8	4	2	1
1	0	0	1	1	1	0	1

128 + 16+8+4 + 1 = 157 en décimal

- 8 bits à 0 donne 0 en décimal
- 8 bits à 1 donne 255 en décimal
- 1 octet peut représenter 256 valeurs (et non 255 !)

Introduction
Conversion binaire - décimal

- Si on raisonne sur 16 bits:
  - 1000110011110000

32768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1
1 0 0 0 1 1 0 0 1 1 1 1 0 0 0 0

32768+ 2048 + 1024 + 128 + 64 + 32 + 16 = 36080 en décimal

- 16 bits à 0 donne 0 en décimal
- 16 bits à 1 donne 65535 en décimal
- Un mot binaire peut représenter 65536 valeurs

Introduction
Conversion décimal-binaire

- Si le nombre est compris entre 0 et 255, on peut le coder sur un octet
  - Exemple: 177, c'est 128 + 32 + 16 + 1
    => 10110001

- On part du MSB (à gauche) et on regarde quels poids binaires «rentrent» dans notre nombre décimal
  - Exemple, pour 76: 128 -> trop grand,
    64 -> ok, il reste donc 76-64-> 12,
    32, 16 -> trop grand
    8 -> ok, il reste 12-8 -> 4
    4 -> ok, il reste 0 ======> 01001100

- Remarque: on n'utilise pas la méthode "des divisions" qui n'est pas assez intuitive et génère une perte de temps

Introduction

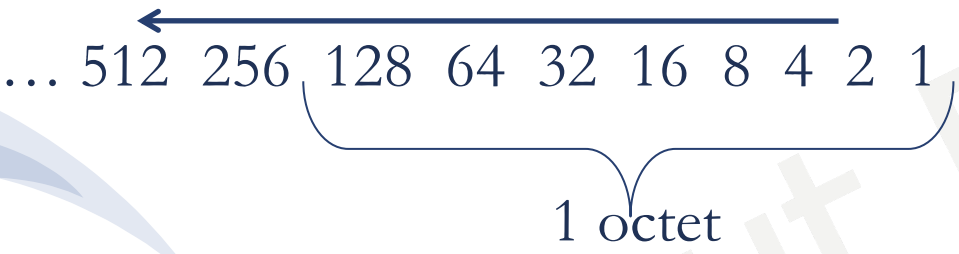
Conversion décimal-binaire

- Si le nombre est compris entre 256 et 65535, on le code sur un mot (2 octets)
  - Exemple: 325, c'est 256 + 64 + 4 + 1
    => 00000001 01000101

- Même méthode que précédemment

Introduction
A retenir

- Vous devez toujours avoir en tête la suite des poids binaires en partant du LSB à droite



- On tient compte du poids binaire pour chaque chiffre à 1 dans notre nombre binaire

Introduction
Binaire - Exercices

- Convertissez en binaire:
  16, 298, 32768, 255, 0, 777

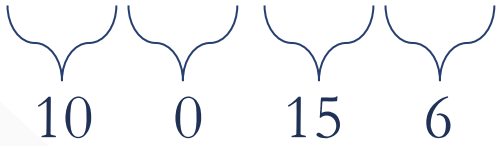
- Convertissez en décimal:
  01011011
  10001011 01001111
  00111111
  00001111
  11101111

Introduction

Système hexadécimal

- Base 16
  - chiffres 0 1 2 3 4 5 6 7 8 9 A B C D E F
  - A vaut 10, B vaut 11, C vaut 12, D vaut 13
    E vaut 14 et F vaut 15
- Permet de représenter un octet avec 2 chiffres
  - Exemple: 11010111 => D7 en hexa

10100000 11110110 => A0 F6



Introduction

Système hexadécimal

- Pour la conversion hexadécimal-binaire, chaque chiffre hexa est remplacé par son équivalent sur 4 chiffres binaires

- Pour la conversion binaire-hexadécimal, on part des bits de poids faibles (à droite), on les groupe par 4 pour les remplacer par un chiffre hexa.

Introduction

Hexadécimal - Exercices

- Convertissez d'hexa en binaire:
  7B, FF, 10, 25, FB, AC

- Convertissez en hexa:
  01011011
  10001011 01001111
  00111111
  00001111
  11101111

Introduction

Système hexadécimal

- Au début de la micro-informatique, quand on programmait en langage machine (instructions de bas niveau, au niveau processeur), les instructions étaient représentées par des séquences de 8 ou 16 bits qu'on encodait sous forme hexadécimal, plus compacte et surtout moins sujet aux erreurs d'encodage

- Les adresses MAC et les adresses IPv6 seront codés en hexadécimal pour le gain de place par rapport au décimal (et a fortiori au binaire)

Adresses MAC
Généralités

- On a vu précédemment que toute carte réseau ou équipement réseau similaire possède une adresse physique, appelée aussi adresse MAC
- C'est valable aussi pour tout l'IoT (Internet of Things)
- Cette adresse est unique au monde et est attribuée une fois pour toute à la fabrication (même si on peut la changer de façon logicielle)
- Les adresses sont définies par l'IEEE (Institute of Electrical and Electronics Engineer) qui attribue des plages d'adresses à chaque fabriquant de cartes réseau
  - Cisco, par exemple, a reçu (entre-autre) la plage 00900C

Adresses MAC
Généralités

- Cette adresse est utilisée au niveau de la couche de liaison (couche 2 du modèle OSI) par beaucoup de technologies réseau dont: Ethernet, Token Ring, Wifi, Bluetooth, …

- Pour communiquer entre-eux, les PC du réseau local utilisent des trames dans lesquelles se trouvent l'adresse MAC de l'expéditeur et du destinataire

- Ces trames sont transmises en série (flot de bits) sous forme de signaux électriques ou d'ondes (Wifi, Bluetooth) ou optiques (fibre)

Adresses MAC

Structure d'une adresse MAC

- Elle est constituée de 6 octets variant de 0 à 255
- L'adresse est donnée sous forme hexadécimale
  - Exemple: 00:06:25:DF:BF:63
- L'adresse FF:FF:FF:FF:FF:FF est l'adresse de diffusion (broadcast) qui permet d'envoyer des données à l'ensemble du réseau
- Les : sont dans certains cas remplacés par des – et les lettres peuvent être en minuscules
  - Exemple: 4a-25-df-25-37-0b

Adresses MAC

Relever l'adresse MAC

- Sous Windows:
  - Dans une invite de commande, tapez ipconfig /all
  - L'adresse MAC se trouve sur la ligne « Adresse Physique »
- Sous MAC:
  - Dans une console, tapez ifconfig
  - L'adresse MAC se trouve après la mention « ether »
- Sous Linux:
  - Dans une console, tapez ip addr
  - L'adresse MAC se trouve après la mention « Hwaddr »

Adresses MAC

Nécessité des adresses IP

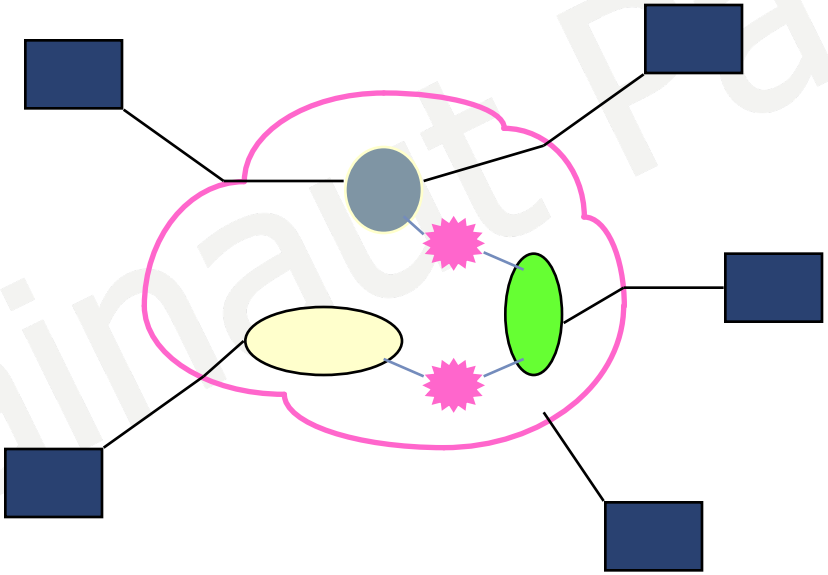
- Adresses MAC uniques mais…
  - Liées au matériel
  - Pas de lien logique possible entre deux adresses

- ➔ Besoin d'adresses logiques
  - Non liées au matériel (possibilité de changer facilement de carte réseau)
  - Identifiant un équipement de façon unique
  - Regroupant logiquement les machines
  - ➔ Adresses IP

Adresses IP

Nécessité des adresses IP

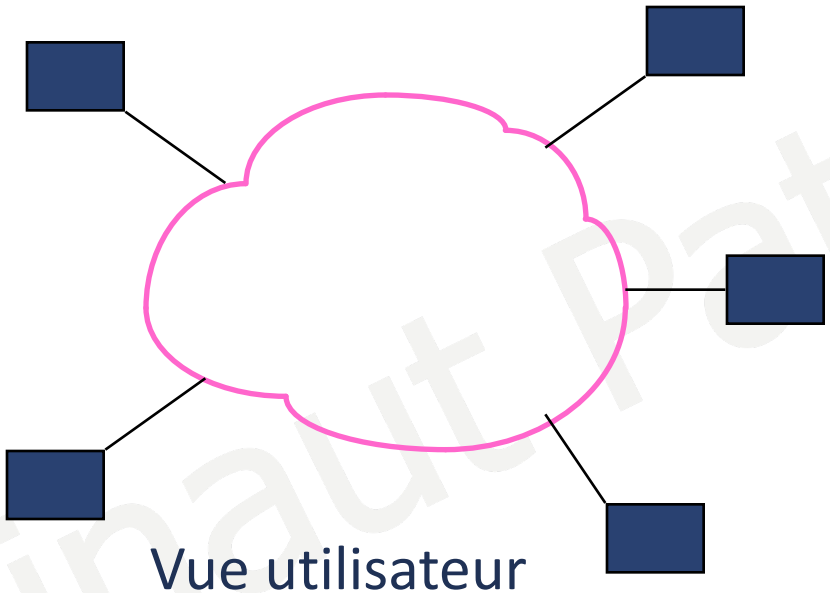
- Le protocole IP permet d'encapsuler les spécificités des différents réseaux physiques (Ethernet, token ring, …) pour proposer un service commun aux applications



Vue réelle du réseau

Adresses IP

Nécessité des adresses IP



- IP fait apparaitre l'ensemble des réseaux disparates comme un seul et unique réseau

Adresses MAC

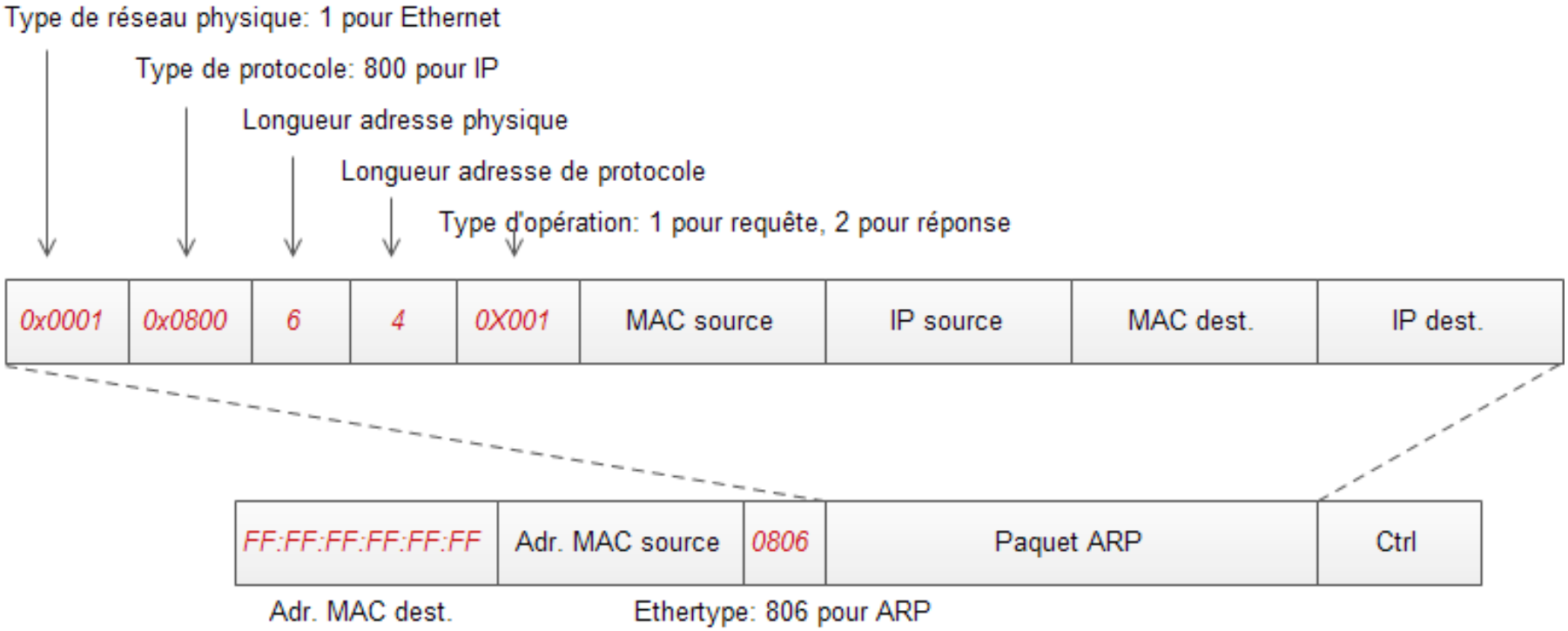
Correspondance entre IP et MAC

- Les PC du réseau local utilisant les adresses MAC pour communiquer et l'utilisateur des adresses IP, il faut donc pouvoir établir une correspondance entre les deux
  - Le protocole ARP (Address Resolution Protocol) permet de connaître l'adresse physique du destinataire à partir de l'adresse IP
  - La requête ARP sera encapsulé dans une trame Ethernet
  - Une table de correspondance est créée
  - L'utilitaire arp permet de consulter cette table
- L'utilitaire arp a été vu dans Theo05b

Adresses MAC

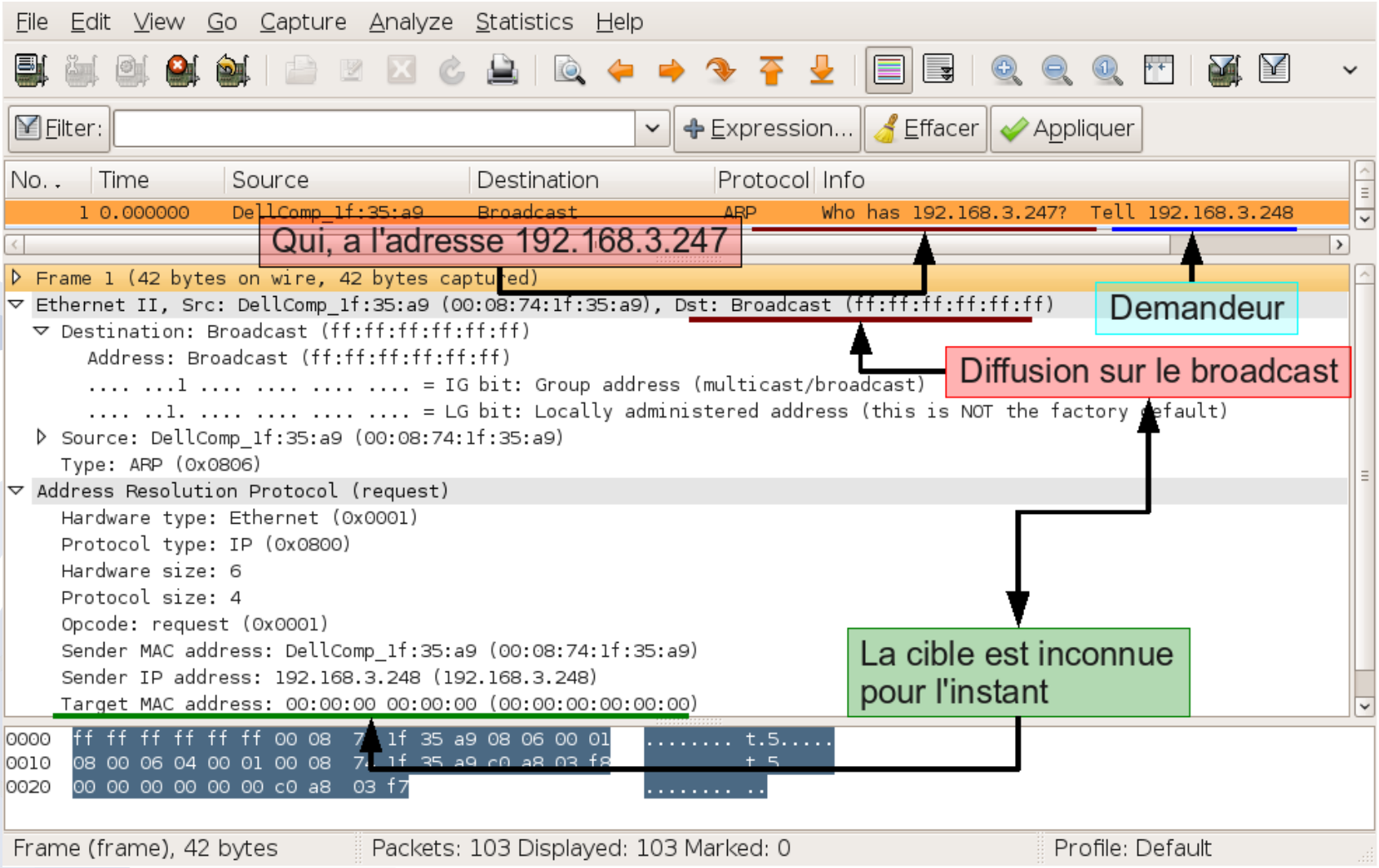
Correspondance entre IP et MAC

- Trame Ethernet II contenant une requête ARP



Adresses MAC

Correspondance entre IP et MAC



Adresses IP
Généralités

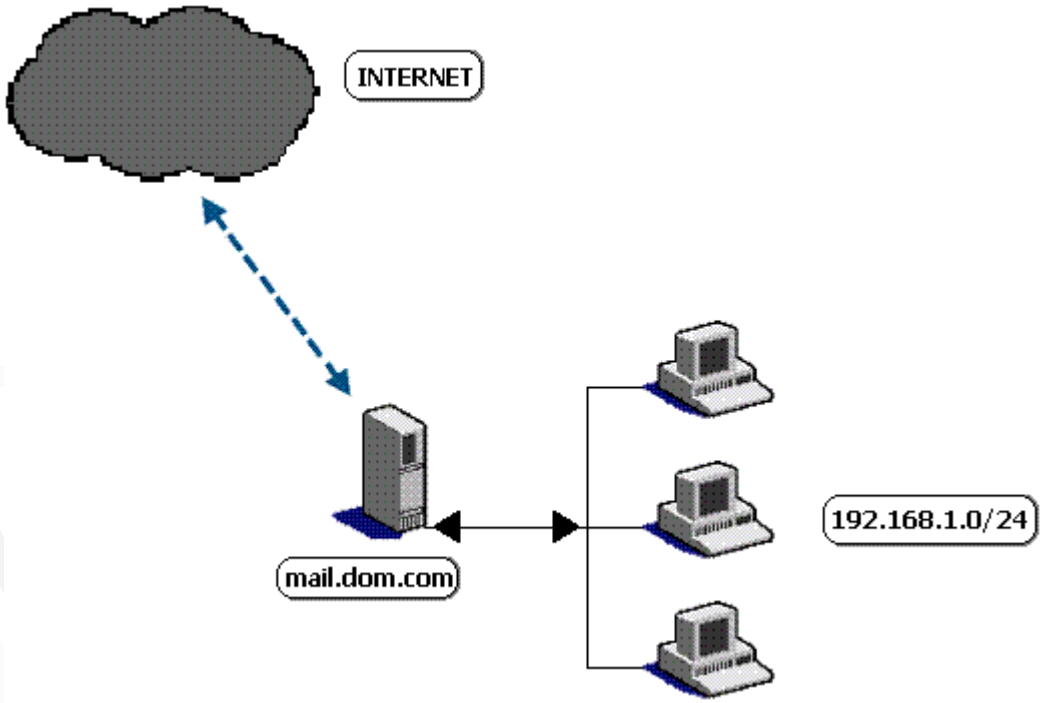
- Deux versions de protocole: IPv4 et IPv6
  - En IPv4, les adresses sont codées sur 4 octets (32 bits)
  - En IPv6, elles sont codées sur 16 octets (128 bits)
  - IPv6 a été mis au point principalement car il y a pénurie d'adresses

- Chaque adresse attribuée doit être unique sur le réseau

- Si le réseau est Internet, l'adresse IP attribuée à chaque PC doit être publique et unique au monde

Adresses IP
Généralités

- Si le réseau est un réseau local, l'adresse IP sera, en général, privée et doit être unique sur le réseau



Adresses IP

Classes IP (IPv4)

- Etant donné les besoins différents en taille de réseau, historiquement, plusieurs classes ont été définies; les classes A, B et C

Adresses IP

Classe A

- Pour la constituer, on met un 0 au bit de poids fort de l'octet de poids fort et on fait varier le reste des bits

  0xxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

  Si x=0, cela donne 0.0.0.0
  Si x=1, cela donne 127.255.255.255

Adresses IP

Classe A

- 0.0.0.0 est réservé comme adresse indéterminée, utilisée pour la passerelle par défaut, on prendra donc 1.0.0.0

  127.X.X.X est réservé pour l'adresse de boucle locale (localhost), on prendra donc 126.255.255.255

- La classe A s'étend donc de 1.0.0.0 à 126.255.255.255

Adresses IP

Classe B

- Pour la constituer, on met un 1 au bit de poids fort de l'octet de poids fort, suivi d'un 0 et on fait varier le reste des bits

  10xxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

  Si x=0, cela donne 128.0.0.0
  Si x=1, cela donne 191.255.255.255

- La classe B s'étend de 128.0.0.0 à 191.255.255.255

Adresses IP

Classe C

- Pour la constituer, on met deux 1 au bits de poids fort de l'octet de poids fort, suivi d'un 0 et on fait varier le reste des bits

  110xxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

  Si x=0, cela donne 192.0.0.0
  Si x=1, cela donne 223.255.255.255

- La classe C s'étend de 192.0.0.0 à 223.255.255.255

Adresses IP
Exercices

- Sur combien de bits sont codées les adresses IP en IPv4 ?

- De quelle classe font partie les adresses IP suivantes:
  12.223.255.254 177.177.177.177
  130.1.256.9 220.0.0.1
  127.0.0.1

- La classe D est réservée au multicast
  En suivant le même raisonnement que pour les autres classes, trouvez l'étendue de la classe D

Adresses IP

Adresses publiques et adresses privées

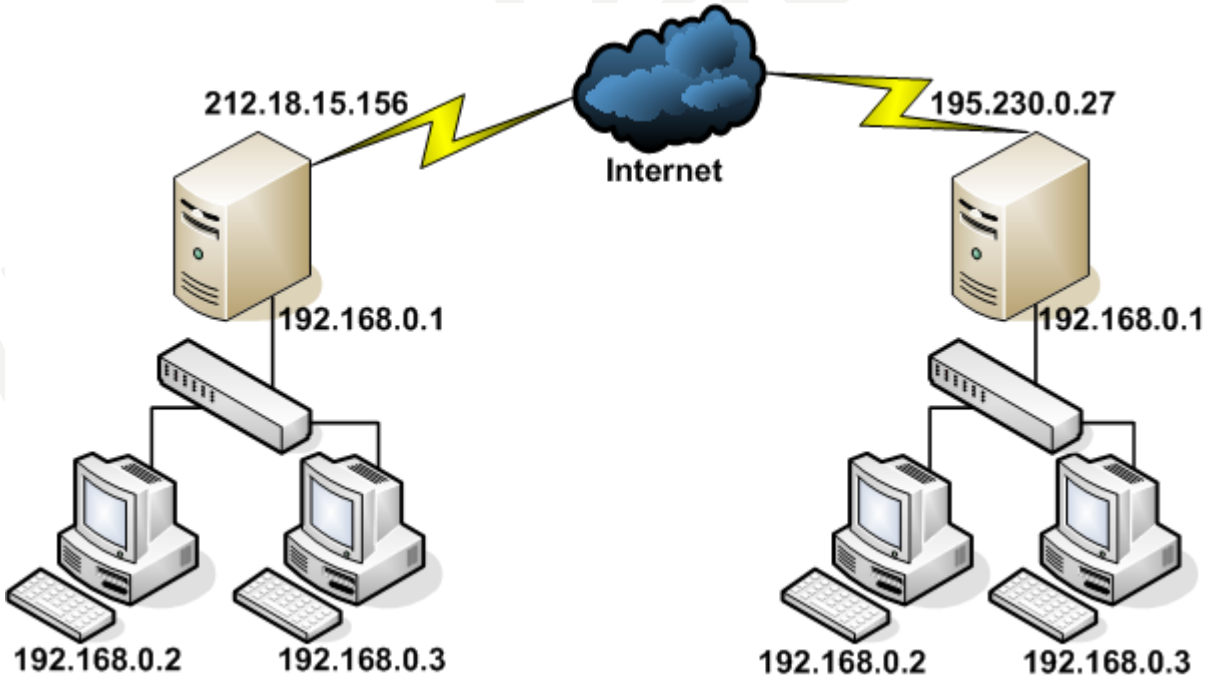
- Rappel: l'ICANN a réservé une poignée d'adresses dans chaque classe pour permettre d'affecter une adresse IP aux ordinateurs d'un réseau local relié à internet sans risquer de créer des conflits d'adresses IP sur le réseau public. Ces plages privées s'étendent:
  - pour la classe A: de 10.0.0.0 à 10.255.255.255
  - pour la classe B: de 172.16.0.0 à 172.31.255.255
  - pour la classe C: de 192.168.0.0 à 192.168.255.255
- A retenir !
- Remarque: ne pas oublier la plage d'autoconfiguration qui est aussi une plage privée: de 169.254.0.0 à 169.254.255.255

Adresses IP

Adresses publiques et adresses privées

- Toutes les adresses IP appartenant à ces plages n'ont pas d'existence sur Internet

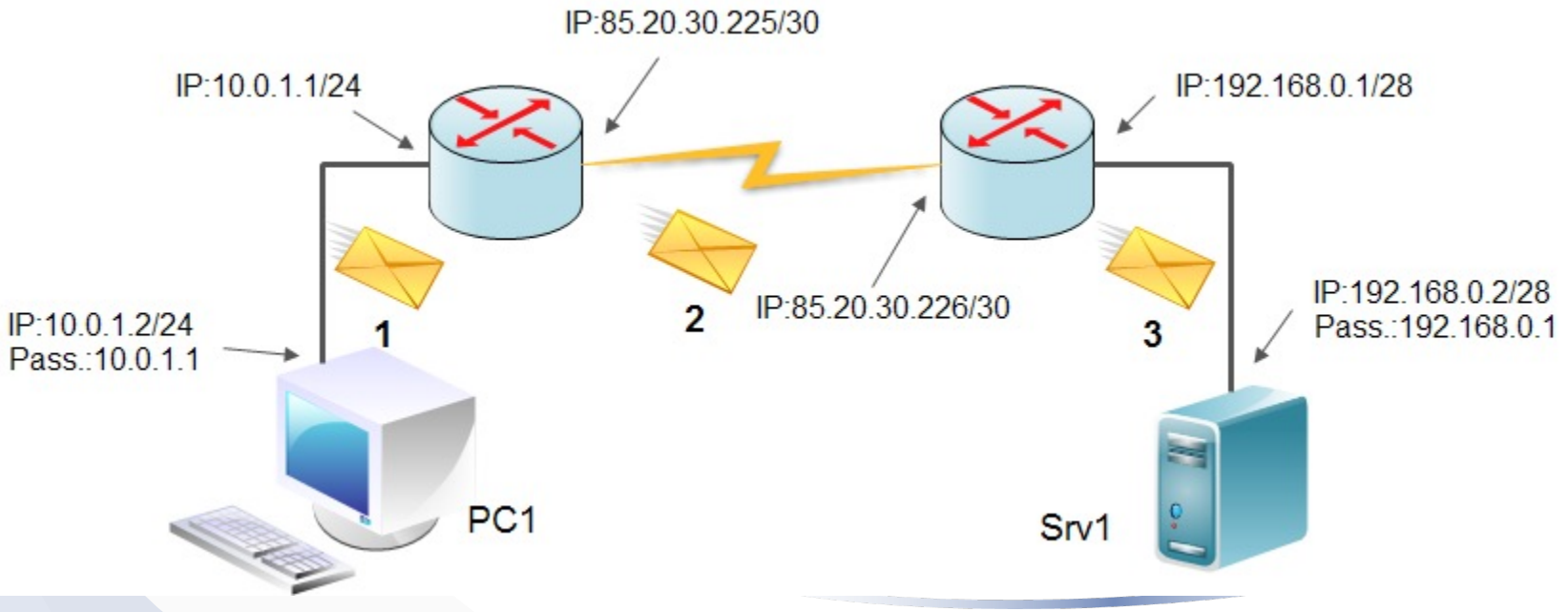
- Tous les réseaux locaux du monde (interconnectés via Internet) peuvent donc employer les mêmes adresses privées sans risque de conflit



Adresses IP

Adresses publiques et adresses privées

- Si ce sont deux réseaux locaux interconnectés directement par deux routeurs, ils devront employer des adresses privées différentes, sinon il y aura un problème de routage de paquets (vers où envoyer le paquet ?)



Adresses IP

NetID et HostID

- Afin d'aiguiller les datagrammes IP, les routeurs doivent être en mesure de distinguer les différents réseaux logiques

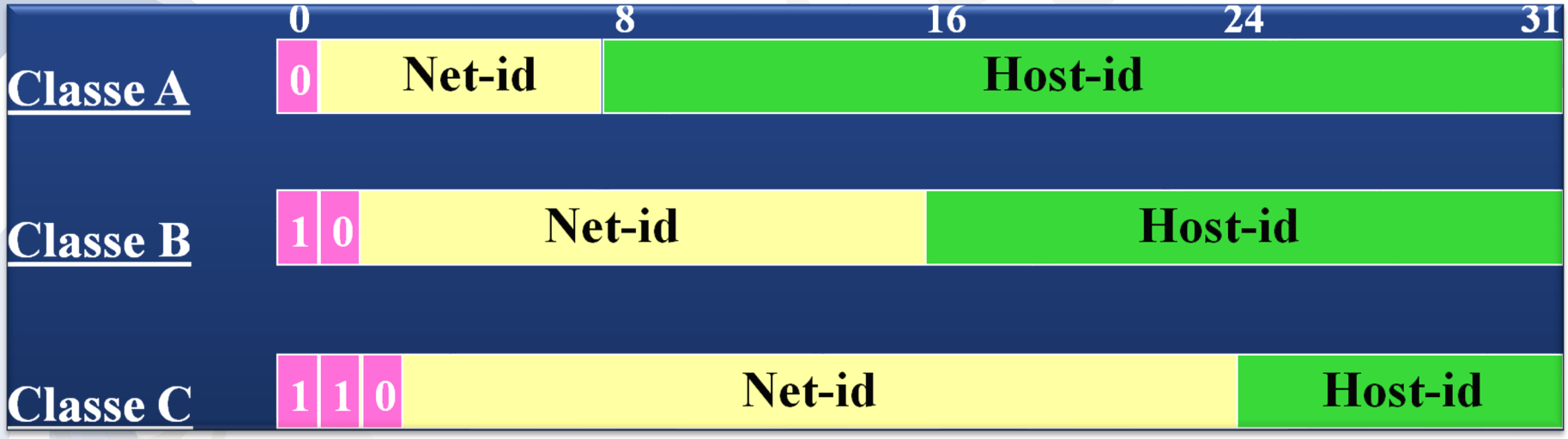
- On a donc décidé de structurer l'adresse IP de façon à ce qu'elle puisse refléter la distinction entre les différents réseaux logiques

- Un certain nombre de bits dans l'adresse IP sont utilisés pour identifier le réseau et les bits suivants permettent d'identifier l'hôte (le PC) au sein du réseau

Adresses IP

NetID et HostID

- Le nombre de bits réservé pour la partie NetID dépend de la classe:



Adresses IP

NetID et HostID

- Pour la classe A, 1 octet est réservé pour le NetID
- Il existe donc au maximum 126 réseaux (puisque la classe est définie de 1 à126 pour le premier octet) de classe A au monde
- Chaque réseau contient au maximum
  (256*256*256)-2 = 16777214 machines

Adresses IP

NetID et HostID

- Pour la classe B, 2 octets sont réservés pour le NetID
- Il existe donc au maximum 64*256 = 16384 réseaux (puisque la classe est définie de 128 à 191 pour le 1er octet) de classe B au monde
- Chaque réseau contient au maximum (256*256)-2 = 65534 machines

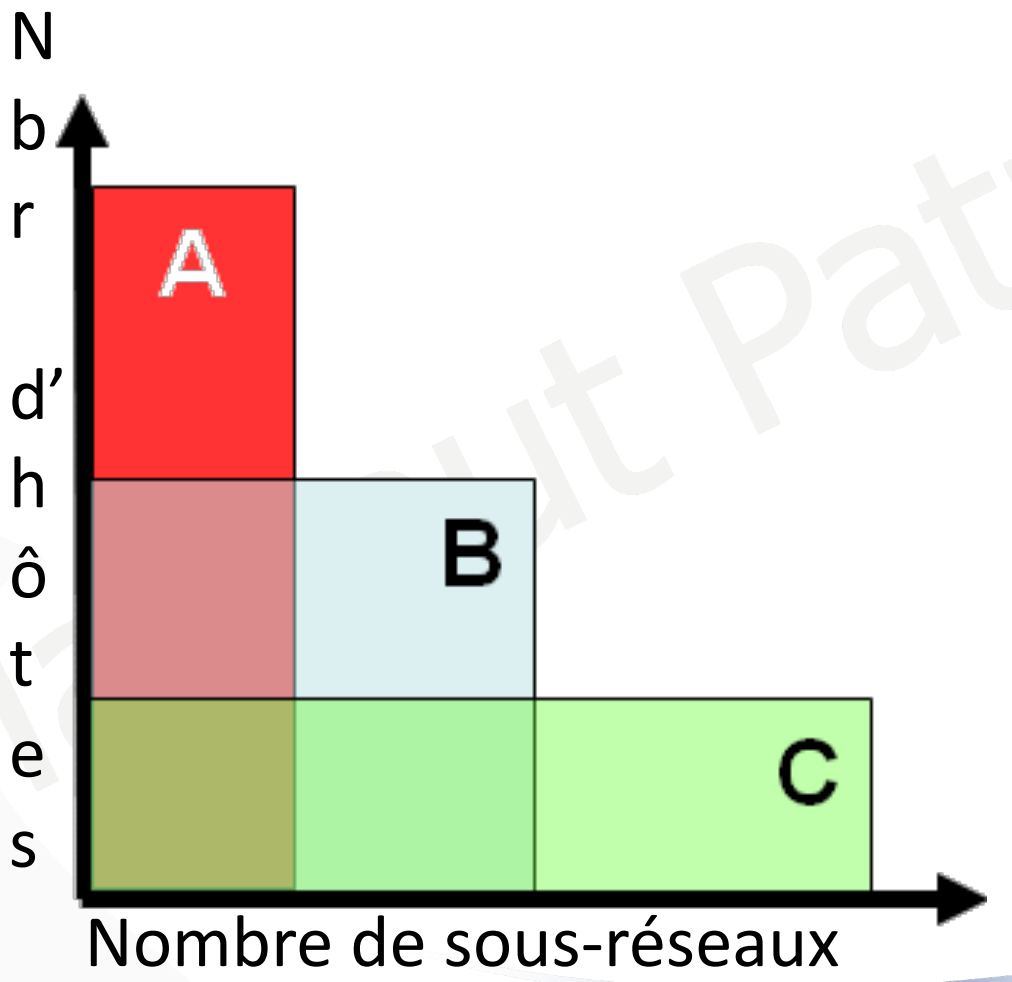
Adresses IP

NetID et HostID

- Pour la classe C, 3 octets sont réservés pour le NetID
- Il existe donc au maximum 32*256*256 = 2097152 réseaux (puisque la classe est définie de 192 à 223 pour le 1er octet) de classe C au monde
- Chaque réseau contient au maximum 256-2 = 254 machines

Adresses IP

NetID et HostID



Adresses IP

Le masque de sous-réseau

- 3 masques par défaut
  - 255.0.0.0 pour la classe A
  - 255.255.0.0 pour la classe B
  - 255.255.255.0 pour la classe C
- Ces masques correspondent à la séparation NetID – HostID
- Un masque est toujours une suite continue de bits à 1
- Le masque de sous-réseau est souvent spécifié en notant le nombre de bits à 1 dans le masque

Adresses IP

Le masque de sous-réseau

- Chaque fois que l'on trouve 255 dans le masque, l'octet correspondant au niveau des adresses IP doit toujours avoir la même valeur pour les PC d'un même réseau

  - Exemple: soit un PC d'adresse IP 10.0.2.50 et de masque de sous-réseau 255.0.0.0
    Les autres PC du même réseau devront avoir une adresse IP commençant par 10. …

Adresses IP

Le masque de sous-réseau

- Chaque fois que l'on trouve 0 dans le masque, l'octet correspondant au niveau des adresses IP peut prendre n'importe quelle valeur (valide !)
  - Exemple: soit un PC d'adresse IP 10.0.2.50 et de masque de sous-réseau 255.0.0.0

    Les autres PC du même réseau peuvent avoir comme adresse IP:
    10.0.2.51, 10.0.3.1, 10.25.0.1, …
    mais pas 10.0.2.50, 10.0.2.256 ou 12.0.2.51

    Pourquoi ?

Adresses IP

Le masque de sous-réseau

- Avant 1993, une adresse d'une classe était automatiquement associé au masque de cette classe
- Les protocoles de routage utilisés dans les routeurs, comme RIPv1 utilisaient ce principe
- Le masque n'était pas transmis dans les infos de routage !
- C'était des protocoles dits « classfull »
- Si on reprend les 5 adresses de classe A:
  10.0.0.1, 10.0.2.1, 10.0.0.10, 10.0.2.254, 10.0.0.15
  on obtient automatiquement un réseau de 5 PC car le masque utilisé est 255.0.0.0

Adresses IP

Le masque de sous-réseau

- Dans cette configuration, beaucoup d'adresses étaient gaspillées et avec le développement d'internet, une pénurie s'annonçait

- On est passé alors à des protocoles de routage dits « classless » comme RIPv2 qui intègrent le masque dans leur mise à jour

- Dès lors, une adresse d'une classe n'est plus automatiquement associé au masque correspondant

- Depuis 1993, une adresse IP doit donc toujours être spécifiée avec son masque de sous-réseau !

Adresses IP

Le masque de sous-réseau

- Une adresse de classe A peut donc être associé à un masque de sous-réseau de classe C

- L'avantage, c'est qu'on passe d'un seul réseau (très grand) à 65536 (256 * 256) réseaux (plus petits)

- En effet, si on prend le réseau 12.0.0.0 et un masque 255.0.0.0, on a un réseau. Si on prend maintenant le même réseau et un masque 255.255.255.0, on fixe 2 bytes supplémentaires, donc 256x256 réseaux de 256 adresses

- C'est le subnetting, et cela permet d'utiliser au mieux les adresses d'une classe

Adresses IP

Le masque de sous-réseau

- Si on reprend les 5 adresses de classe A:
  10.0.0.1, 10.0.2.1, 10.0.0.10, 10.0.2.254, 10.0.0.15

  avec un masque de classe C; 255.255.255.0, on obtient un réseau de 3 PC et un réseau de 2 PC

- **La classification des adresses IP en classe est donc obsolète !** (mais permet d'expliquer d'où on vient et dans Windows, le masque par défaut est toujours celui de la classe)

- Une adresse IP est toujours associée à un masque de sous-réseau

- On parle de routage interdomaine sans classe, dont l'abréviation est CIDR (Classless Inter-Domain Routing)

Adresses IP

Le masque de sous-réseau

- Dans certains cas particuliers, on associe une adresse de classe C avec un masque moins restrictif -> supernetting

- Cela permet, dans les routeurs, de faire un résumé de route et de réduire ainsi la taille des tables routage

Adresses IP

Le masque de sous-réseau

- Exemple: les routes des sous-réseaux 192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24 et 192.168.3.0/24 peuvent être collectivement agrégés avec un sur-réseau 192.168.0.0/22

- Le masque 255.255.252.0 (/22) est un masque de super-réseau

- En effet, /22 correspond pour le 3ème byte à 11111100
  La plage IP est donc de 192.168.0.0/22 à 192.168.3.255/22

Adresses IP

Adresses de réseau et de diffusion

- Rappel: dans chaque réseau, la première adresse disponible est l'adresse de réseau (network), utilisée dans les tables de routage et pour désigner un réseau, jamais sur une interface réseau

- La dernière adresse disponible est l'adresse de diffusion (broadcast), permettant d'envoyer un message à toutes les machines du réseau, elle n'est jamais utilisée pour adresser une interface réseau

- Exemple: soit le PC d'adresse 192.168.10.5/24
  - le masque de sous-réseau sera donc 255.255.255.0
  - L'adresse de réseau sera 192.168.10.0
  - L'adresse de diffusion sera 192.168.10.255

Adresses IP

Exemple

- Soit l'adresse 192.168.1.137/27
  - Le masque est 255.255.255.224
  - La partie réseau sera 192.168.1.128
  - Le partie hôte sera .137

- Par rapport à l'exemple de la dia précédente, la fin du net-id s'est déplacé vers la droite et n'est plus entre deux nombres mais dans un nombre, il faut donc descendre au niveau du bit pour déterminer le dernier byte de l'adresse réseau

Adresses IP

Exemple

- En effet, avec un masque 255.255.255.224
  - le dernier byte du masque est 11100000
  - le dernier byte de l'adresse est 10001001

- On regarde les bits d'adresse à 1 correspondant aux bits à 1 du masque, et on élimine le reste, ce qui donne, dans ce cas-ci: 10000000 , c'est-à-dire 128 en décimal

Adresses IP

Sous-réseaux (subnetting)

- Soit un réseau 193.18.34.0/24

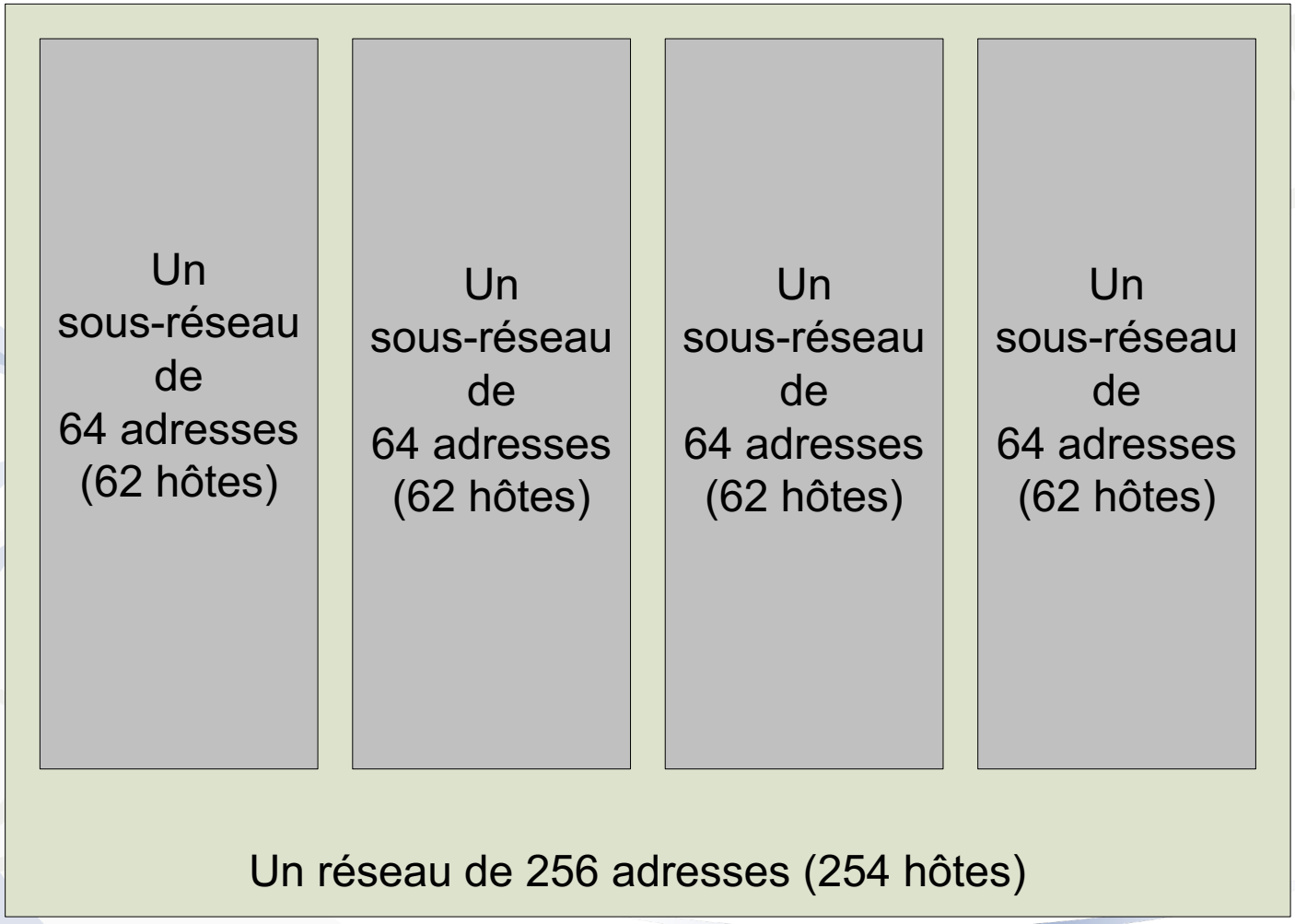
- Ce réseau de classe C a les caractéristiques suivantes:
  - masque de sous-réseau: 255.255.255.0
  - adresse de réseau: 193.18.34.0
  - adresse de diffusion: 193.18.34.255
  - adresse du premier PC: 193.18.34.1
  - adresse du dernier PC: 193.18.34.254
  - Nombre de PC au maximum dans le réseau: 254

Adresses IP
Sous-réseaux

- Si je veux répartir ces adresses dans 4 labos:
  - Soit je garde le réseau par défaut, mais les 4 labos ne seront pas indépendants
  - Soit je fais 4 sous-réseaux à partir du réseau d'origine, de manière à avoir 4 réseaux indépendants qui ne se voient pas
  - La deuxième solution est évidement la meilleure car:
    - cela permet de créer des réseaux petits et gérables
    - cela limite le trafic réseau
    - le trafic local à un segment de réseau peut être gardé localement , réduisant le trafic global

Adresses IP

Sous-réseaux



Adresses IP

Sous-réseaux

- Pour passer d'un réseau de 254 PC à quatre sous-réseaux (avec forcément moins de PC), je dois ajouter une contrainte supplémentaire au niveau du masque de sous-réseau, seul élément sur lequel je peux agir

- Le masque par défaut est : 255.255.255.0

- Le net-id est défini par les bits à 1 du masque, c'est la partie réseau de l'adresse (193.18.34)

- Si je veux augmenter le nombre de sous-réseaux, je dois étendre le net-id en rajoutant des bits à 1 dans mon masque de s-r

Adresses IP

Sous-réseaux

- Je vais donc agir sur le dernier octet (0) et ajouter des bits de réseau (à 1)
- Combien ? -> cela dépend du nombre de sous-réseaux
- Pour contrôler 4 sous-réseaux, il faut rajouter 2 bits à 1 dans le masque de sous-réseaux car $2^2=4$
- Si on veut 8 s-r, on prendra 3 bits réseaux, et ainsi de suite …
- Remarque: retenez $2^5=32$ et $2^{10}=1024$, vous déduirez les autres valeurs de celles-ci

Adresses IP

Sous-réseaux

- Une adresse IPv4 comporte 32 bits et ces bits sont soit des bits réseaux (bits à 1 dans le masque et situés à gauche), soit des bits machines (bits à 0 dans le masque et situés à droite)

- La règle est que x bits contrôlent 2^X items (s-r ou adresses)

- Exemple: si j'ai besoin de 12 machines dans mon réseau, il me faut 14 adresses (adr. réseau + diffusion), j'en prend donc 16 (il faut que ça corresponde à un poids binaire) et comme $16=2^4$, je réserve donc 4 bits machines, ce qui me laisse 32-4=28 bits réseaux

Adresses IP
Sous-réseaux

- Revenons à notre exemple

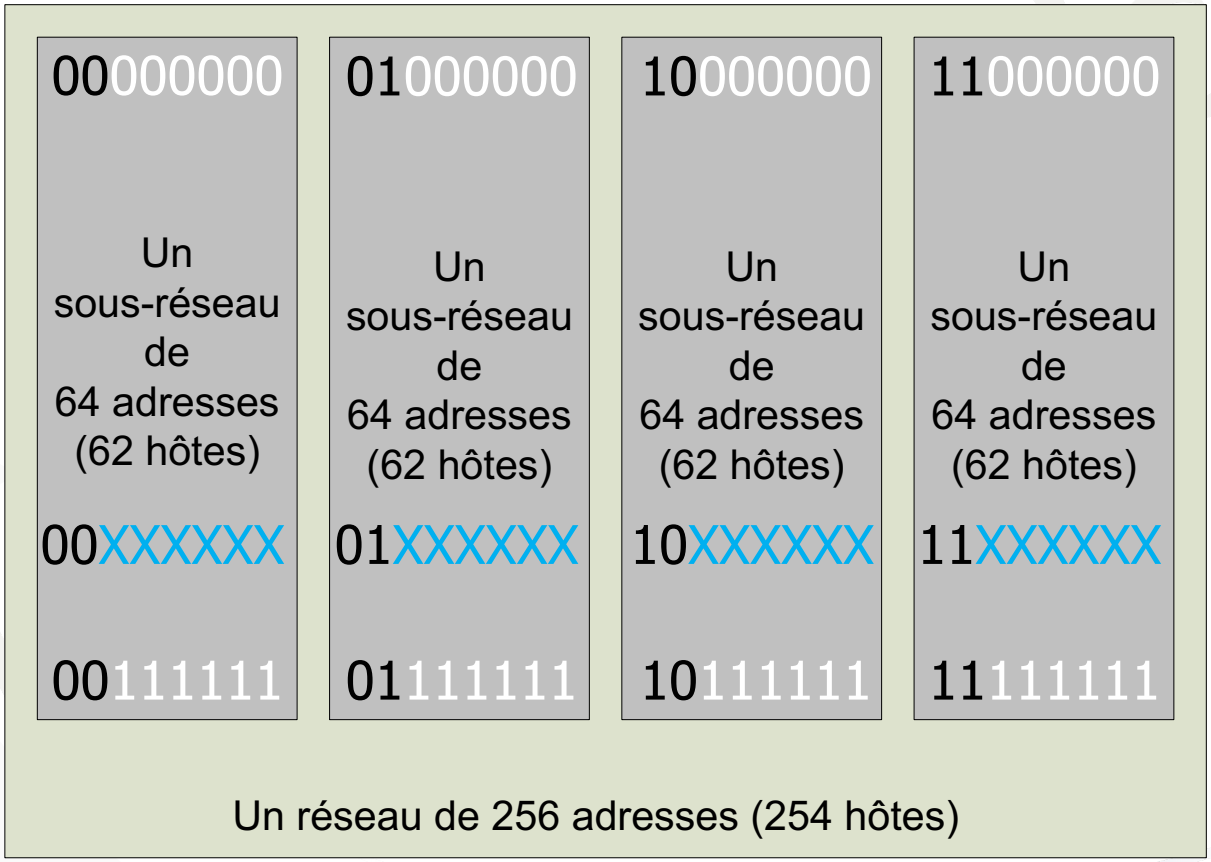
- Le dernier octet du masque de sous-réseau va donc passer de 00000000 à 11000000, ce qui donne 192 en décimal

- Si l'on fait varier ces 2 bits au niveau de l'adresse, on obtient 4 cas possibles:
  - 00
  - 01
  - 10
  - 11

Adresses IP

Sous-réseaux

- Ces 4 cas possibles définissent les 4 sous-réseaux



Adresses IP

Sous-réseaux

- Attention, un masque de sous-réseau est toujours une suite continue de bits à 1 (donc on aurait pas pu passer de 00000000 à 00000011)
- Le masque de sous-réseau passe de 255.255.255.0 à 255.255.255.192
- C'est le même masque pour les 4 sous-réseaux
- Pour définir chaque sous réseau, et pour chaque combinaison (00,01,10 et 11), on fait varier les 6 bits restants du dernier octet d'adresse

Adresses IP

Sous-réseaux

- Pour 00, cela donne:
  193.18.34.00XXXXXX
  Si x vaut 0, on obtient 193.18.34.0
  Si x vaut 1, on obtient 193.18.34.63

- Le 1er sous-réseau a donc une plage s'étalant de 193.1834.0 à 193.18.34.63
  - masque de sous-réseau: 255.255.255.192
  - adresse de sous-réseau: 193.18.34.0
  - adresse de diffusion: 193.18.34.63
  - Nombre de PC au maximum dans le sous-réseau: 62

Adresses IP

Sous-réseaux

- Pour 01, cela donne:
  193.18.34.01XXXXXX
  Si x vaut 0, on obtient 193.18.34.64
  Si x vaut 1, on obtient 193.18.34.127

- Le 2ème sous-réseau a donc une plage s'étalant de 193.1834.64 à 193.18.34.127
  - masque de sous-réseau: 255.255.255.192
  - adresse de sous-réseau: 193.18.34.64
  - adresse de diffusion: 193.18.34.127
  - Nombre de PC au maximum dans le sous-réseau: 62

Adresses IP

Sous-réseaux

- Pour 10, cela donne:
  193.18.34.10XXXXXX
  Si x vaut 0, on obtient 193.18.34.128
  Si x vaut 1, on obtient 193.18.34.191

- Le 3ème sous-réseau a donc une plage s'étalant de 193.1834.128 à 193.18.34.191
  - masque de sous-réseau: 255.255.255.192
  - adresse de sous-réseau: 193.18.34.128
  - adresse de diffusion: 193.18.34.191
  - adresse du premier PC: 193.18.34.129
  - adresse du dernier PC: 193.18.34.190

Adresses IP
Sous-réseaux

- Pour 11, cela donne:
  193.18.34.11XXXXXX
  Si x vaut 0, on obtient 193.18.34.192
  Si x vaut 1, on obtient 193.18.34.255

- Le 4ème sous-réseau a donc une plage s'étalant de 193.18.34.192 à 193.18.34.255
  - masque de sous-réseau: 255.255.255.192
  - adresse de sous-réseau: 193.18.34.192
  - adresse de diffusion: 193.18.34.255
  - adresse du premier PC: 193.18.34.193
  - adresse du dernier PC: 193.18.34.254

Adresses IP

Remarques

- Le masque de sous-réseau est le même pour tous les sous-réseaux
- En « classfull », comme le masque n'est pas transmis, on ne sait pas distinguer l'adresse de réseau d'une plage IP et l'adresse de sous-réseau du 1er sous-réseau
- Idem pour l'adresse de diffusion de la plage IP et celle du dernier sous-réseau
- On éliminait donc systématiquement le 1er et le dernier sous-réseau (RFC950), ce qui est maintenant obsolète depuis qu'on est en classless (RFC1878)

Adresses IP

Remarques

- Exemple: le réseau 192.168.0.0 /24

  Adr. de réseau: 192.168.0.0
  Adr. de diffusion: 192.168.0.255

  Si on divise en 8 sous-réseaux (de 32 adr), l'adr. réseau du 1er sous-réseau sera aussi 192.168.0.0 (mais /27)
  et l'adr. de diffusion du dernier sous-réseau sera aussi 192.168.0.255 (mais /27)

- Donc si le masque n'est pas diffusé, ça pose problème, ce qui n'est plus le cas puisqu'on diffuse le masque à chaque fois

Adresses IP
Exercices

- 1. Soit le réseau 192.234.15.0 /24 . On veut créer 12 sous-réseaux
  Décrivez le 5ème sous-réseau (plage IP, masque, nbr d'adresses et de PC/s-r)
- 2. Soit le réseau 183.23.0.0 /16 . On veut créer 27 sous-réseaux
  Décrivez le 10ème sous-réseau
- 3. Soit le réseau 12.0.0.0 /8 . On veut créer 60 sous-réseaux
  Décrivez le 31ème sous-réseau
- 4. Soit le réseau 220.18.15.0 /24 . On veut obtenir 30 PC par sous-réseau
  Combien de sous-réseaux allons-nous obtenir ? Décrivez le dernier s-r

Adresses IP

Exercices

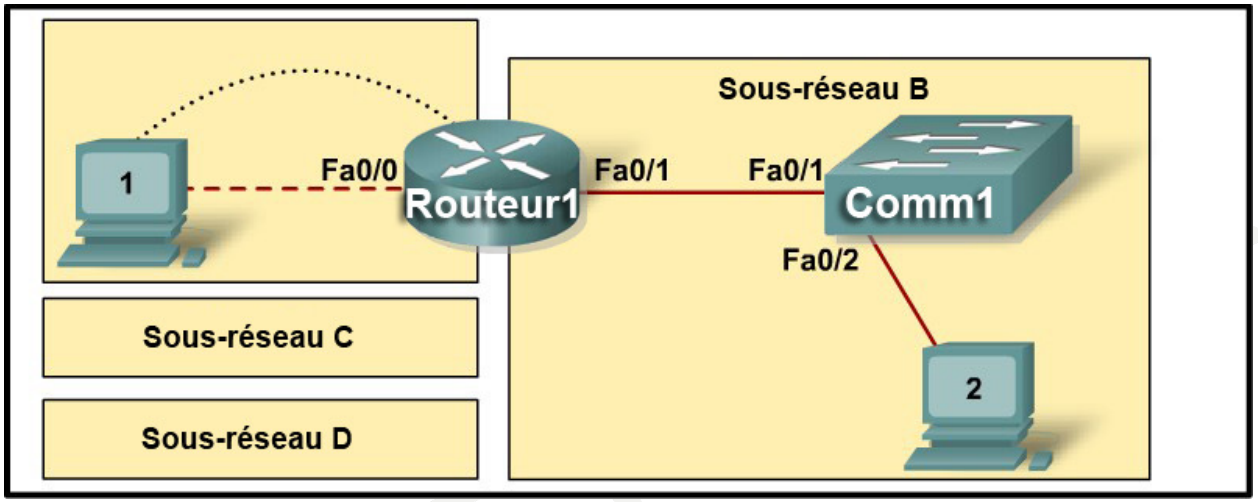
- 5. Soit le réseau 172.15.0.0/16. Configurez des sous-réseaux de manière à avoir 4092 PC par sous-réseau.

  Combien de sous-réseaux obtient t-on ?

- 6. Soit le réseau 162.10.0.0/16. On veut créer 256 sous-réseaux.

  Décrivez le 201ème sous-réseau

Adresses IP
Exercices



- 7. A partir de l'adresse 193.18.15.0/24, concevez un modèle d'adressage IP qui remplisse les conditions suivantes :

Sous-réseau A	2 hôtes
Sous-réseau B	14 hôtes
Sous-réseau C	30 hôtes
Sous-réseau D	6 hôtes

Adresses IP

Exercices

Pour chaque sous-réseau, donnez:

- Nombre de bits dans le sous-réseau
- Nouveau masque IP (binaire)
- Nouveau masque IP (décimal)
- Nombre maximal de sous-réseaux utilisables (y compris le sous-réseau 0)
- Nombre d'hôtes utilisables par sous-réseau
- Sous-réseau IP
- Première adresse hôte IP
- Dernière adresse hôte IP

Adresses IP
Exercices

- 8. Soit l'adresse de réseau 193.255.255.64/26, utilisez le 5ème sous-réseau d'un total de 8

  Combien d'hôtes seront disponibles dans ce sous-réseau ?
  Indiquez les paramètres réseaux du premier PC du sous-réseau sachant qu'il utilise la première adresse disponible et que la passerelle utilise la dernière
  La passerelle aura-t-elle la même adresse pour les autres PC du sous-réseau ?

Adresses IP

Exercices

- 9. Soit l'adresse 223.15.89.45/27

  De quelle plage de sous-réseau fait-elle partie ?

  En combien de sous-réseaux viables maximum, peut être divisé ce (sous-)réseau ?

  Quel sera le nombre d'hôtes par sous-réseau ?

Adresses IP
Exercices

- 10. Soit l'adresse 220.188.10.211/26. De quelle plage de sous-réseau fait-elle partie ? Divisez ce sous-réseau en 5 sous-réseaux. Décrivez l'avant-dernier sous-réseau à partir du nombre nécessaire de sous-réseaux
- 11. Soit l'adresse 200.16.75.200/28. De quelle plage de sous-réseau fait-elle partie ? Divisez ce sous-réseau en 4 sous-réseaux. Décrivez l'avant-dernier sous-réseau à partir du nombre nécessaire de sous-réseaux
- 12. Soit l'adresse 130.0.89.211/21. De quelle plage de sous-réseau fait-elle partie ? Divisez ce sous-réseau en 50 sous-réseaux. Décrivez le sous-réseau 25
- 13. Soit l'adresse 145.15.178.233/19. De quelle plage de sous-réseau fait-elle partie ? Divisez ce sous-réseau en sous-réseaux de 7 machines. Combien de s-r allons-nous obtenir. Décrivez le 10ème sous-réseau

Adresses IP
A retenir

- IPv4:
  - Une adresse se compose de 4 octets et utilise la représentation décimale pointée
  - Elle se décompose en une partie réseau et une partie hôte
  - Le masque de sous-réseau permet de différencier la partie réseau de la partie hôte
  - Pour chaque (sous-)réseau, une adresse de (sous-) réseau et une adresse de diffusion sont définies

Adresses IP

A retenir

- IPv4:
  - 3 classes IP ont été définies à l'origine avec un masque de sous-réseau par défaut pour chacune
  - Vu le manque d'adresses, on est passé d'un routage par classe (classfull) à un routage sans classe où un suffixe CIDR (Classeless Inter-Domain Routing) indique le nombre de bits utilisés pour le réseau (Ex.: /24)
  - Les adresses IP sont soit publiques (visibles sur Internet), soit privées (uniquement pour les réseaux locaux)

Adresses IP
A retenir

- IPv4:
  - Une plage d'adresse est également prévue pour l'auto-configuration IP des PC clients DHCP qui n'obtiennent pas d'adresses (plage de 169.254.0.0 à 169.254.255.255)
  - Chaque hôte possède une adresse de bouclage interne qui commence par 127 (traditionnellement 127.0.0.1)

Adresses IP
A retenir

- IPv4:
  - 3 catégories d'adresses:
    - Monodiffusion (Unicast): identifie une interface unique. Le paquet est envoyé à un ordinateur spécifique.
    - Multidiffusion (Multicast): identifie de 0 à n interfaces. Le paquet est envoyé à un groupe d'ordinateurs. On utilise des adresses de la classe D (224. … à 239. …)
    - Diffusion (Broadcast): identifie n interfaces. Le paquet est envoyé à tous les ordinateurs du (sous-)réseau. On utilise l'adresse de diffusion du (sous-)réseau.

Le VLSM
Introduction

- VLSM = **V**ariable **L**enght **S**ubnetwork **M**ask
  Masque de sous-réseau de longueur variable

- VLSM a été développé pour permettre de multiplier les niveaux de 'subnet' au sein d'un même réseau, c'est à dire que le masque de sous-réseau ne reste plus figé et identique pour chaque sous-réseau (plusieurs masques de sous-réseau dans le même réseau)

- Cela permet donc d'optimiser l'utilisation des adresses IP dans un 'range' (important dans un 'range' public)

Le VLSM

Introduction

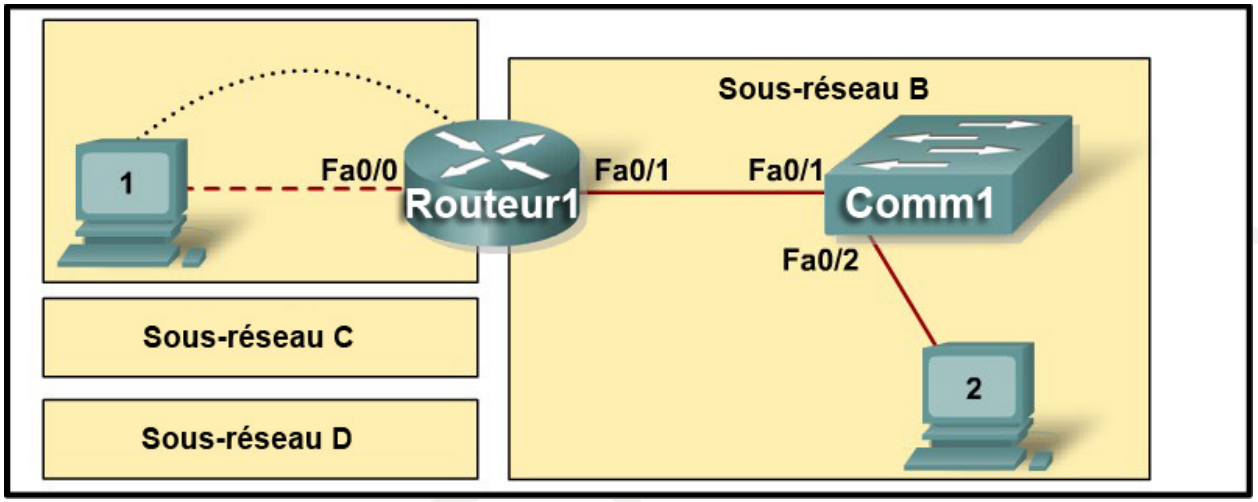
- En quelque sorte, "on subnette un subnet", ce qui va augmenter l'efficacité d'adressage et va permettre de "résumer les routes" (route summarization)

- VLSM est en quelque sorte une extension de CIDR

- Ces deux notions sont en fait étroitement lié, la seule différence est que VLSM est destiné à un réseau interne propre à une organisation (mais généralement constitué d'adresses IP publiques), tandis que CIDR lui peut agir dans le réseau internet (mondial)

Le VLSM

Exemple



- Reprenons l'exercice n° 7, mais optimisons les résultats
- A partir de l'adresse 193.18.15.0/24, concevez un modèle d'adressage IP qui remplisse les conditions suivantes :

Sous-réseau A	2 hôtes
Sous-réseau B	14 hôtes
Sous-réseau C	30 hôtes
Sous-réseau D	6 hôtes

Le VLSM
Exemple

- Pour l'ensemble des sous-réseaux, nous avons besoin de 4+16+32+8 = 60 adresses
  (il faut ajouter pour chaque sous-réseau, les adresses de sous-réseau et de diffusion)

  Avec l'adresse réseau 193.18.15.0/24, la plage IP est de 256 adresses disponibles, donc on peut continuer

Le VLSM
Exemple

- Nous allons commencer notre plan d'adressage par le plus gros sous-réseau, dans notre cas
  32 adresses (30 hôtes)
  $32 = 2^5$ -> 5 bits machines sont nécessaires

  Sur les 32 bits d'adresse, 5 seront nécessaires pour adresser les machines, ce sera le host-id. Le reste, c'est-à-dire les 27 premiers bits, sont les bits réseaux, le net-id

  -> de 193.18.15.0 / 27 à 193.18.15.31 / 27

Le VLSM
Exemple

- Ensuite, nous avons besoin de 16 adresses

  16 = 2^4 -> 4 bits machines sont nécessaires

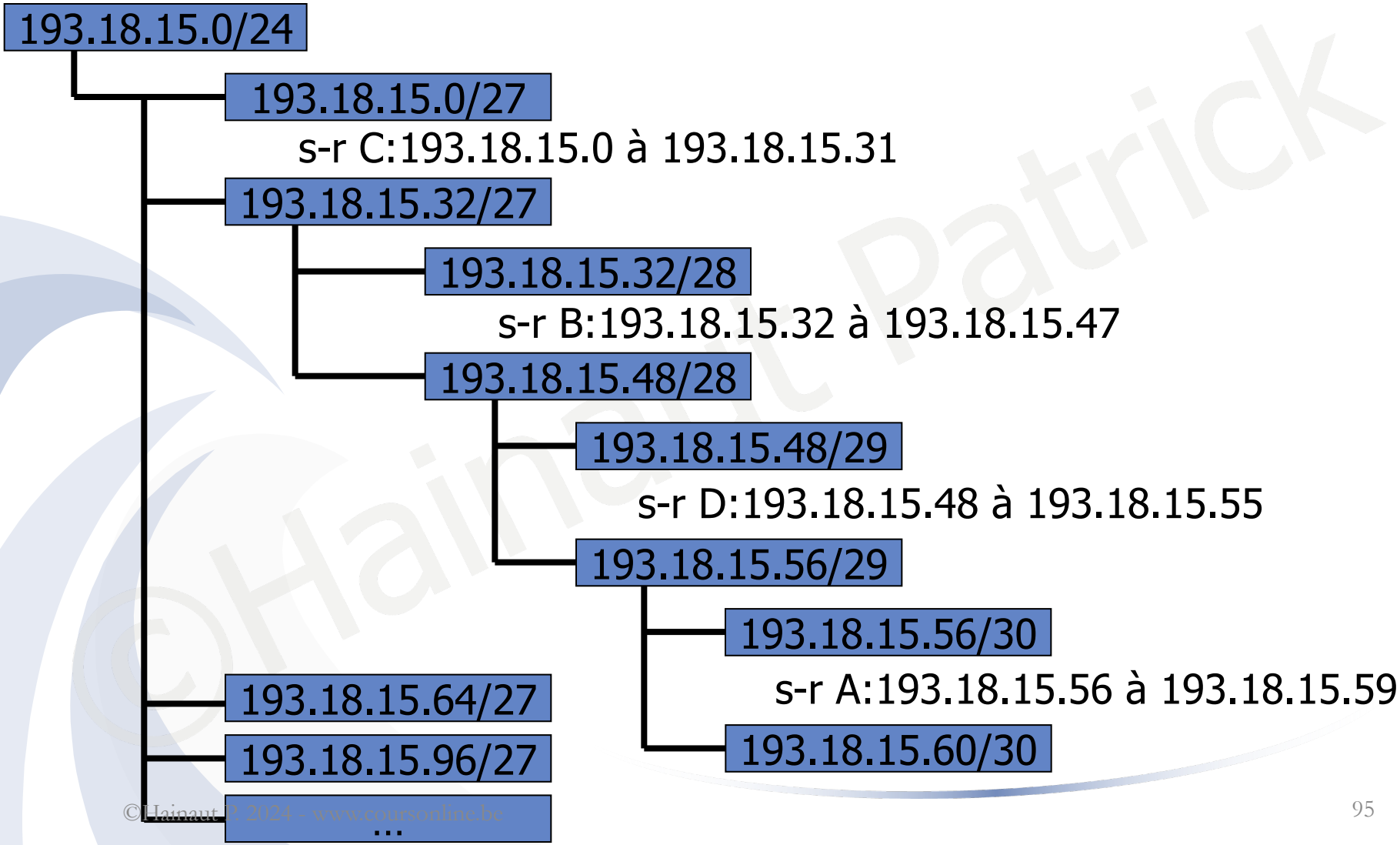
  -> de 193.18.15.32 / 28 à 193.18.15.47 / 28

- Puis, de 8 adresses
  -> de 193.18.15.48 / 29 à 193.18.15.55 / 29

- Et enfin, de 4 adresses
  -> de 193.18.15.56 / 30 à 193.18.15.59 / 30

Le VLSM

Exemple



Le VLSM

Sans VLSM

1 niveau de s-r

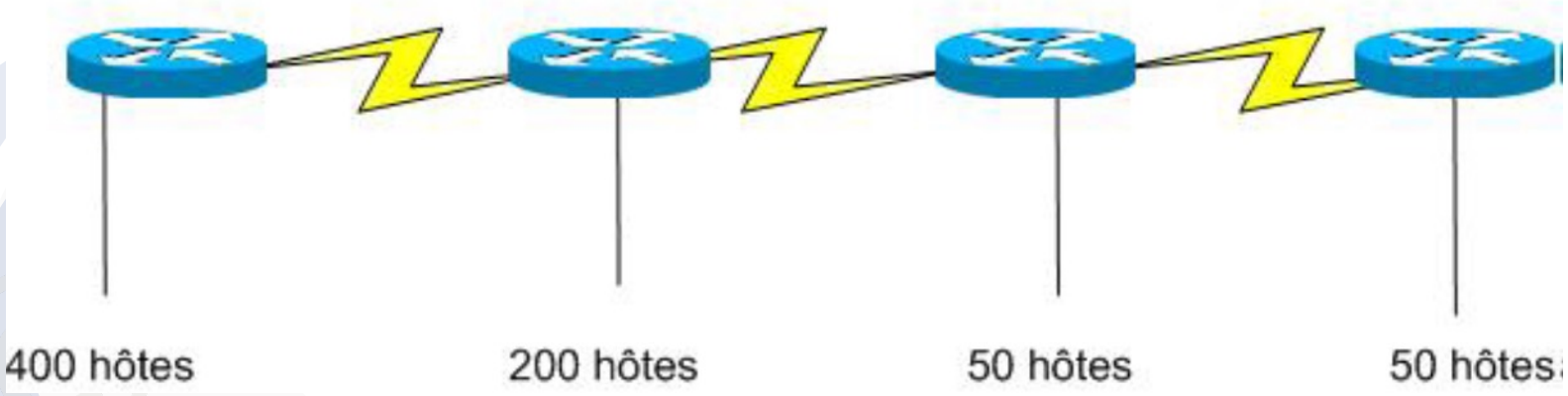
Avec VLSM

4 niveaux de s-r

Le VLSM

Exercice 1

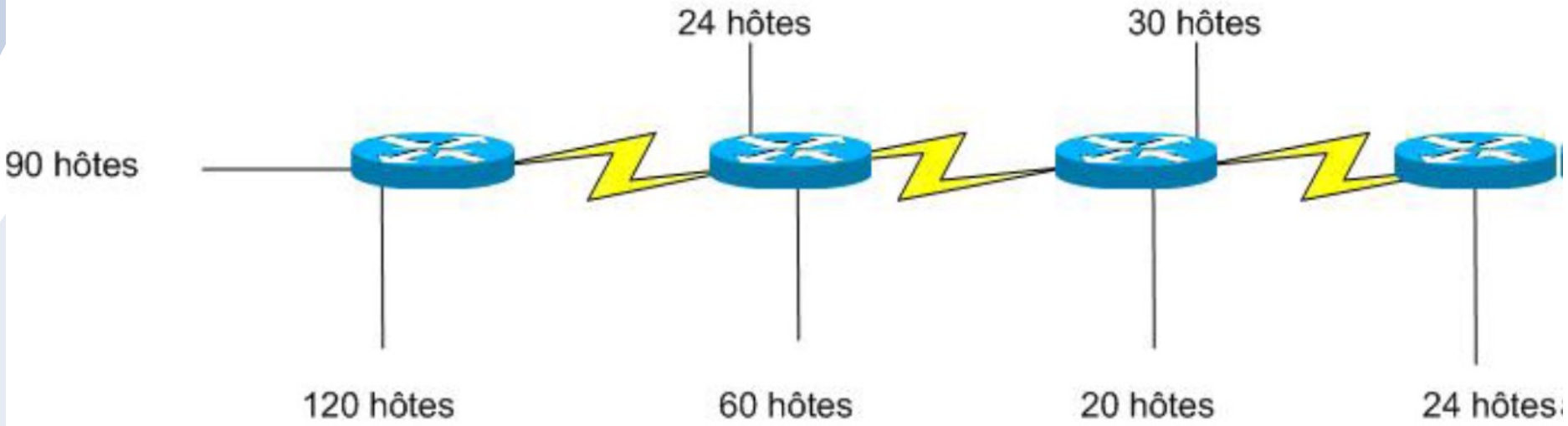
- Combien y a-t-il de réseaux sur la figure ?
- L'adresse CIDR 193.168.24.0/22 est attribuée et doit prendre en charge le réseau indiqué dans le schéma
- Créez un système d'adressage conforme aux exigences du schéma



Le VLSM

Exercice 2

- Combien y a-t-il de réseaux sur la figure ?
- L'adresse CIDR 193.168.30.0/23 est attribuée et doit prendre en charge le réseau indiqué dans le schéma
- Créez un système d'adressage conforme aux exigences du schéma



Le VLSM

Exercice 2

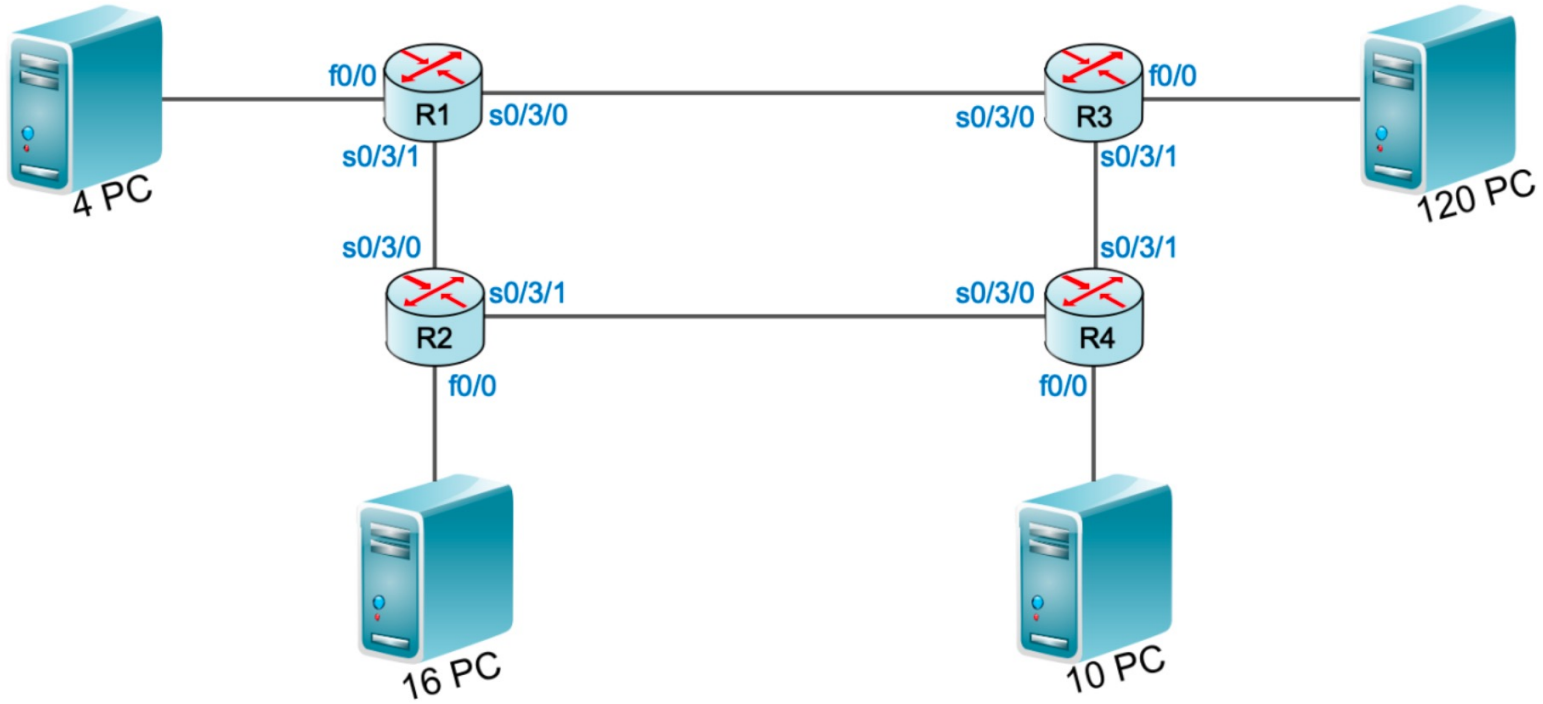
- Dans l'exercice précédent, aurait-on pu utiliser la méthode classique ?

- Pourquoi ?

Le VLSM

Exercice 3

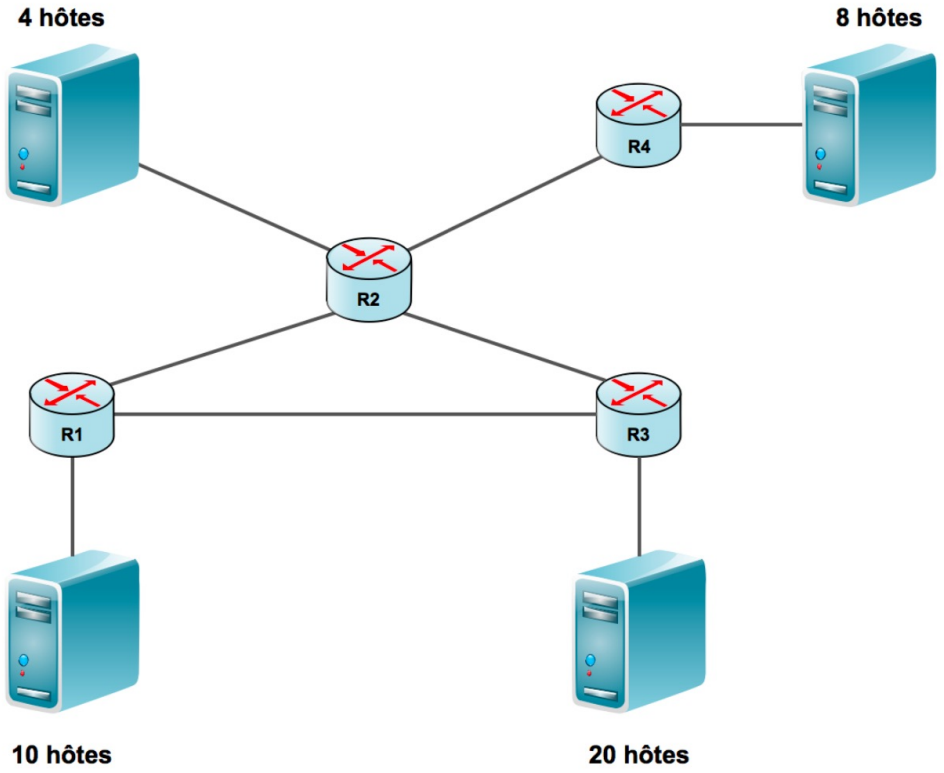
- Combien y a-t-il de réseaux sur la figure ?
- Donnez la plage IP de chaque réseau à partir de l'adresse 172.30.128.0/24



Le VLSM

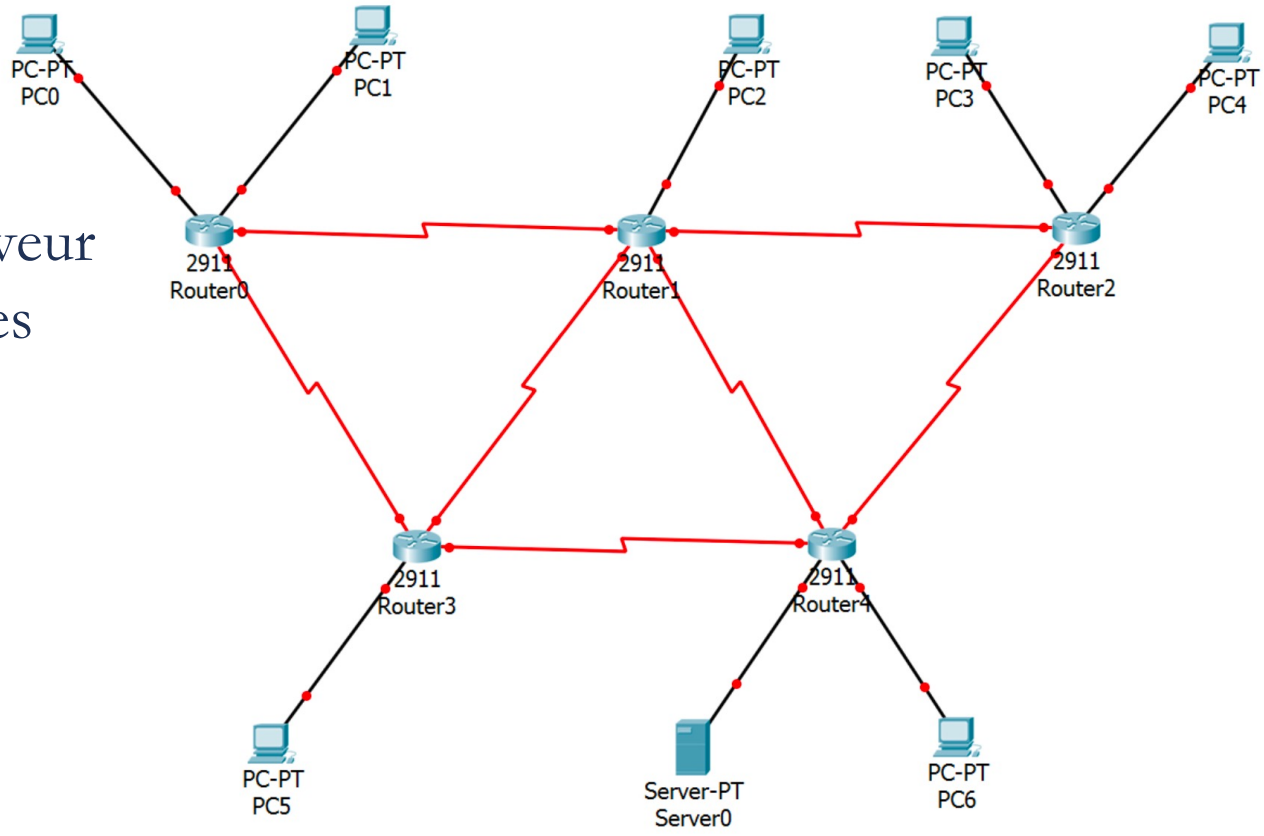
Exercice 4

- Combien y a-t-il de réseaux sur la figure ?
- Donnez la plage IP de chaque réseau à partir de l'adresse 95.30.12.128/25



Le VLSM
Exercice 5

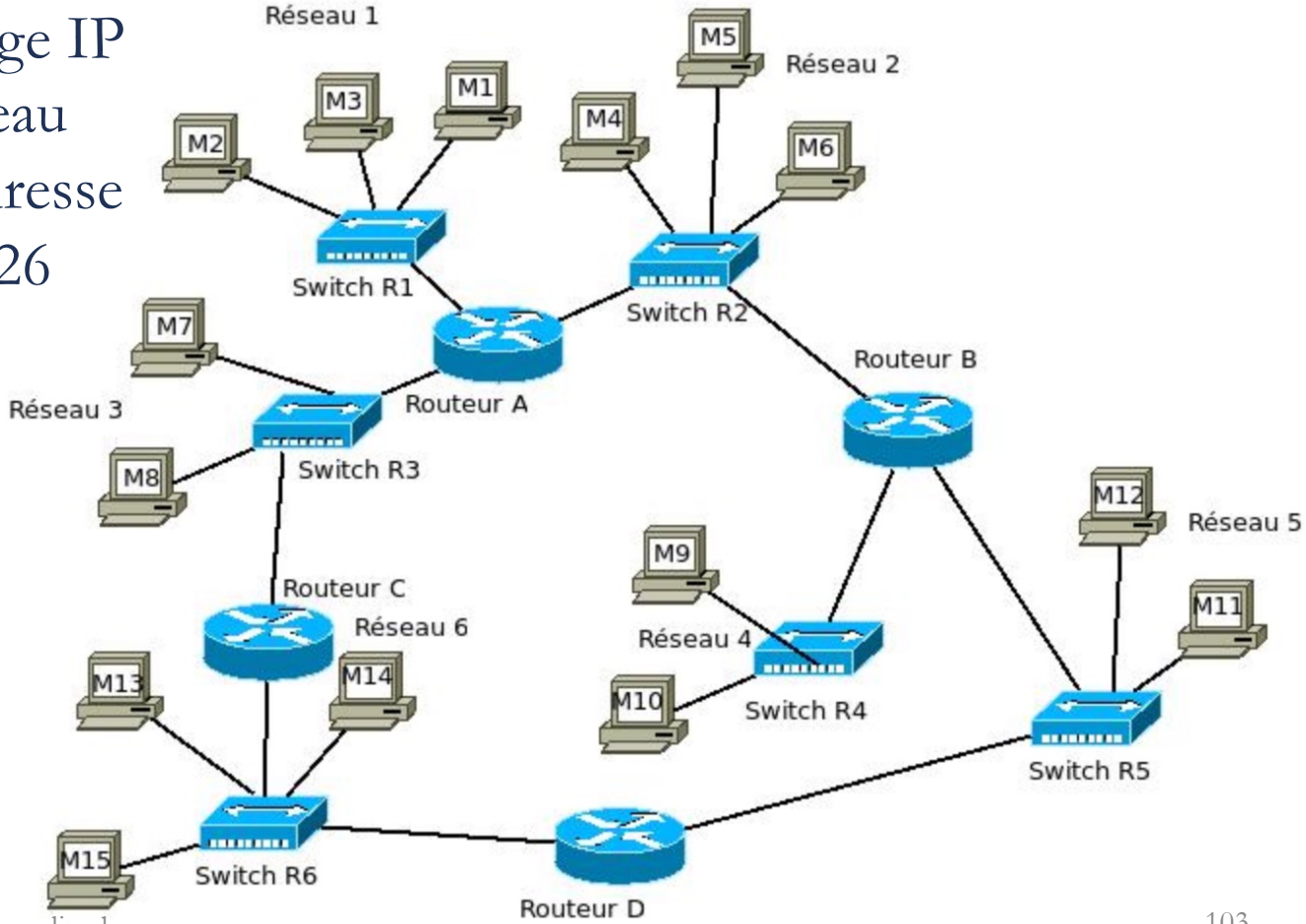
- Combien y a-t-il de réseaux sur la figure ?
- Chaque réseau de PC contient 6 hôtes
- Le réseau du serveur comprend 2 hôtes
- Donnez la plage IP de chaque réseau à partir de l'adresse 172.30.128.0/25



Le VLSM

Exercice 6

- Combien y a-t-il de réseaux sur la figure ?
- Donnez la plage IP de chaque réseau à partir de l'adresse 85.30.128.64/26



Liens
Cours

- Quelques liens de cours bien faits:

  http://www.linux-france.org/prj/edu/archinet/systeme/index.html

  http://fr.wikibooks.org/wiki/Réseaux_TCP/IP_:_adressage_IP_v4

Liens

Exercices

- Quelques liens pour vous exercer:
  http://fanocayoo.free.fr/exo_ip.htm

  http://www.reseaumaroc.com/files/Controle%206.pdf
  (mais n'enlevez pas les 1er et dernier sous-réseaux comme préconisé par ce pdf …)

- Un calculateur très pratique:
  http://www.vlsm-calc.net/

Outils réseaux

netstat

- Cet utilitaire permet d'afficher, entre autre, la table de routage locale et les connexions actives

- Faites un netstat -r pour voir la table de routage

- Equivalent à route print et à route -n sous linux

- Faites un netstat -n pour voir les connexions actives

- Tirez quelques conclusions de vos observations

Outils réseaux

wireshark

- wireshark est un analyseur de trames libre qui sert à l'étude et au dépannage des réseaux

- Il existe pour différents OS dont Windows, OSX et Linux

- Nous intéresseront ici sommairement à la version Windows, la version Linux étant vue dans Manip6

- On peut le télécharger sur https://www.wireshark.org/download.html

Outils réseaux

wireshark

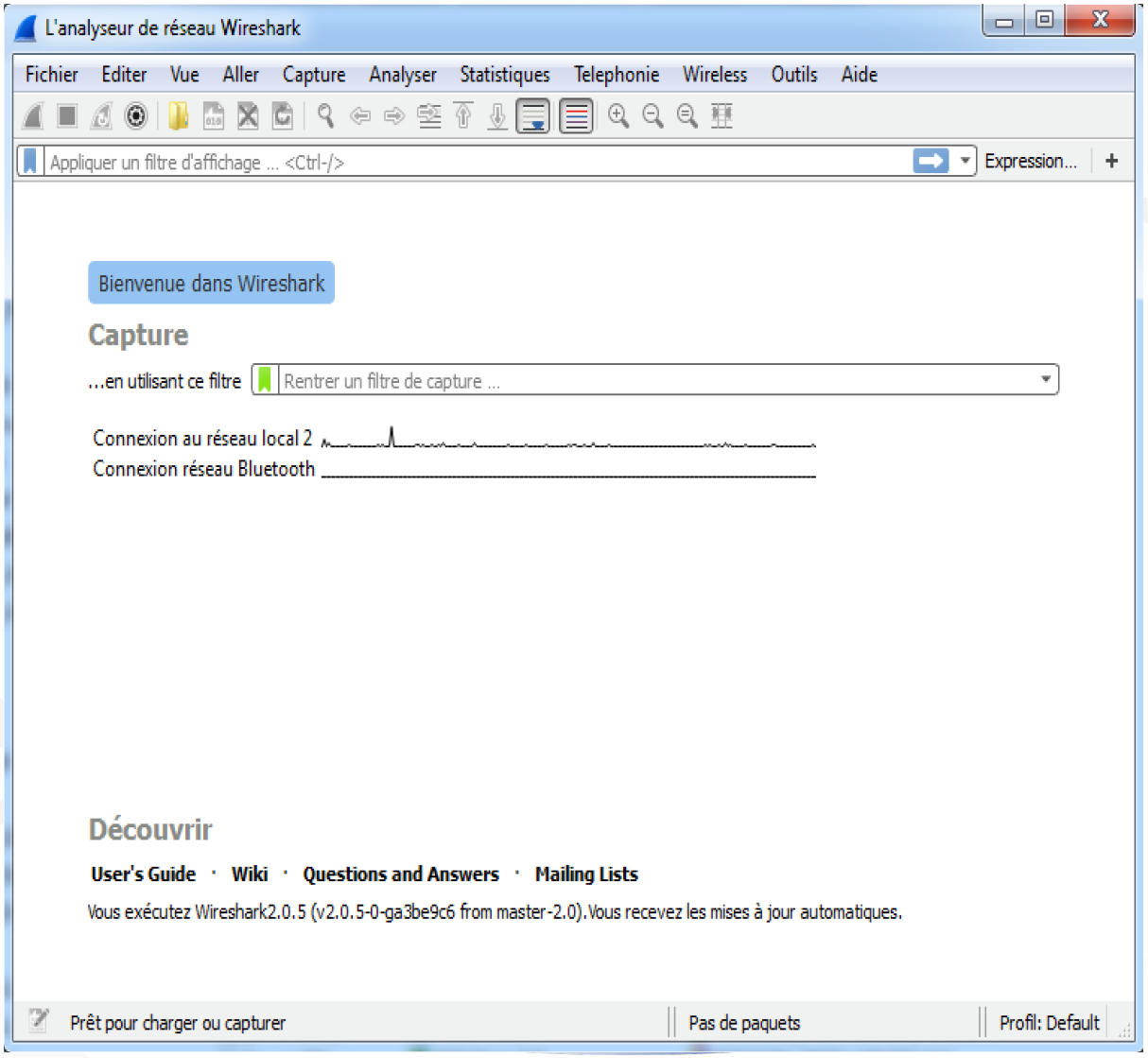
- L'installation se fait en deux étapes: d'abord wireshark, notre analyseur réseau et puis winpcap, le moteur de capture et de filtrage des paquets

- L'installation peut se faire par défaut, rien de particulier n'est à signaler

Outils réseaux

wireshark

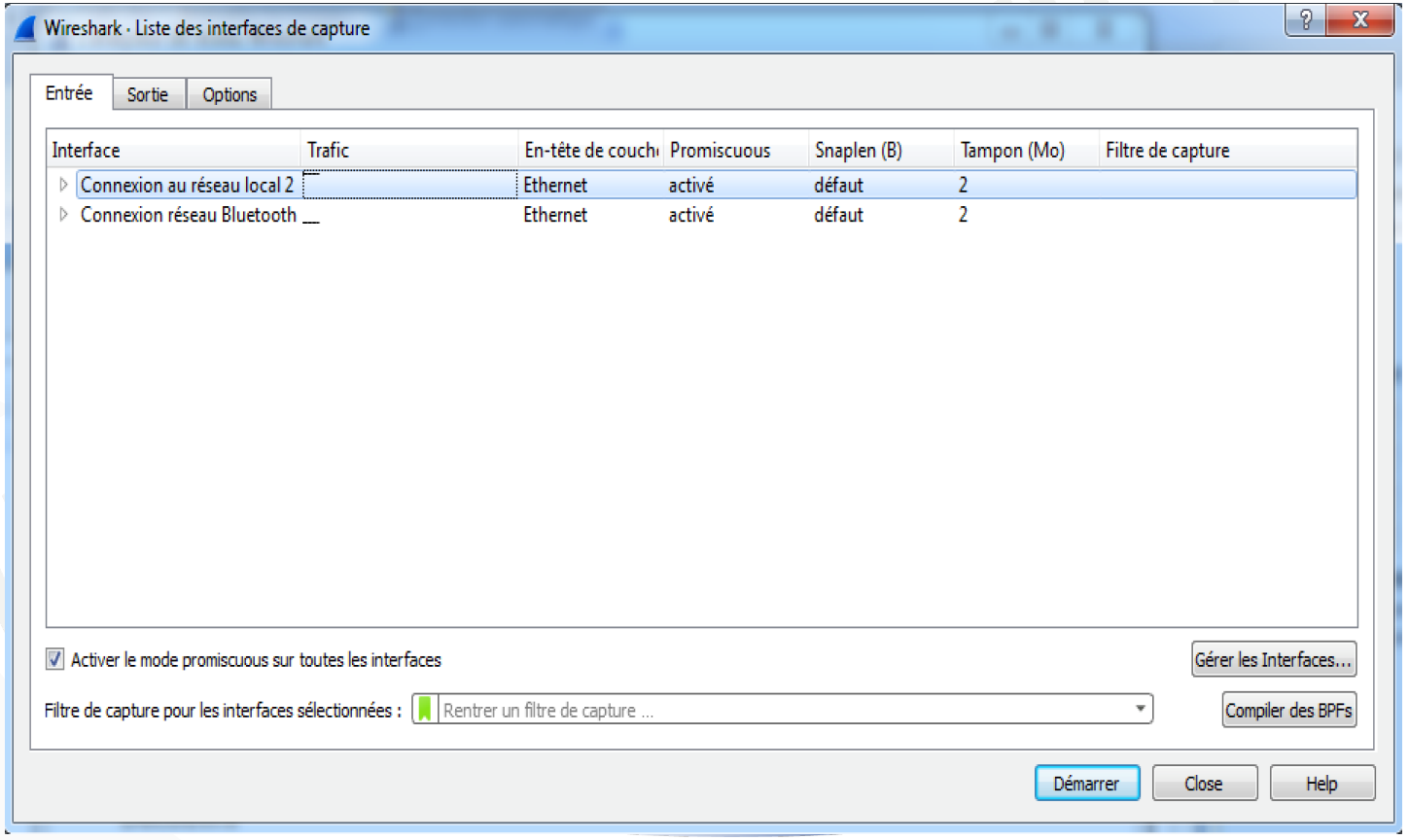
- Interface:



Outils réseaux

wireshark – notre première capture

- Il faut premièrement sélectionner l'interface réseau sur laquelle l'analyseur "écoute"
- Pour cela, allez sur Capture -> Options

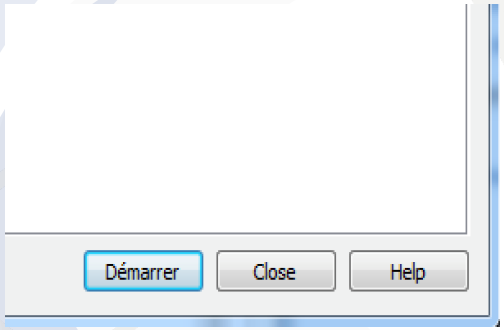


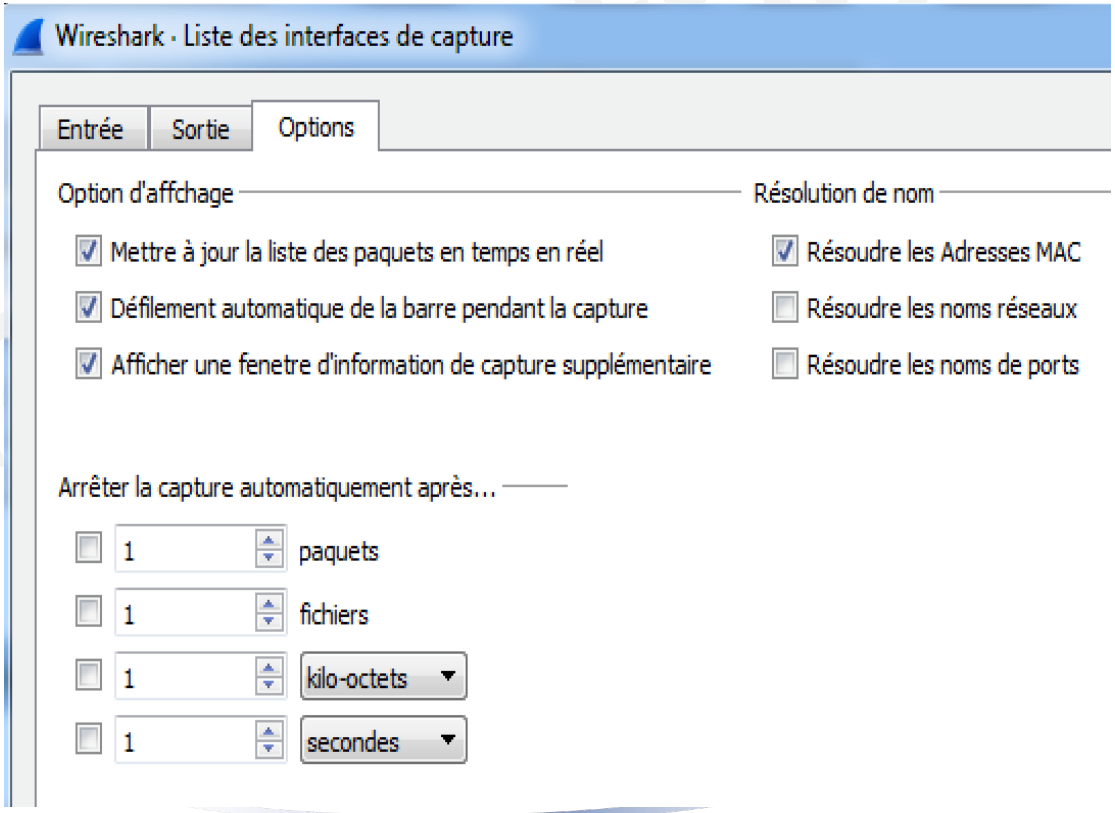
Outils réseaux

wireshark – notre première capture

- Dans le troisième onglet, remarquez quelques options intéressantes comme la résolution DNS et les options d'arrêts de capture

- On peut maintenant cliquer sur Démarrer

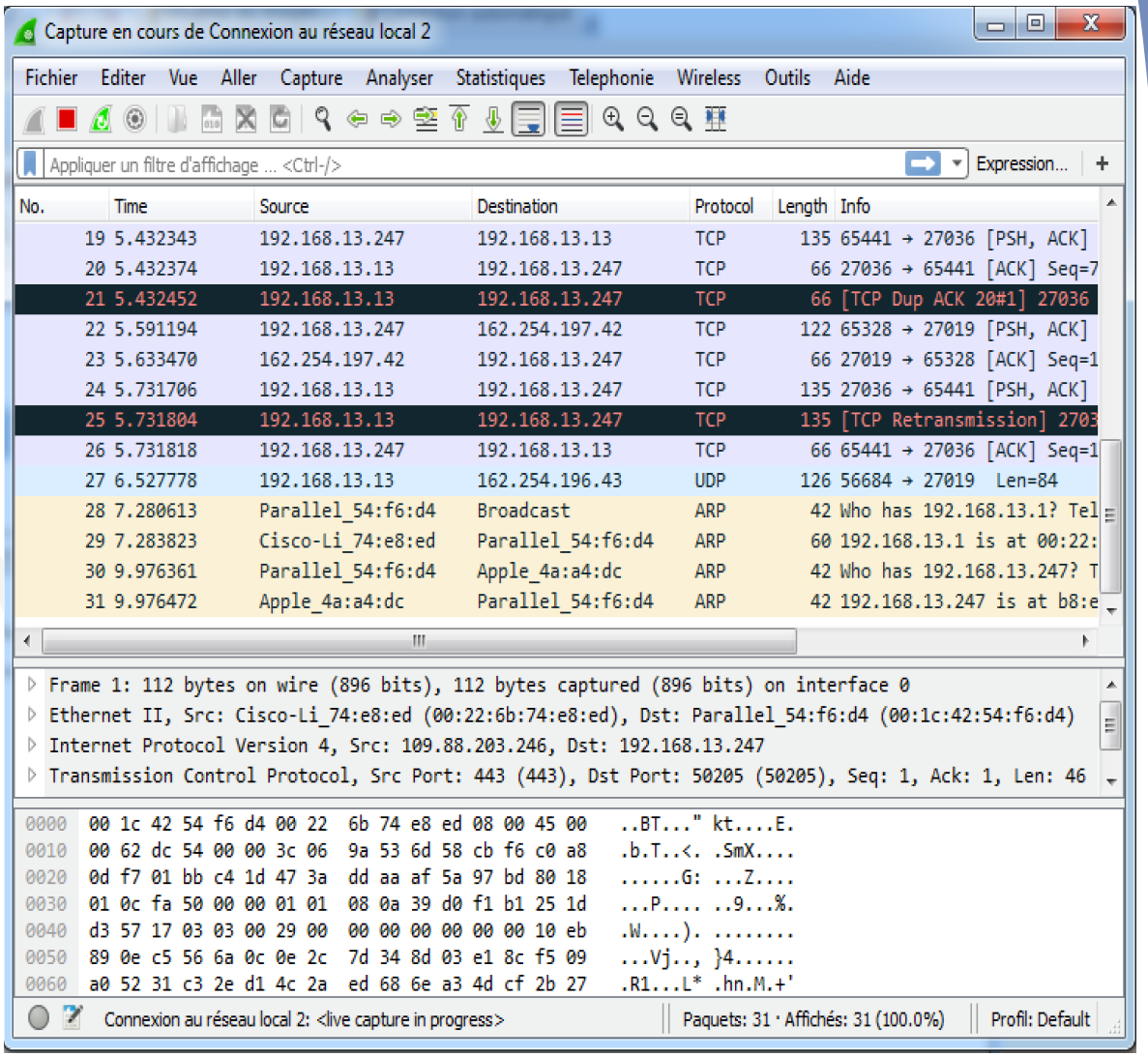




Outils réseaux

wireshark – notre première capture

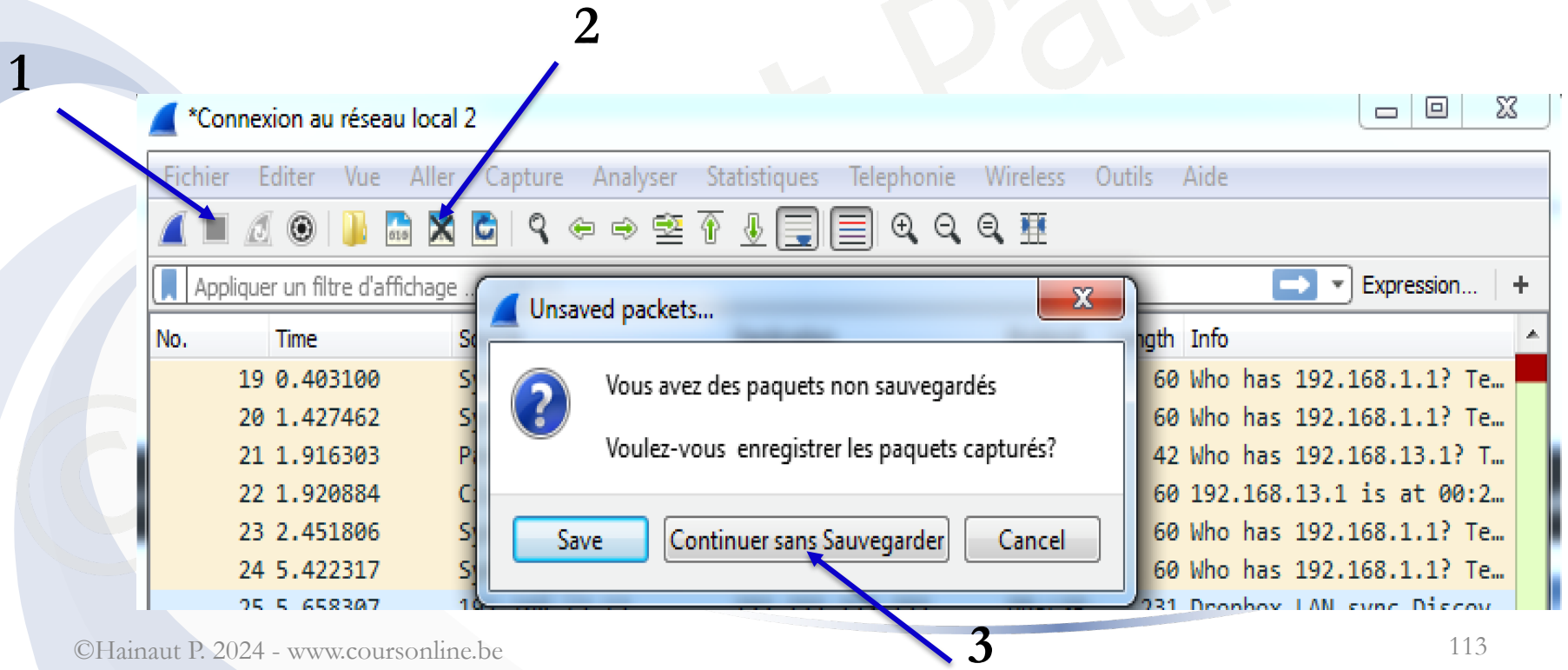
- Sans filtrage, le nombre d'informations capturées est grand et il n'est pas facile de trouver ce qu'on cherche …



Outils réseaux

wireshark – notre première capture

- Pour arrêter la capture, appuyez sur le bouton Stop (en rouge)
- Pour effacer les données capturées, appuyez sur le bouton Fermer …



Outils réseaux

wireshark – les filtres

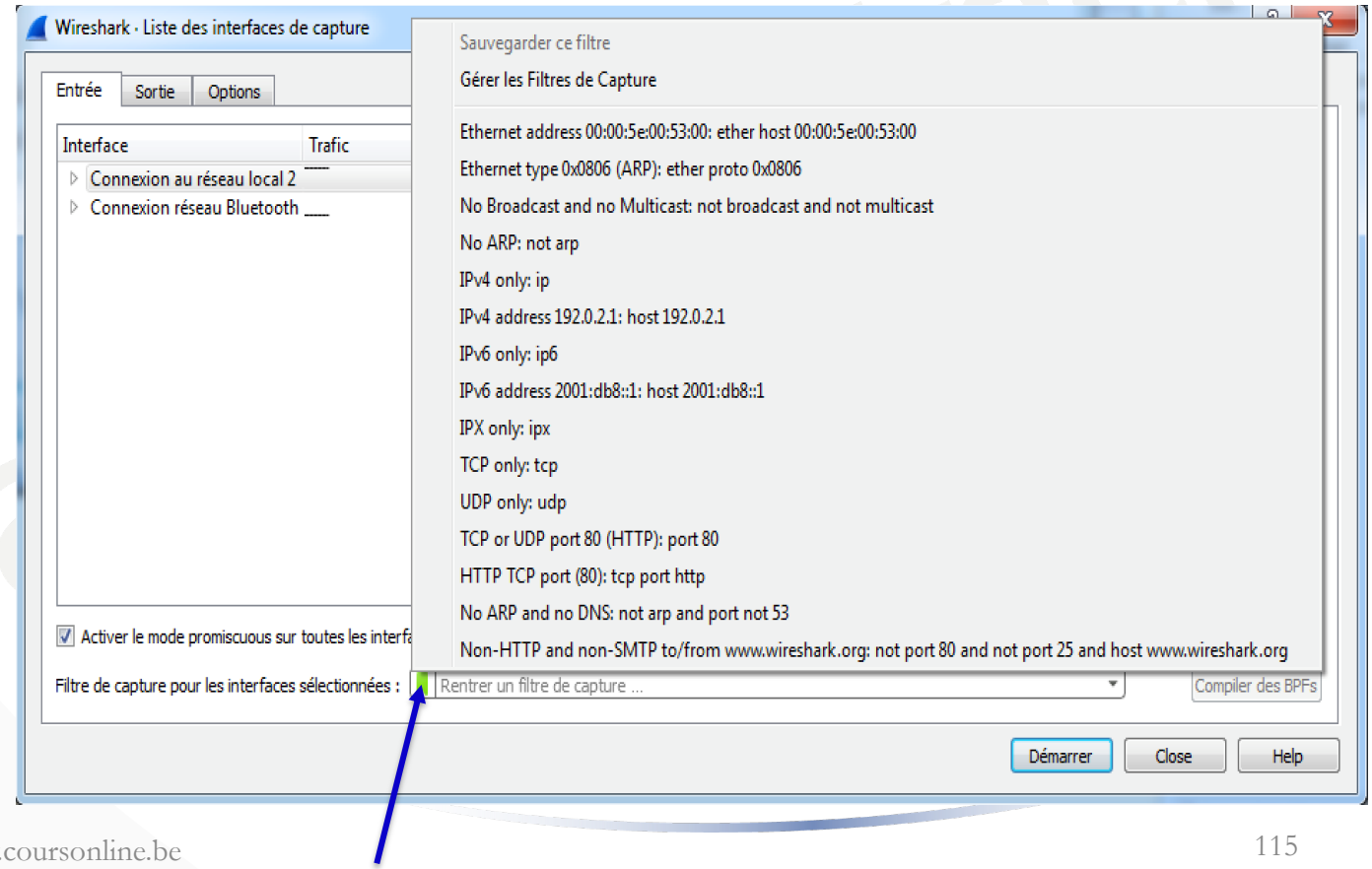
- Deux types de filtres existent:
  - Le filtre de capture: définit les types de données qui seront enregistrés lors de la capture et limite ainsi la taille du journal
  - Le filtre d'affichage: permet de sélectionner des données dans le journal, pendant ou une fois la capture effectuée

Outils réseaux

wireshark – le filtre de capture

- Dans Capture -> Options -> Entrée, appuyez sur le drapeau vert au niveau du filtre, une liste déroulante s'ouvre dans laquelle vous pouvez sélectionner votre filtre

- Cliquez ensuite sur Démarrer



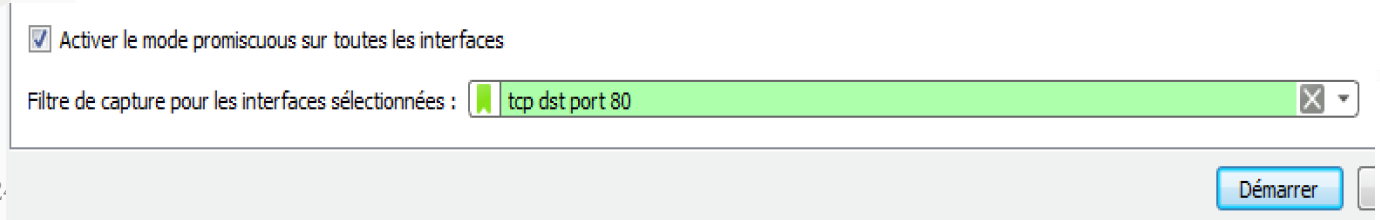
Outils réseaux

wireshark – le filtre de capture

- Une manière plus précise de filtrer est d'écrire son filtre dans le champ prévu à cet effet

- La requête comprend 4 éléments:
  - protocole: ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp
  - direction: src, dst, src and dst, src or dst
  - hôte(s): net, port, host, portrange
  - opérations logiques: not, and, or

- Exemple:



Outils réseaux

wireshark – le filtre de capture

- Tous les éléments ne doivent pas forcément être précisés:
  - protocole: si aucun protocole n'est précisé, tous les protocoles sont utilisés
  - direction: par défaut, **src or dst** est utilisé
  - hôte(s): par défaut, **host** est utilisé
  - opérations logiques: par défaut, il n'y en a pas

- Exemples de syntaxe valide:
  - host 192.168.13.234
  - no icmp
  - src host 192.168.13.234 and dst net 10.0.10.0/24

Outils réseaux

wireshark – le filtre d'affichage

- Le filtre d'affichage permet de filtrer les données récoltées (via un filtre de capture ou pas)

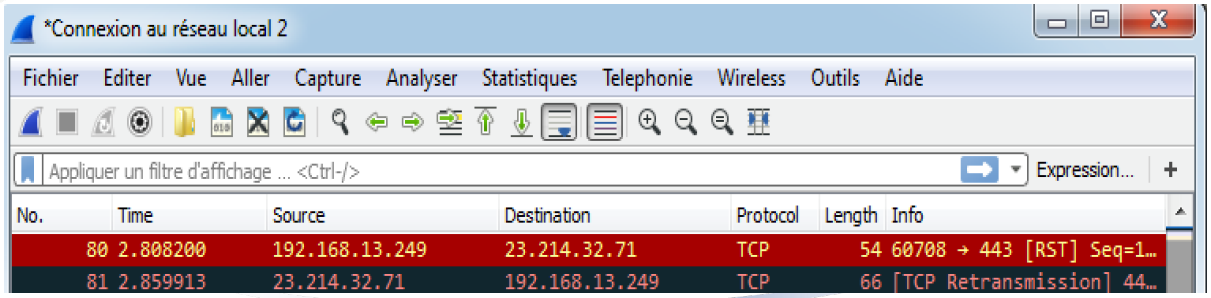
- Il comporte beaucoup plus d'items de recherche que le filtre de capture

- Il n'est pas nécessaire de refaire une capture si on change le filtre d'affichage

Outils réseaux

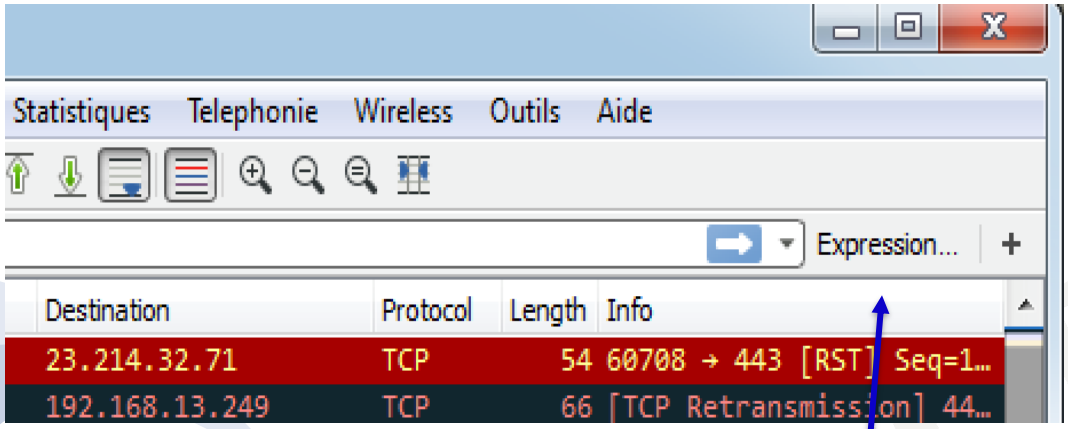
wireshark – le filtre d'affichage

- La requête comprend 4 éléments:
  - protocole: beaucoup de protocoles du modèle OSI
    - Champ1, champ2: paramètres optionnels liés au protocole
  - opérateurs de comparaisons: equal (==), not equal (!=), plus grand (>), plus petit (<), plus grand ou égal (>=), plus petit ou égal (<=)
  - valeur: une adresse IP
  - opérateurs logiques: not, and, or, xor
  - Autres paramètres: optionnel

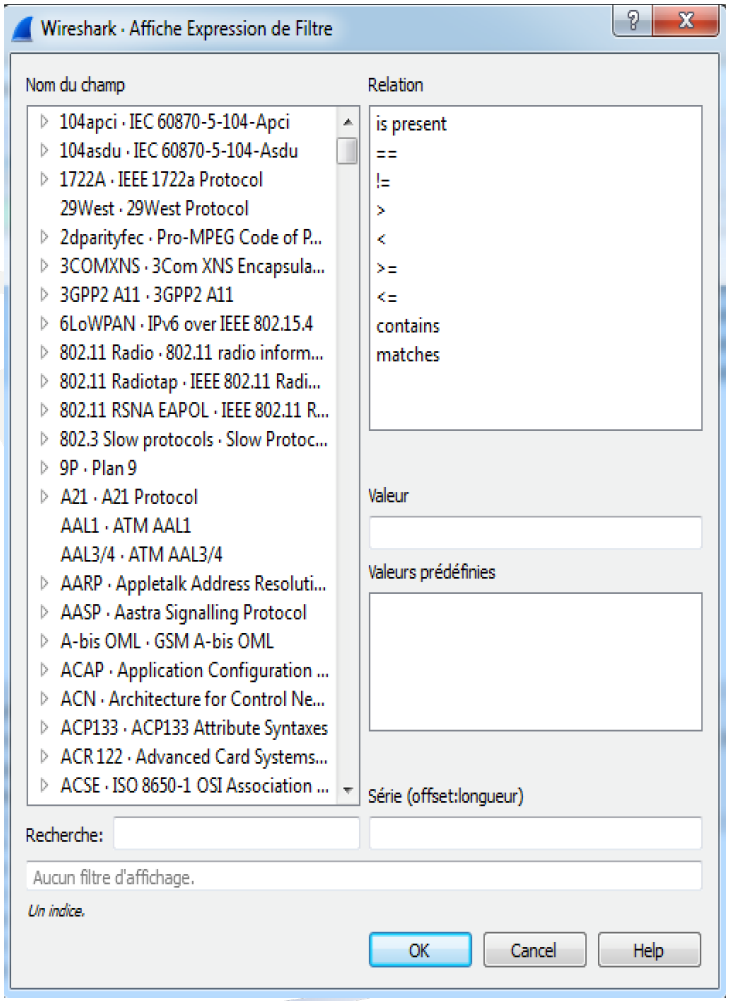


Outils réseaux

wireshark – le filtre d'affichage



- Pour faciliter l'écriture du filtre, utilisez le bouton expression



Outils réseaux

wireshark – le filtre d'affichage

- Quelques exemples de requêtes:
  - ip.addr == 192.168.13.249 (notez que ip == 192.168.13.249 ne fct pas)
  - ip.dst != 192.168.13.1
  - udp.port == 53
  - dns or icmp

Outils réseaux

wireshark – analyse d'une trame

- Grâce au filtres, on peut sélectionner le type de trafic que l'on veut examiner

- Et, une fois les données capturées et filtrées, on peut les analyser

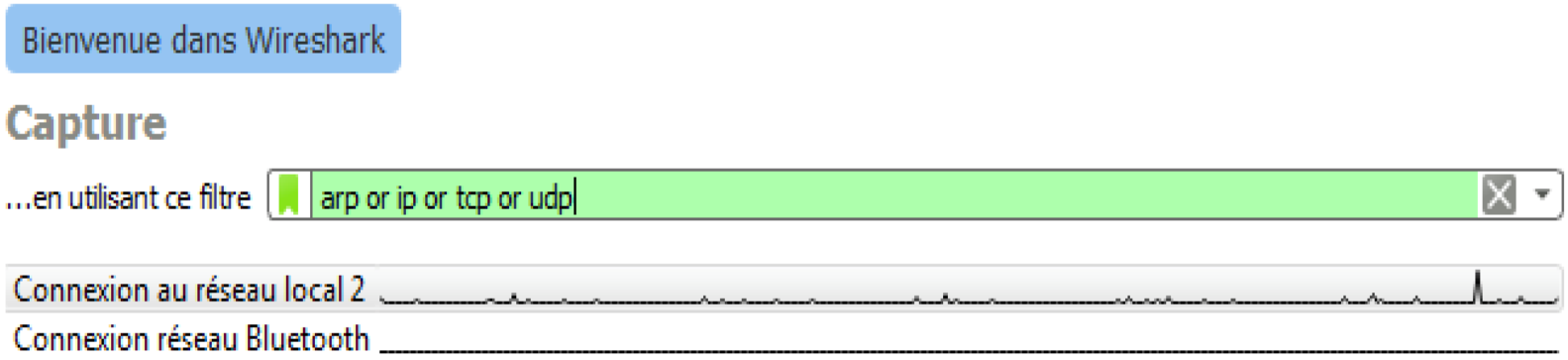
- On clique sur une trame et on peut voir ce qu'elle contient en détail

Outils réseaux

wireshark – analyse d'une trame

- Exemple: je veux examiner le téléchargement d'un fichier pdf du site coursonline.be sur mon ordinateur

- Pour le filtre de capture, je vais sélectionner les protocoles IP, UDP et TCP

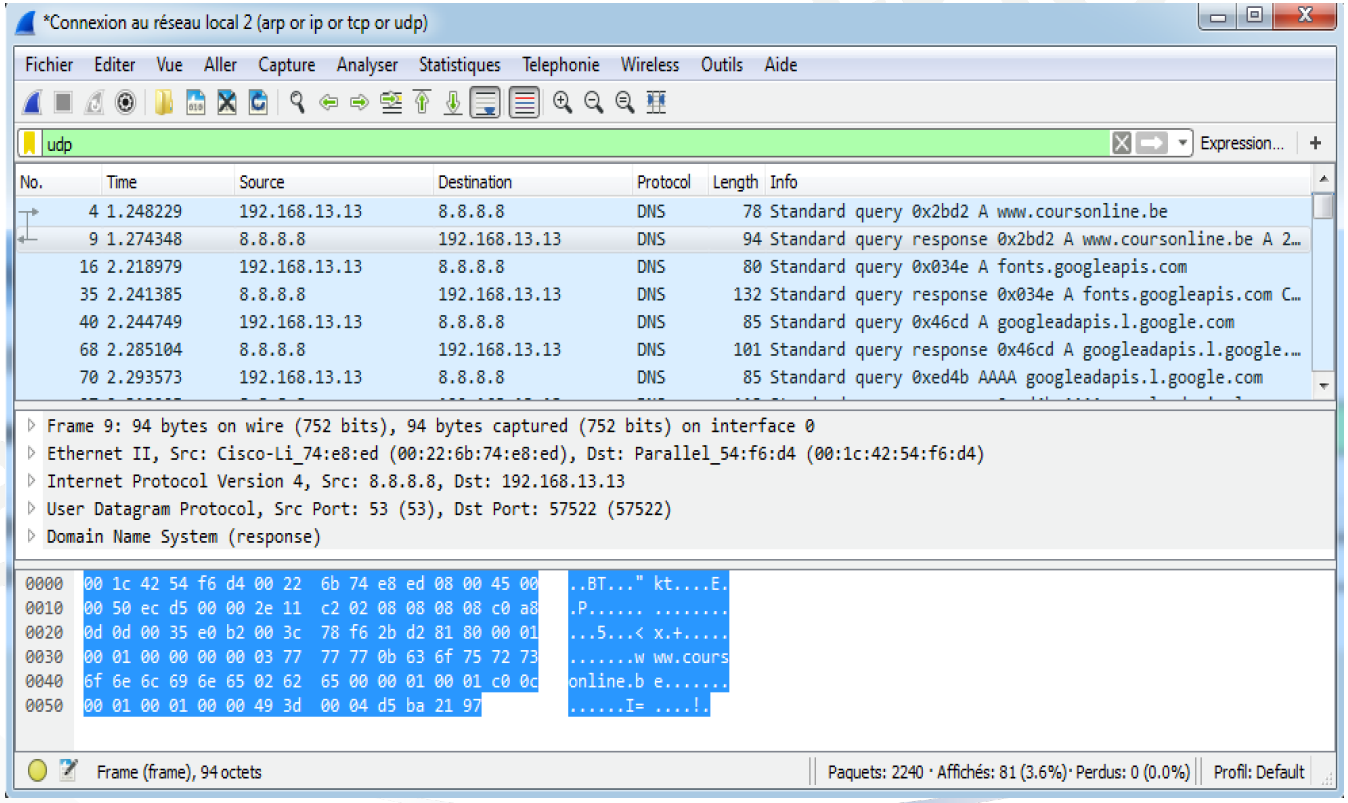


Outils réseaux

wireshark – analyse d'une trame

- Voyons voir le trafic UDP qui nous permet de voir la requête vers le serveur DNS et sa réponse

- On sélectionne la trame 9 et on peut cliquer sur chaque ▷ pour avoir le détail

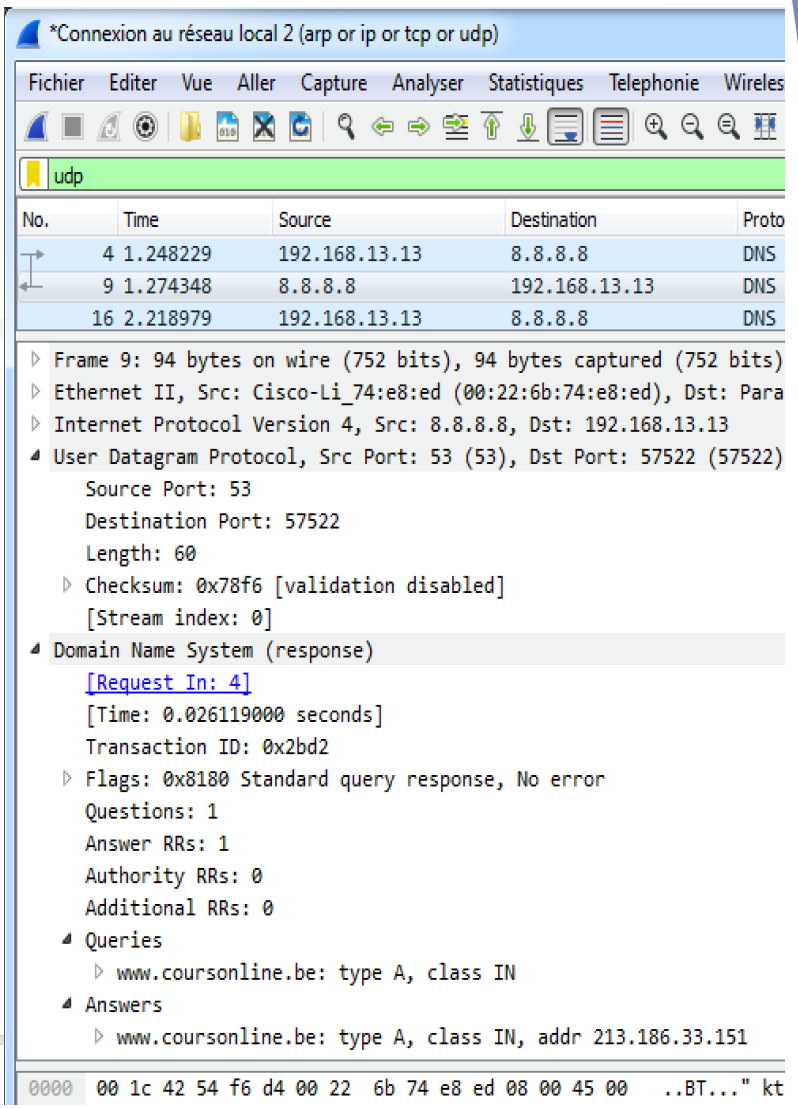


Outils réseaux

wireshark – analyse d'une trame

- La trame de type Ethernet II contient un paquet IPv4, qui contient un message UDP qui contient une donnée DNS

- On peut voir (avec la trame 4) que 192.168.13.13(l'adresse du PC) est à l'origine de la requête et que 8.8.8.8 est le serveur DNS (de Google dans ce cas) qui a traité cette requête

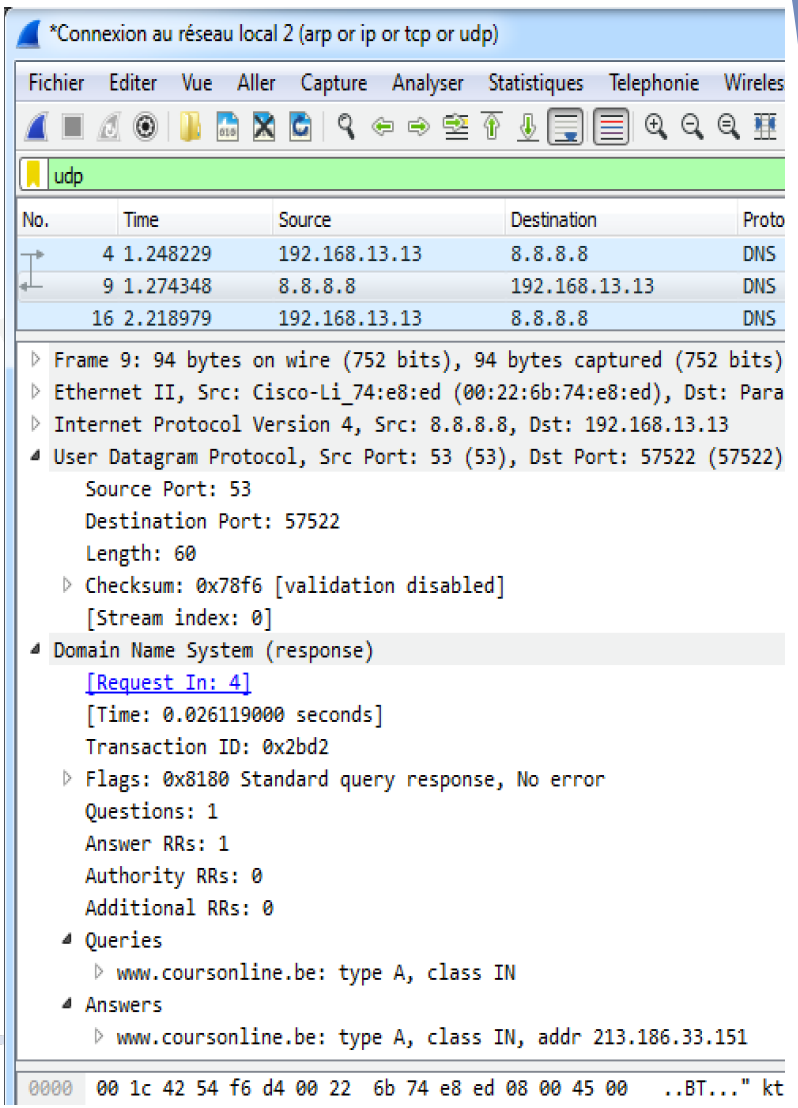


Outils réseaux

wireshark – analyse d'une trame

- On voit que c'est une réponse DNS, que l'objet de la requête était le nom de domaine www.coursonline.be, et que la réponse (l'adresse IP correspondante) est 213.186.33.151

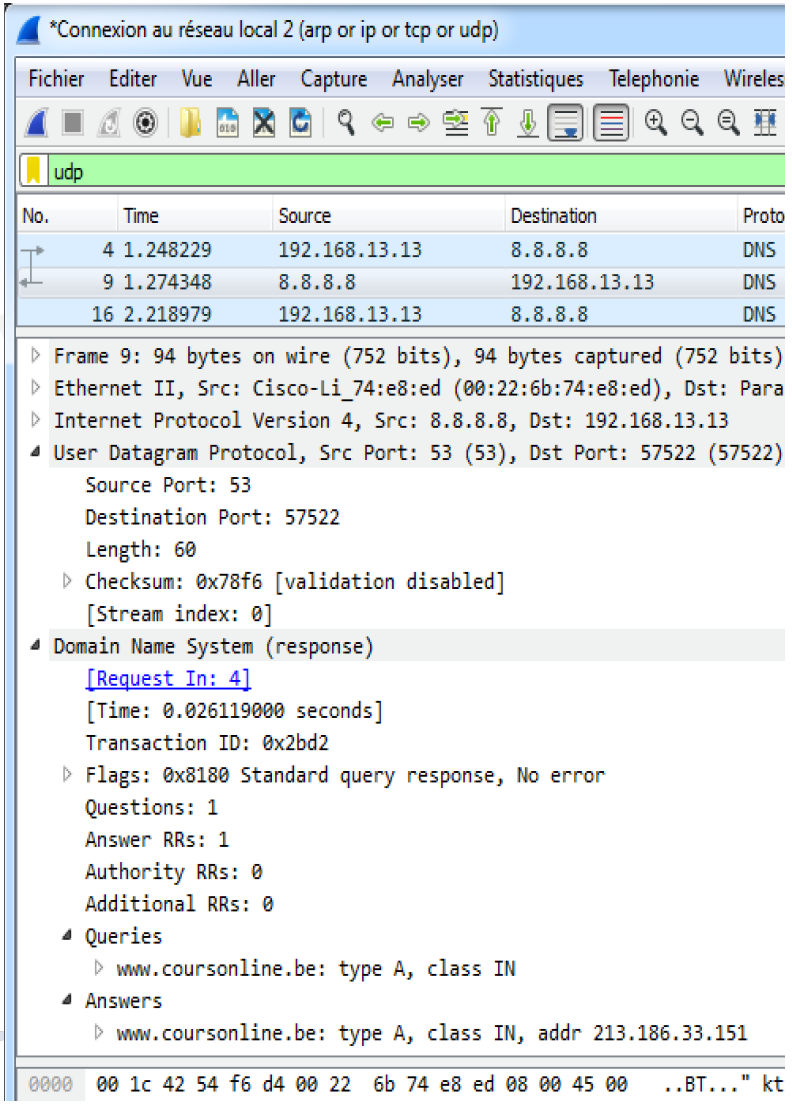
- L'adresse IP source est donc celle du serveur et l'adresse IP de destination, celle du PC



Outils réseaux

wireshark – analyse d'une trame

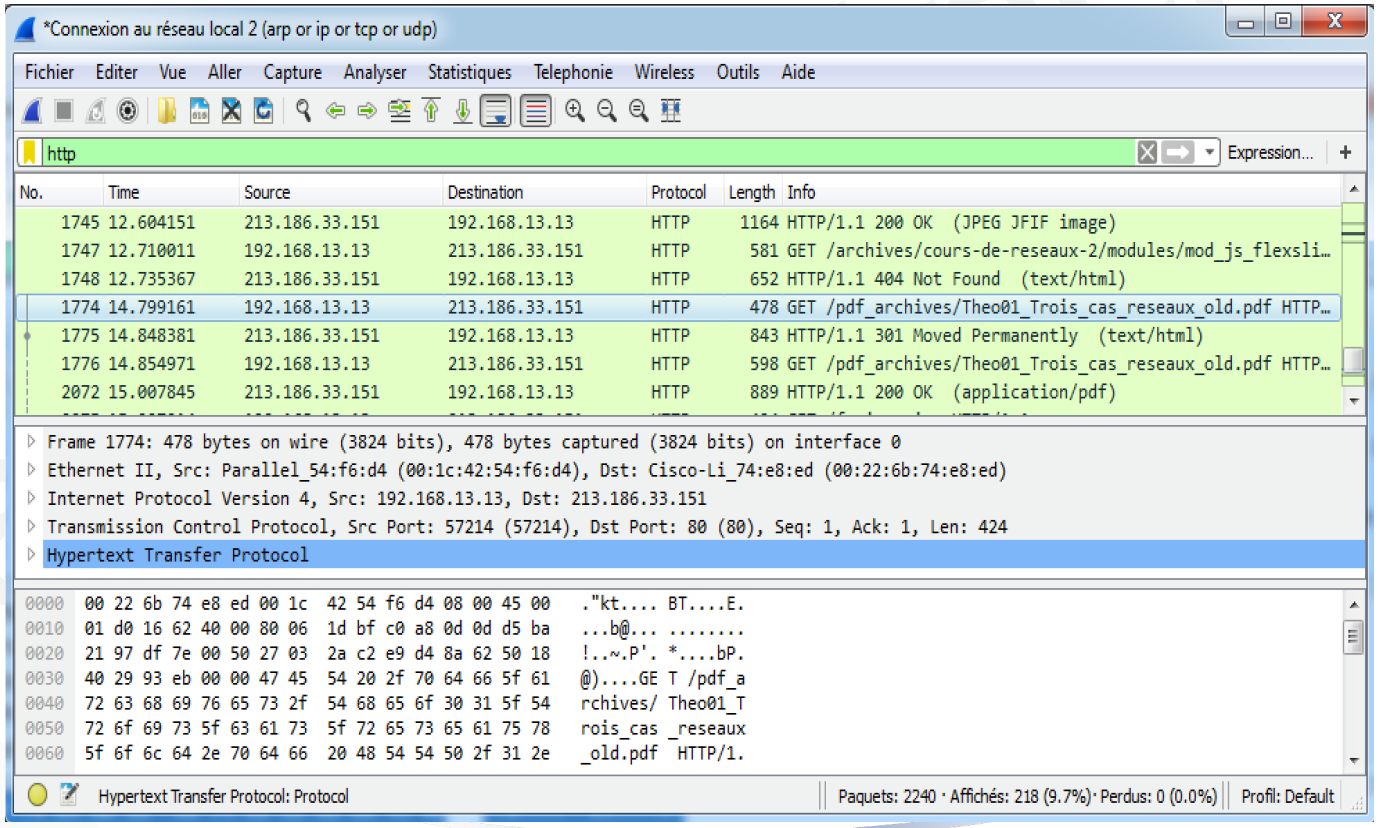
- Le port source est celui du serveur (53 n° de port standard DNS), celui de destination est un port > 1024 choisi aléatoirement lors de l'établissement de la requête DNS en trame 4



Outils réseaux

wireshark – analyse d'une trame

- Dans le trafic HTTP, nous repérons la trame qui demande le téléchargement du pdf Theo01

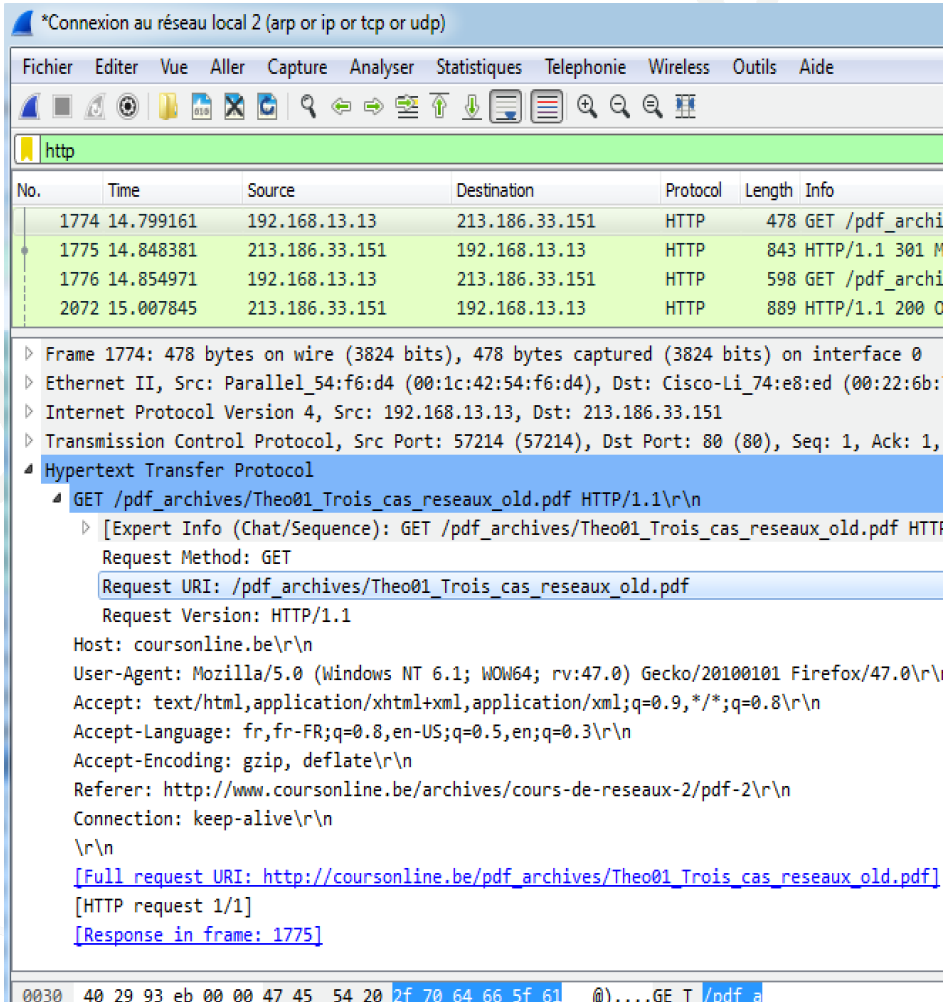


Outils réseaux

wireshark – analyse d'une trame

- La trame contient un paquet IPv4, qui contient un message TCP qui contient une donnée HTTP

- L'adresse source est celle du PC (192.168.13.13), l'adresse de destination est celle du serveur Web (213.186.33.151)
- Le port source est un port aléatoire > 1024 (57214) et le port destination est le port HTTP du serveur (80)



Conclusion

ouf …

- Voilà un gros morceau, si pas le gros morceau de notre apprentissage …
- Revenez souvent sur ces documents Theo05a, b et c au fur à mesure de votre apprentissage pour en comprendre toutes les subtilités

- Toutes les notions abordées sont importantes

- Ne négligez pas la présentation sur le calcul IP et les vidéos réalisées pour vous aider

- La suite dans la présentation Theo05d sur IPv6 …

- Merci de votre attention