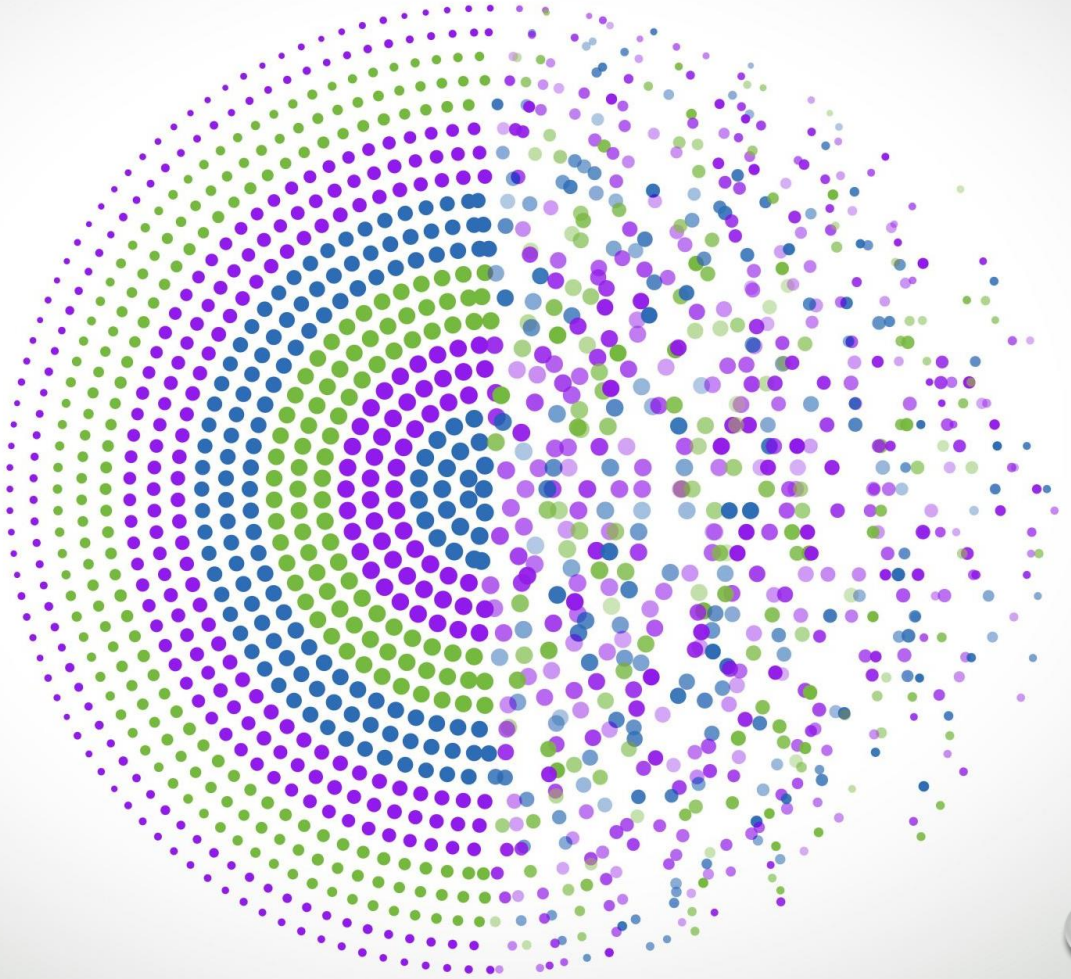
R2.04 – LES BASES DES RÉSEAUX

RESPONSABLE : CRISTINA ONETE

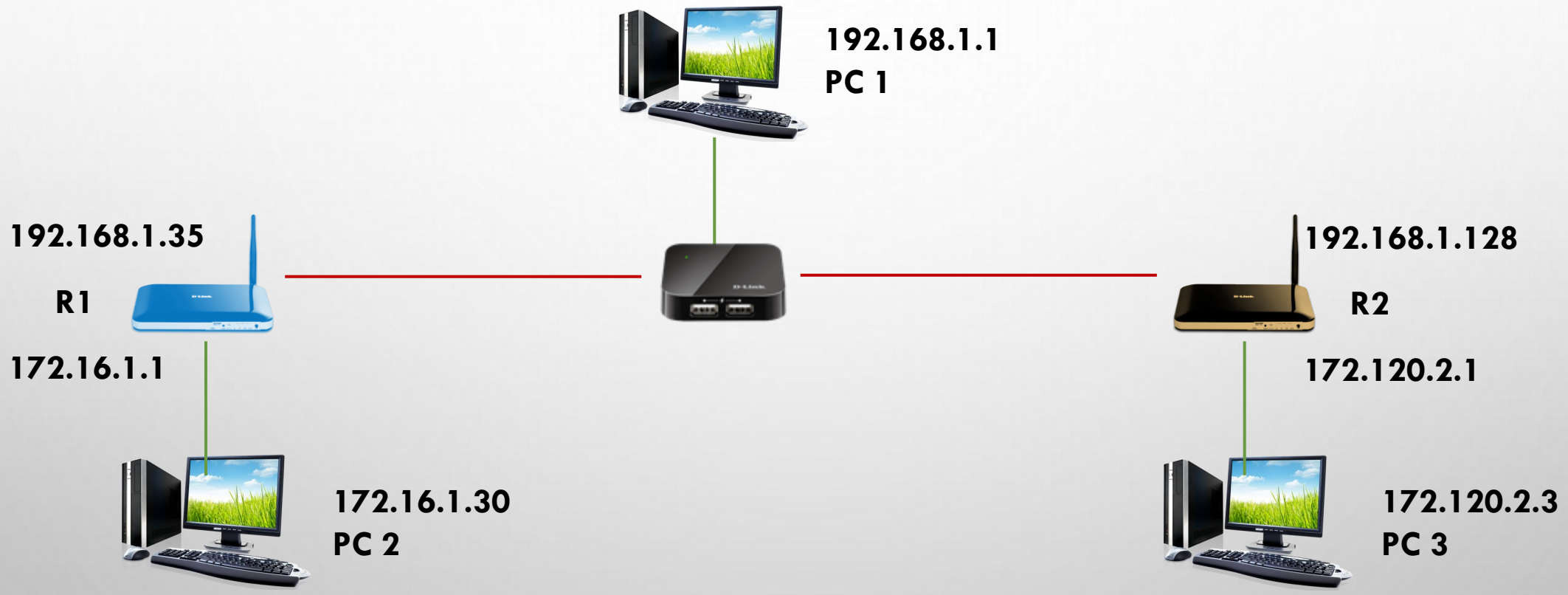
MATERIEL : HTTPS://WWW.ONETE.NET/TEACHING.HTML

EMAIL : MARIA-CRISTINA.ONETE@UNILIM.FR



COUCHE RÉSEAU : LE ROUTAGE

PROBLÉMATIQUE -- LE ROUTAGE



Comment PC 1 peut-il envoyer un message à PC 2 ou PC 3 ?

LES TABLEAUX DE ROUTAGE

- **OBJECTIF** : Établir des chemins entre n'importe quels deux utilisateurs
- Configurés dans des nœuds de lien (routeurs)
- **ROUTAGE STATIQUE** : tableau de routage programmé à chaque nœud,
  - Écrit à la main par l'administrateur de réseau à chaque modification
  - Peut être optimisé, fonctionne bien seulement dans des petits réseaux
- **ROUTAGE DYNAMIQUE** : tableaux de routage qui s'adaptent dynamiquement
  - Les routeurs se parlent l'un à l'autre pour adapter leurs routes

LE ROUTAGE : UNE AUTRE VUE

- CIRCULATION : A CHAQUE POINT, UNE DIRECTION VERS LE PROCHAIN POINT
  - Au rondpoint 1 : prenez à droite
  - Plus tard : prenez à gauche
  - Au rondpoint 2 : tout droit

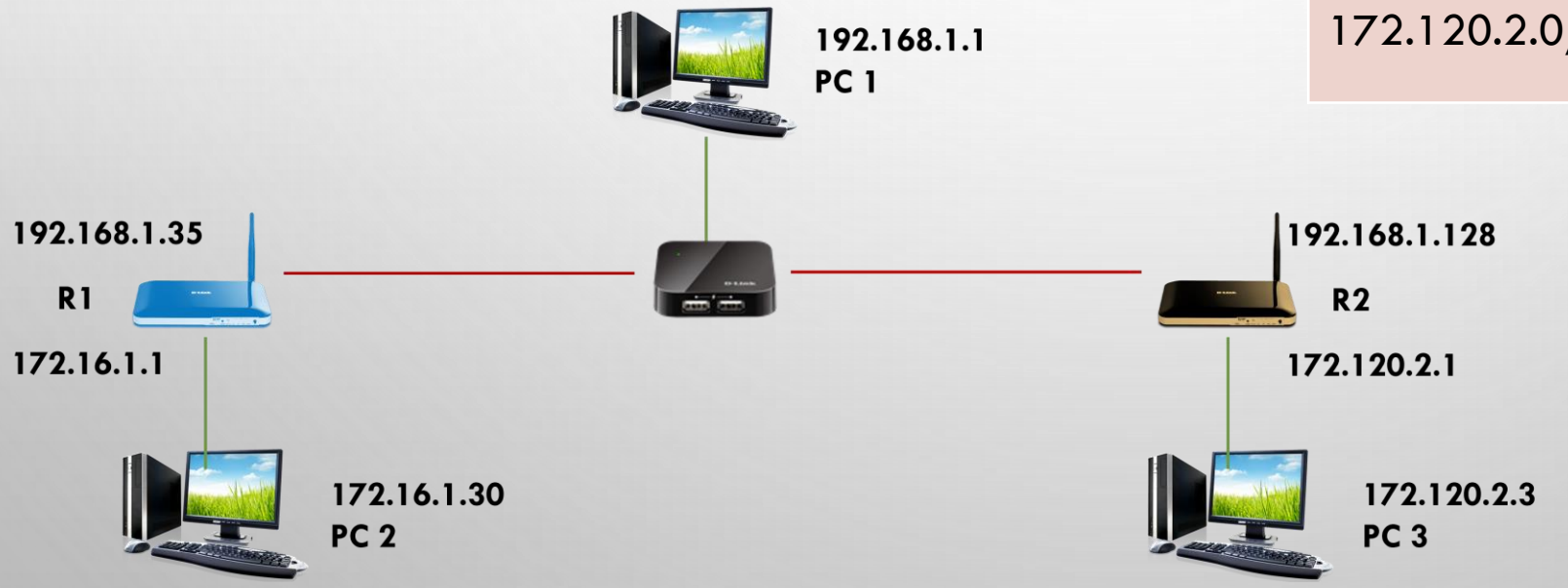
- ROUTAGE : CHAQUE ROUTEUR, PROCHAINE ÉTAPE
  - R1 : prochain routeur : R2
  - R2 : pour aller vers 192.168.1.0/24, R3

    pour tout autre chemin, R0

- ON N'INDIQUE JAMAIS TOUT LE CHEMIN

ROUTAGE STATIQUE : SOLUTION 1

Destination	Router
192.168.1.0/24	none
172.16.1.0/24	192.168.1.35
172.120.2.0/24	192.168.1.128



PROGRAMMER LE TABLE DE ROUTAGE

```
ip route <add ou del> <destination>/CIDR via <IP router>
```

@dest. finale des packets

Routage toujours necessaire hors réseau, jamais dans le réseau

- EXEMPLE : `ip route add 172.168.1.0/24 via 192.168.1.35`

- POUR VOIR LES RÉSULTATS :

```
ip route list
```

Destination	Router
192.168.1.0/24	none
172.16.1.0/24	192.168.1.35
172.120.2.0/24	192.168.1.128

UNE MEILLEURE SOLUTION

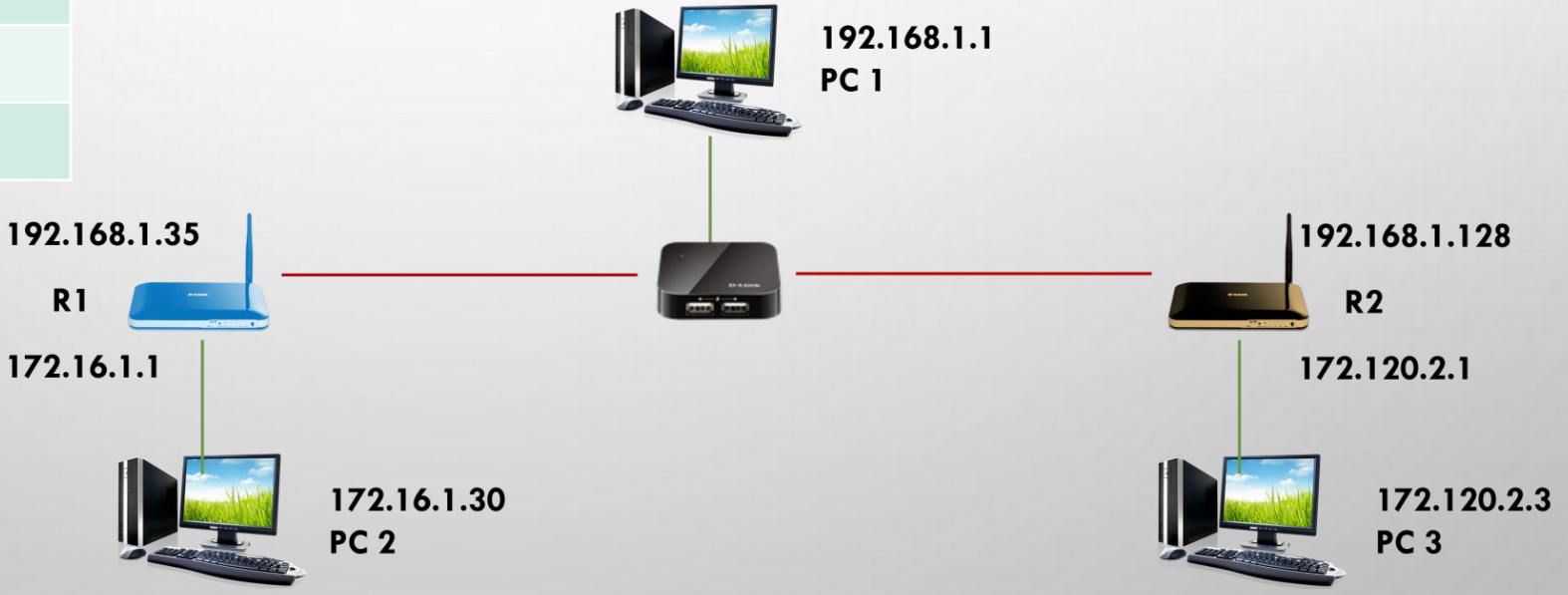
Solution précédente : OK mais non-optimale

Une meilleure solution serait de :

Choisir routeur par défaut PCA	**Ce router décide le routage plus loin**	**PLUS FACILE** : routage plus facile à modifier	**PLUS FLEXIBLE** : routage modifiable selon besoins et routeurs

Destination	Router
192.168.1.0/24	none
172.16.1.0/24	none
172.120.2.0/24	192.168.1.128

Destination	Router
192.168.1.0/24	none
0.0.0.0/0	192.168.1.35



IMPLEMMENTER CETTE SOLUTION

- POUR PC 1 : UN ROUTAGE PAR DÉFAUT

```
ip route add default via 192.168.1.35
```

Destination	Router
192.168.1.0/24	none
0.0.0.0/0	192.168.1.35

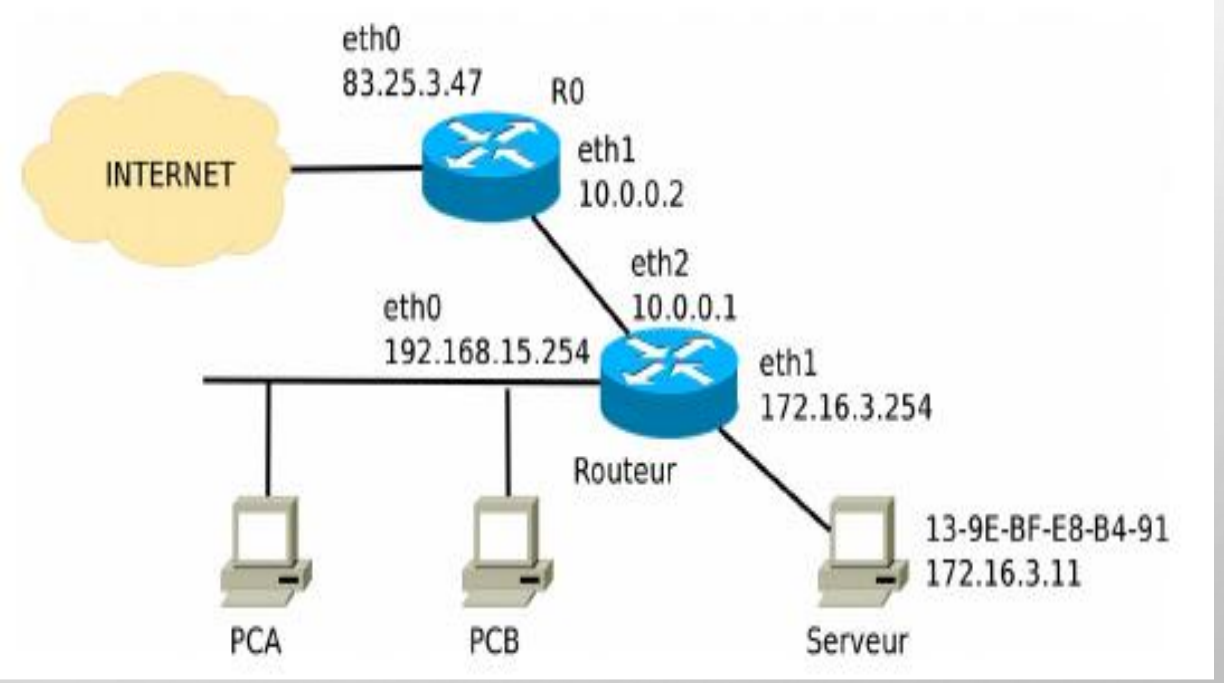
- POUR ROUTEUR R1 :

```
ip route add 172.120.2.0/24 via 192.168.1.128
```

Destination	Router
192.168.1.0/24	none
172.16.1.0/24	none
172.120.2.0/24	192.168.1.128

ROUTE PAR DÉFAUT VS ROUTE SPÉCIFIQUE

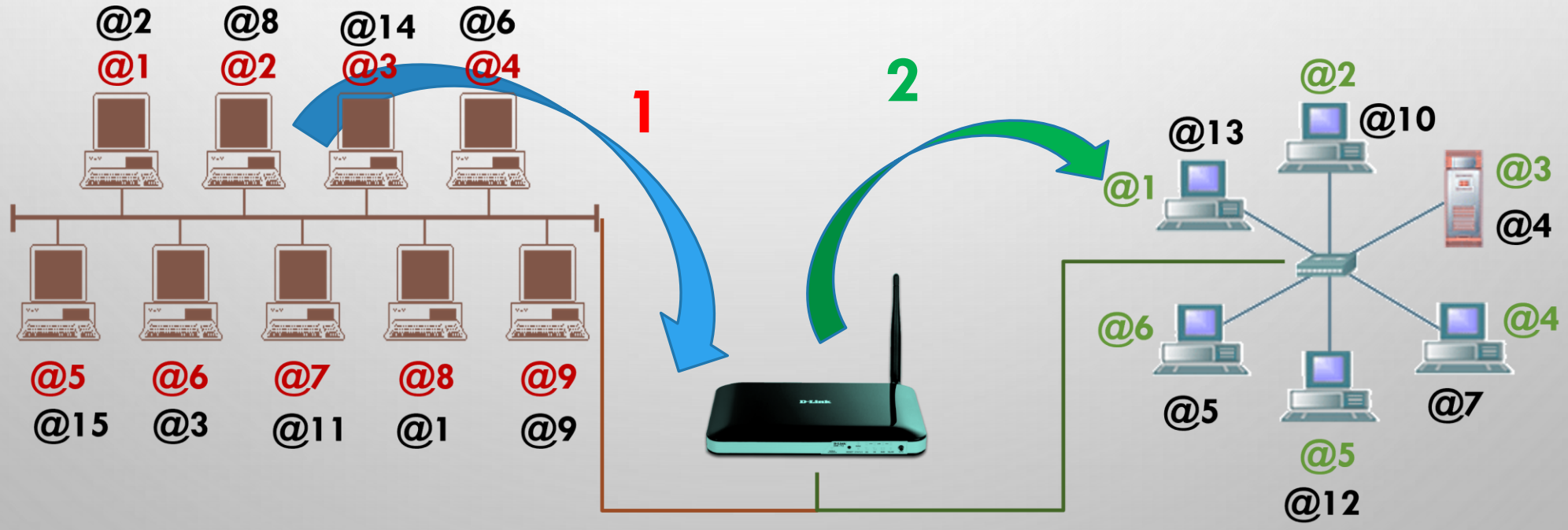
- ROUTE PAR DÉFAUT :
  - Typique machines utilisateurs (PCA, PCB, Serveur)
  - Peut indiquer une route "principale" pour un routeur (ex. la route vers l'Internet)
- ROUTE SPÉCIFIQUE :
  - En général utilisée pour les routeurs, pour indiquer une route secondaire



COUCHE RÉSEAU : ENVOYER UN PACKET

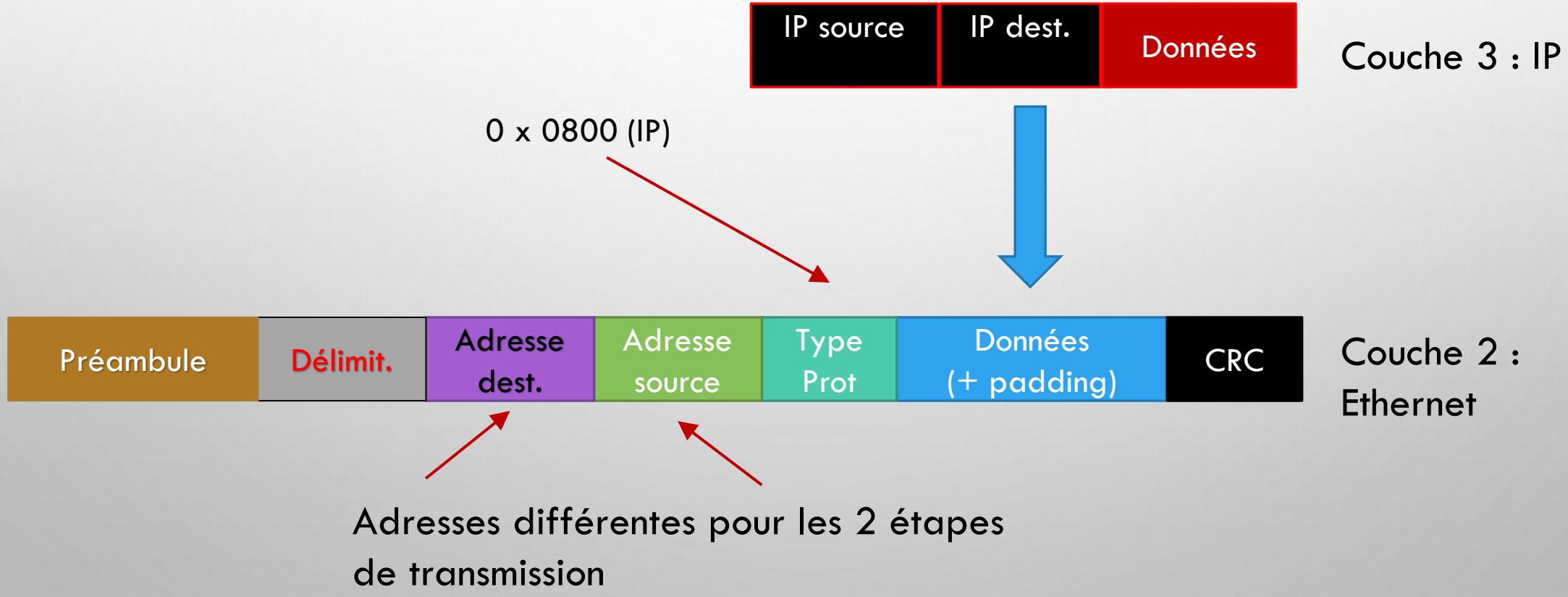
ENVOI INTER-RÉSEAU : 2 ÉTAPES

- PROBLÉMATIQUE : comment @2 (**@8**) peut-il envoyer un message à @6 (**@5**) ?



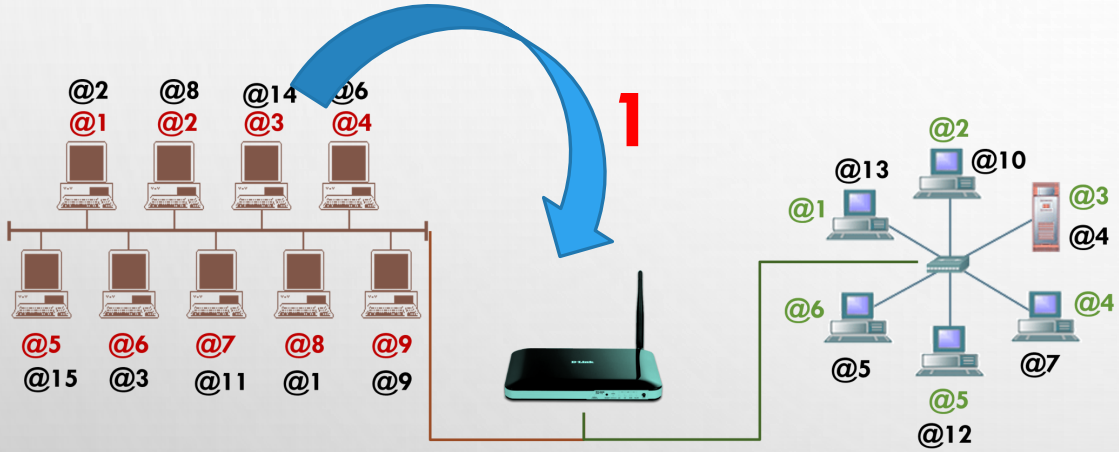
TRANSMISSION INTER-RÉSEAUX

- DEUX ÉTAPES DE TRANSMISSION INTRA-RÉSEAU
- ENCAPSULATION : >> IP >> ETHERNET 2



TRANSMISSION INTER-RÉSEAUX

- PREMIÈRE ÉTAPE DE TRANSMISSION : @IP = @8, @MAC = @2 → PASSERELLE



IP source	IP dest.	Données

Couche 3 : IP

Préambule	Délimit.	Adresse dest.	Adresse source	Type Prot	Données (+ padding)	CRC

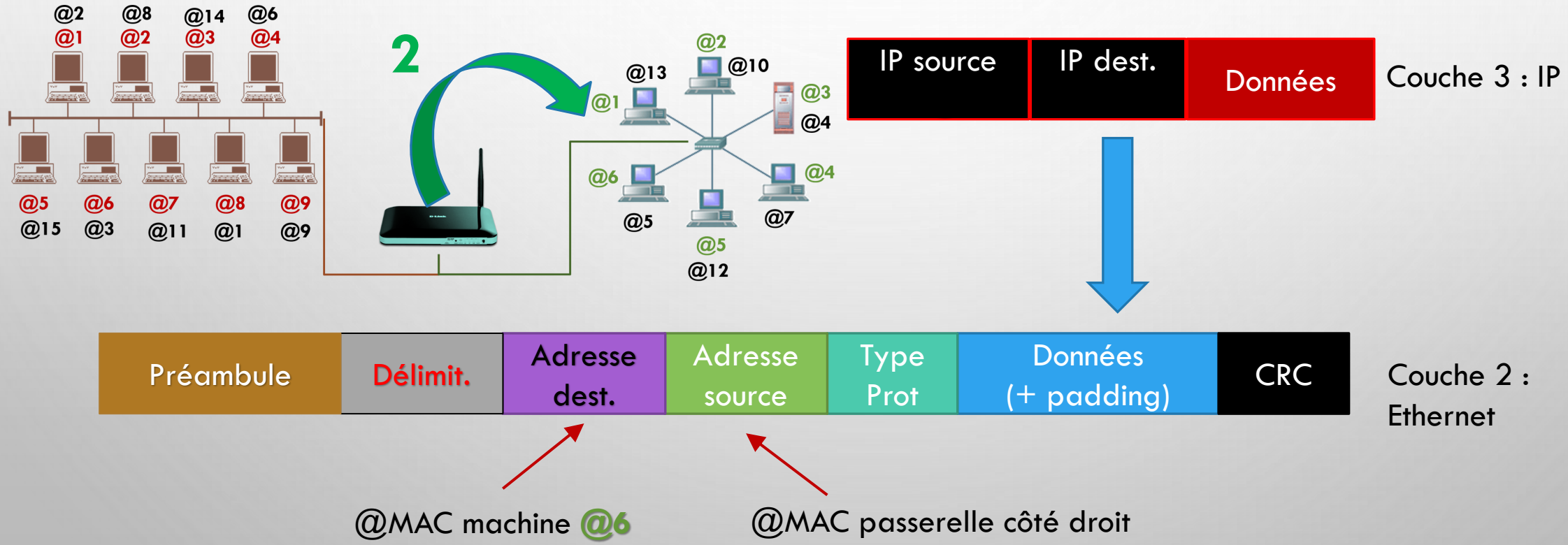
Couche 2 : Ethernet

@MAC : passerelle, côté gauche

@MAC machine @2

TRANSMISSION INTER-RÉSEAUX

- DEUXIÈME ÉTAPE : PASSERELLE → @IP = @5, @MAC = @6



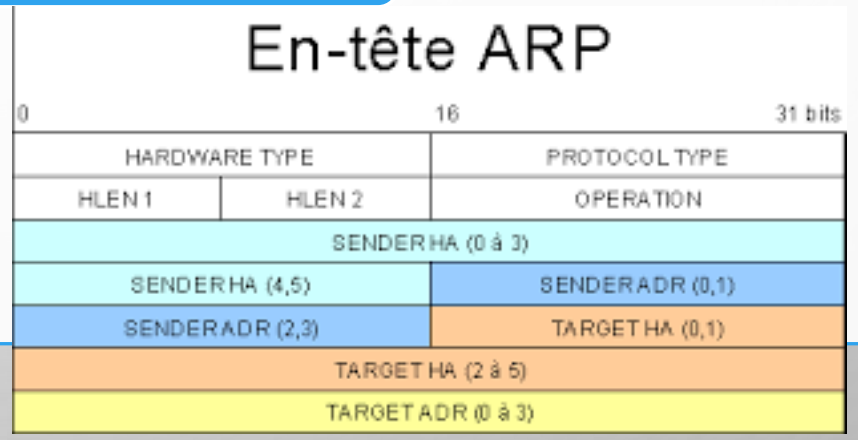
ASSOCIATION ADRESSES MAC/IP

Comment savoir quelle adresse MAC correspond à quelle adresse IP ?

- Protocole ARP, couche 2
- Ne peut s'exécuter que dans un seul réseau (DC)
- Exécuté par une machine cherchant une @ MAC à partir d'une @IP et une machine qui connaît l'@MAC cherchée

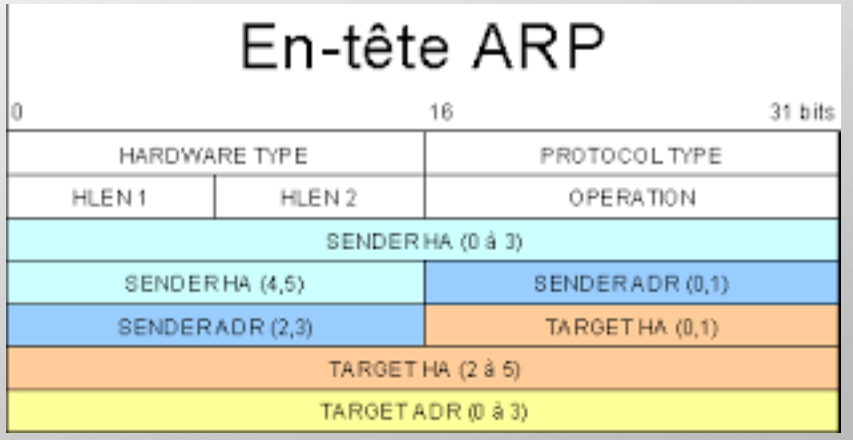
ARP : 2 types de messages : requête, réponse, encapsulées en Ethernet

- En-tête spécifique ARP
- Chaque message ARP spécifie :
  - Machine src : @MAC, @IP
  - Machine dst : @MAC
  - Machine "cible" : @MAC, @IP



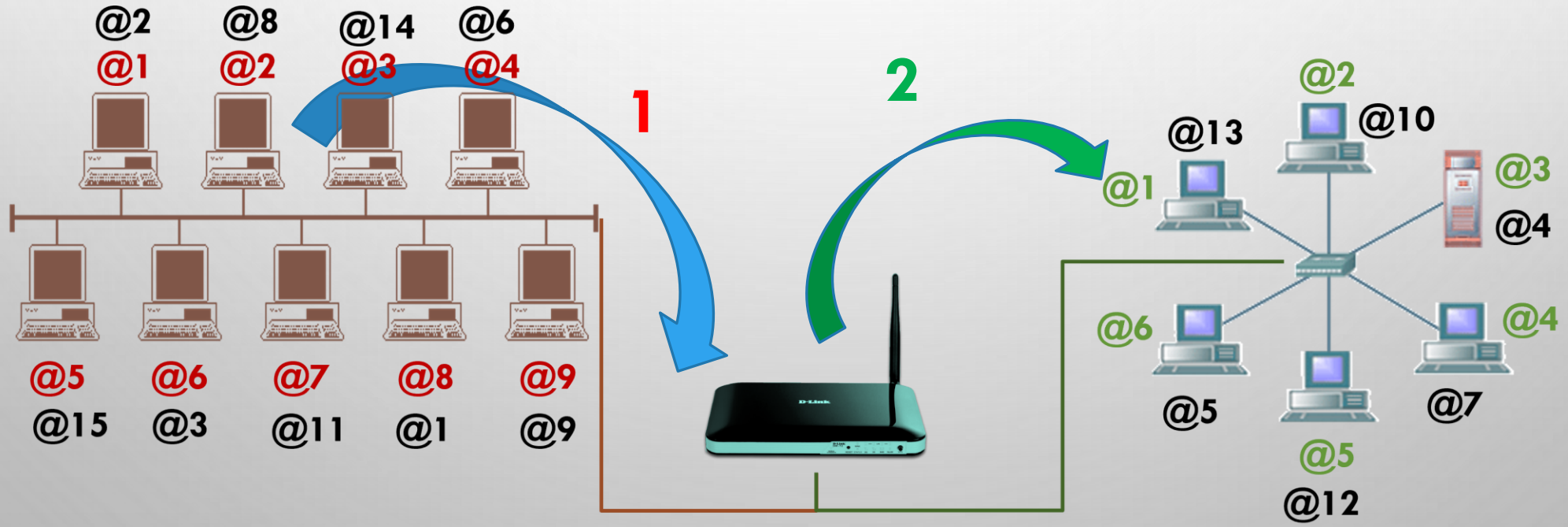
ASSOCIATION ADRESSES MAC/IP

- Situation : Machine A (@IP ip_A @MAC mac_A) cherche @MAC de la machine B (ip_B)
  - Machine C (ip_C, mac_C) connaît la réponse
- ARP requête :
  - Machine src : MAC mac_A, IP ip_A
  - Machine dst : MAC broadcast : FF:FF:FF:FF:FF:FF : broadcast intra-réseau !
  - Machine cible : @MAC inconnue 00:00:00:00:00:00
- ARP réponse :
  - Machine src : MAC mac_C , IP ip_C
  - Machine dst : MAC mac_A
  - Machine cible : @MAC mac_B , IP ip_B



ICMP : DÉBOGAGE À LA COUCHE 3

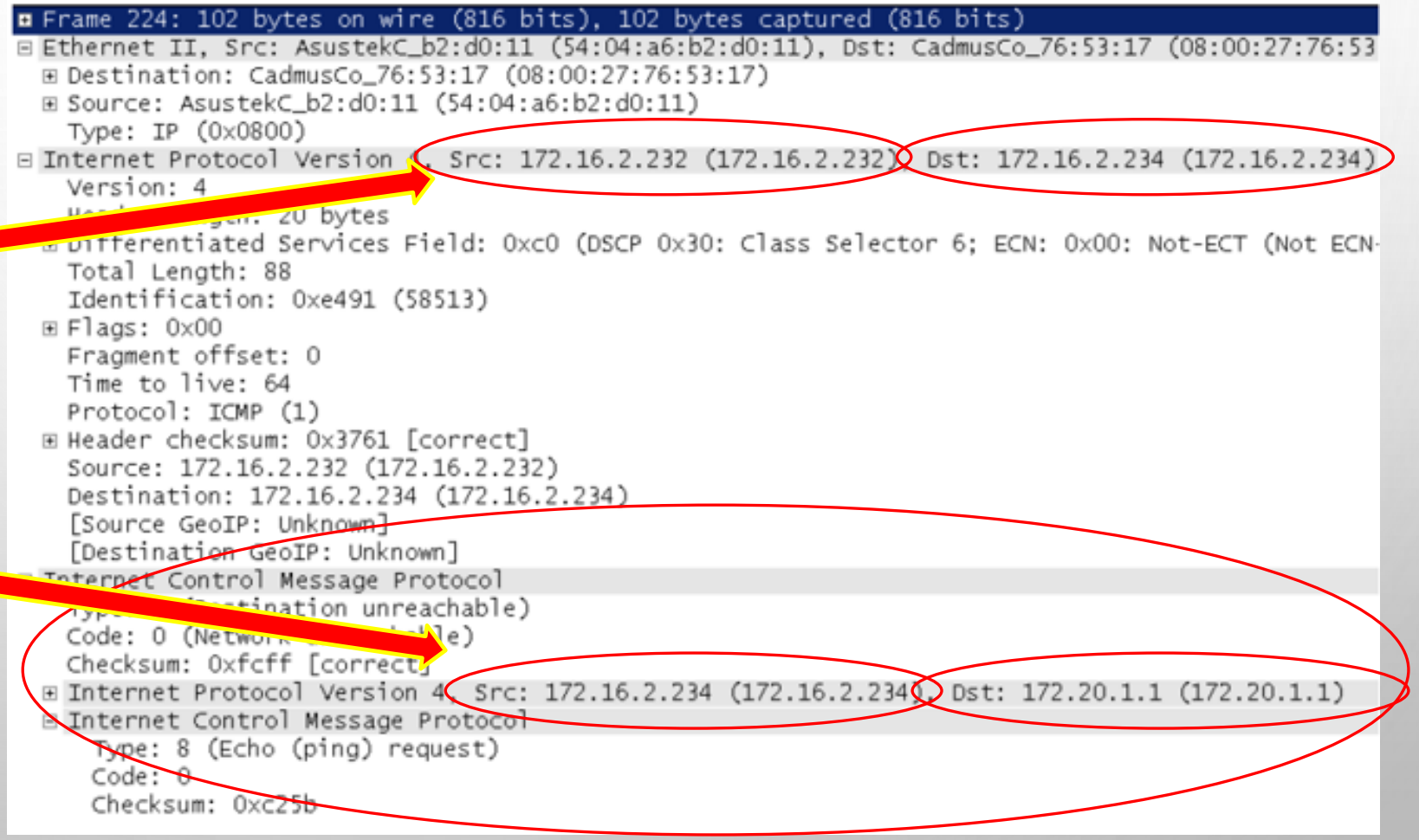
- SUPPOSONS UNE ERREUR DE TRANSMISSION À L'ÉTAPE 2
- UN MESSAGE D'ERREUR EST ENVOYÉ PAR LA PASSERELLE À L'EXPÉDITEUR (MESSAGE ICMP)



UN MESSAGE ICMP

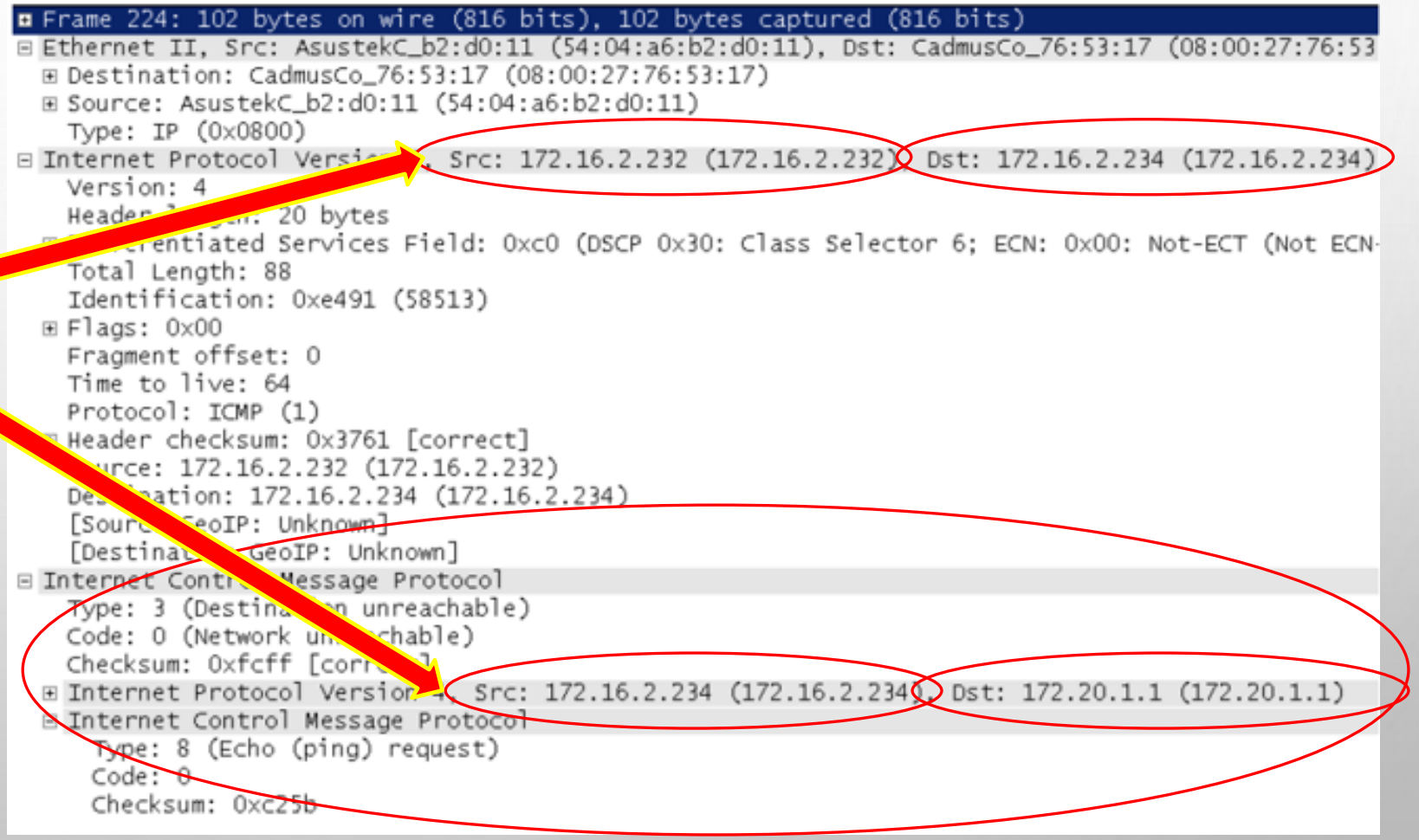
- Source/destination du message ICMP
- Source/destination du message encapsulé par ICMP

Pourquoi une différence ?



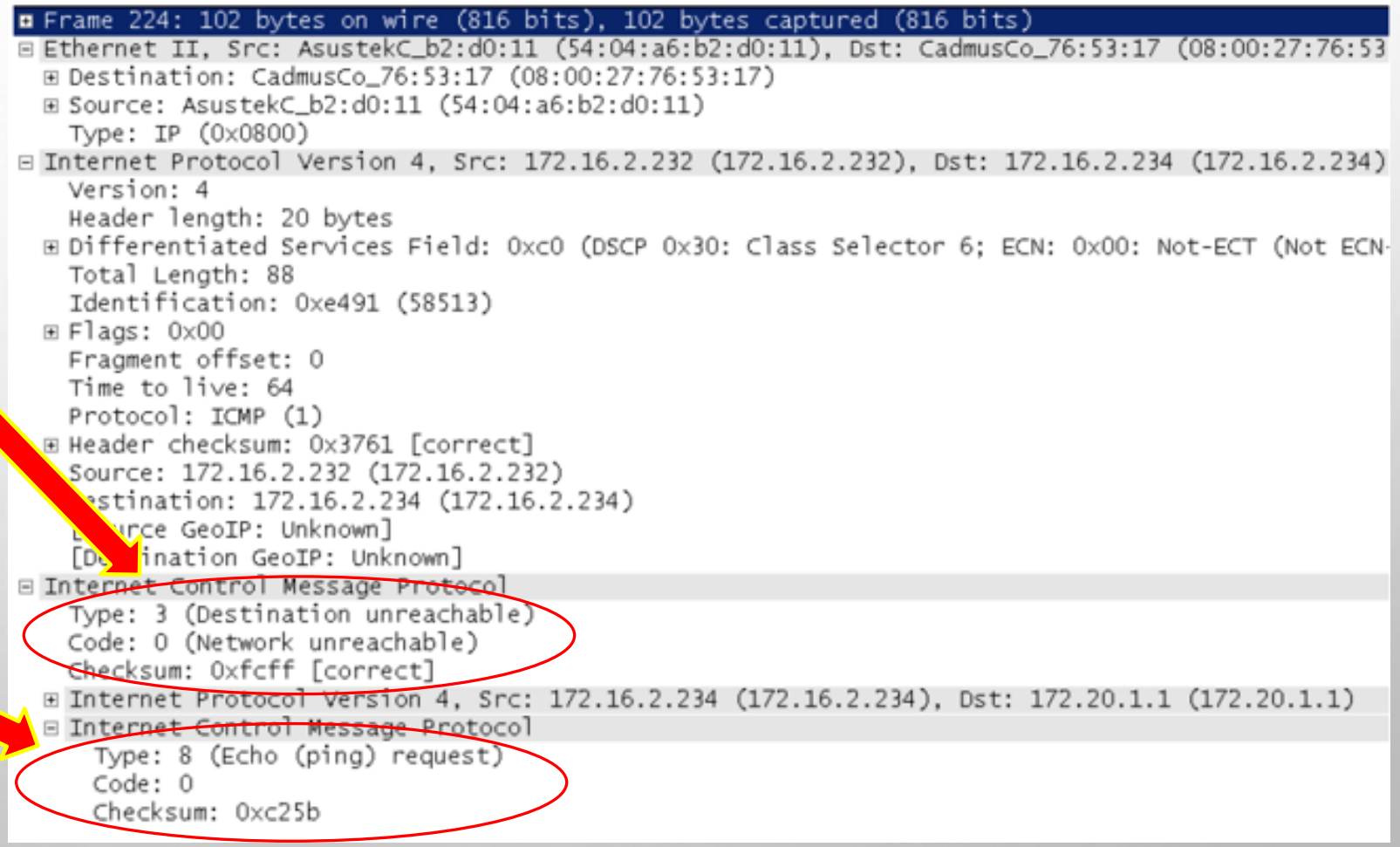
UN MESSAGE ICMP

- 172.16.2.234 veut envoyer un message (ping) à 172.20.1.1
- 172.20.1.1 est dans un autre réseau
- 172.16.2.232 est alors une passerelle



UN MESSAGE ICMP

- Type et code de l'erreur signallée par ICMP
- Message qui a causé le problème



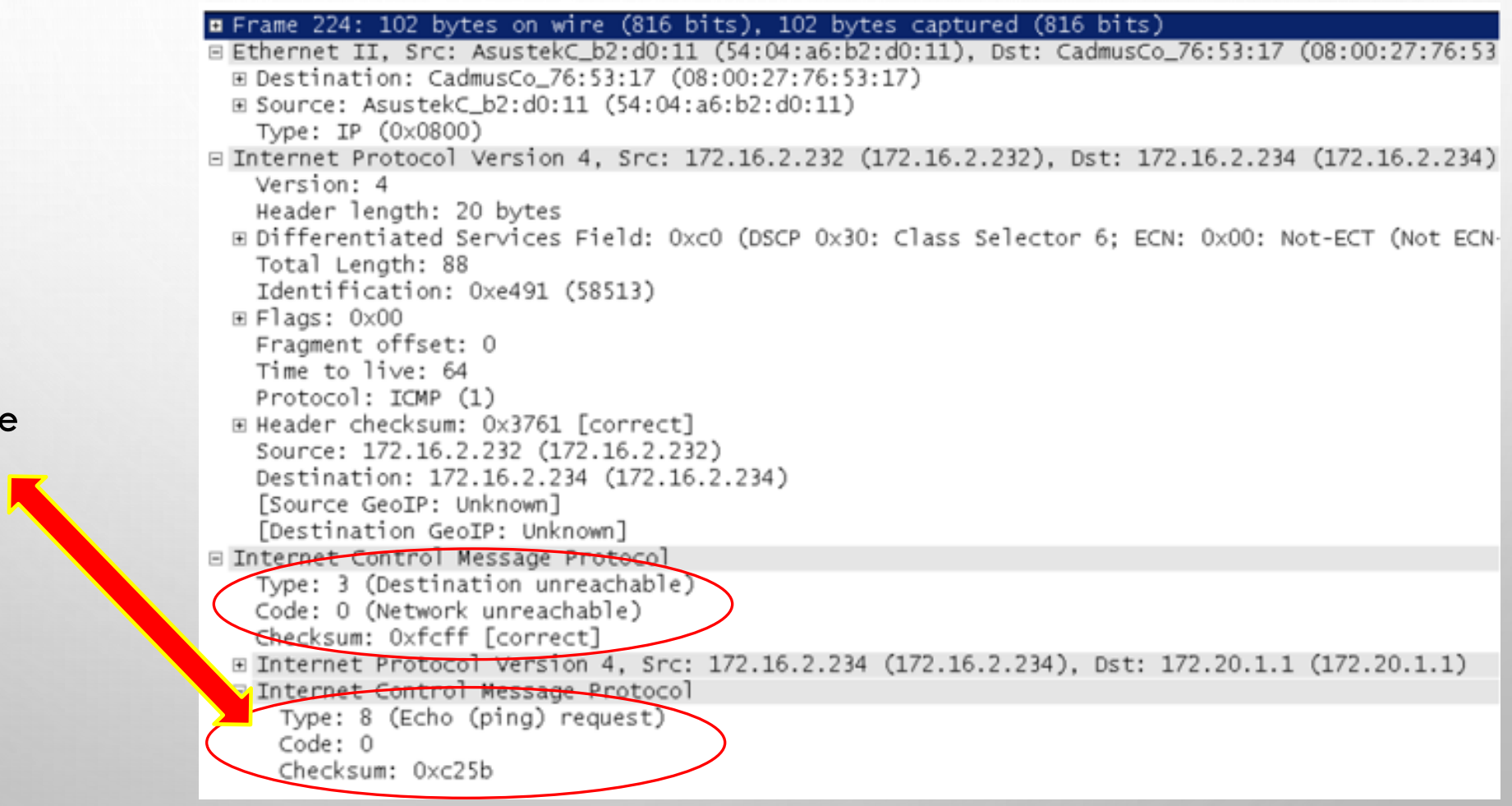
UN MESSAGE ICMP

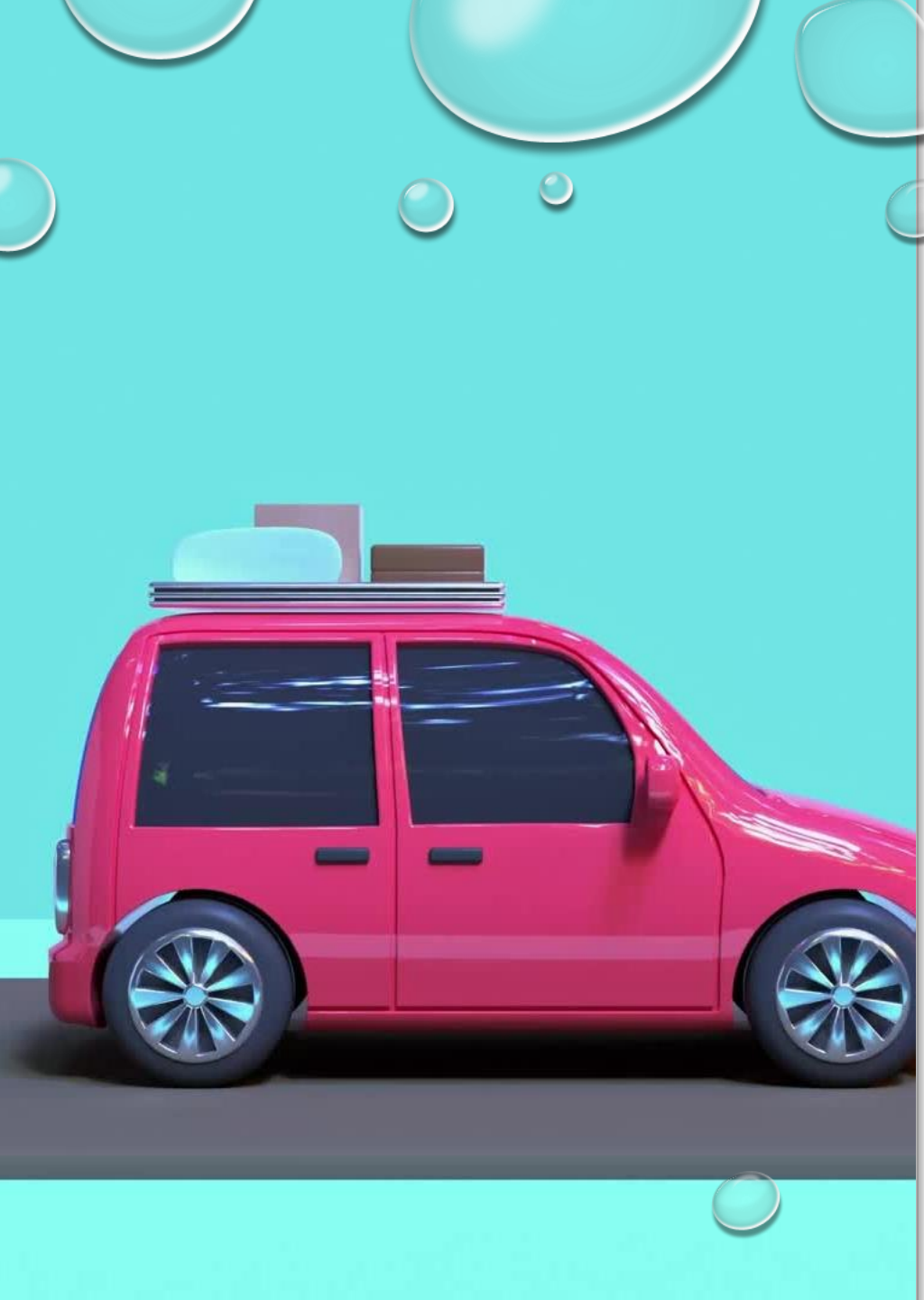
- Problème :

  Ping envoyé par

  172.16.2.234

On n'a pas pu contacter ce réseau !

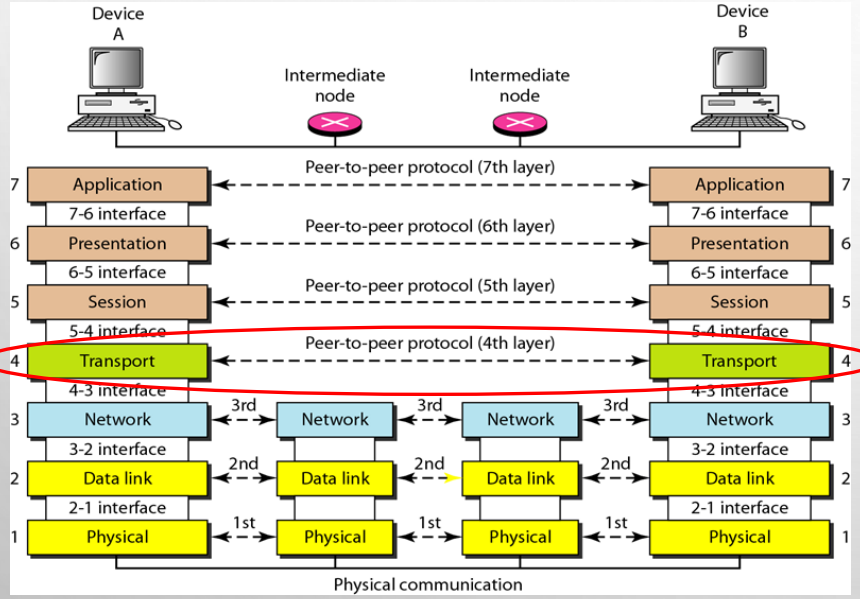




LA COUCHE TRANSPORT

LA COUCHE TRANSPORT

La couche transport



Objectif : la fiabilité des communications

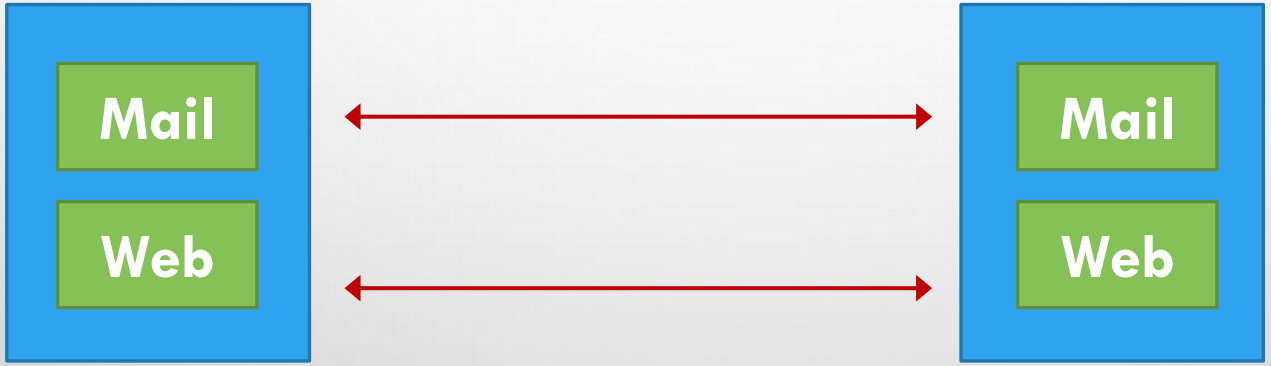
Couche transport : des sessions

- Sessions de communication : flow bidirectionnel
- Fiabilité : accusé de réception/renvoi
- Notion de service et protocole client/serveur

À cette couche : transmissions fiables, ports, protocoles UDP, TCP

LA PROBLÉMATIQUE DES PORTS

- DEUX ORDINATEURS PEUVENT ÉCHANGER DES MESSAGES CONCERNANT PLUSIEURS SERVICES



- UN SERVEUR PEUT HÉBERGER PLUSIERUS SERVICES
- ON VEUT QUE LE BON MESSAGE SOIT TRAITÉ PAR LE BON SERVICE

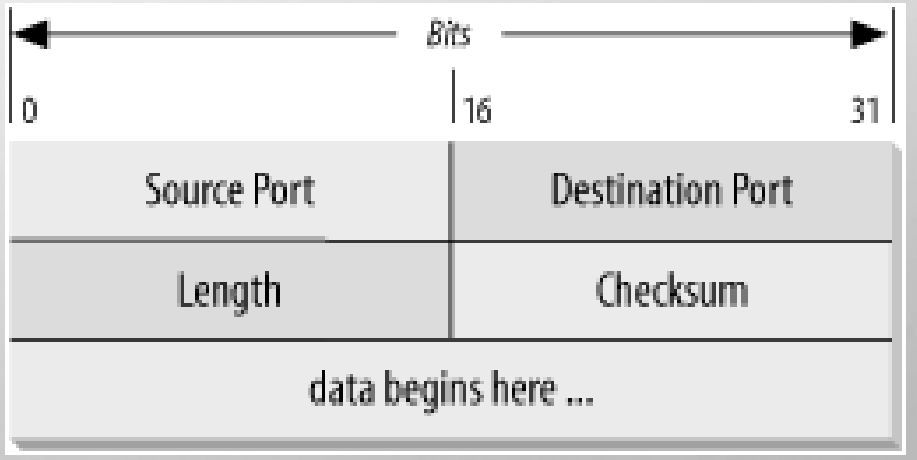
SOLUTION : LES PORTS

- CHAQUE SERVICE = UN NOUVEAU "PORT"

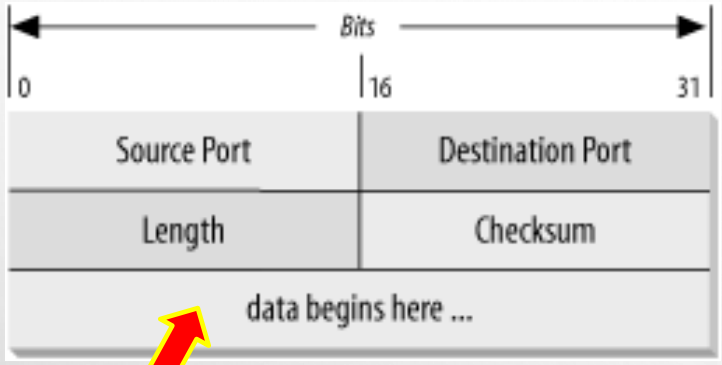
- 1 PORT = 16 BITS
- IL Y A DES PORTS RÉSERVÉS
- 1 APPLICATION ← → (ADRESSE IP, PROTOCOLE, PORT)

LE PROTOCOLE UDP

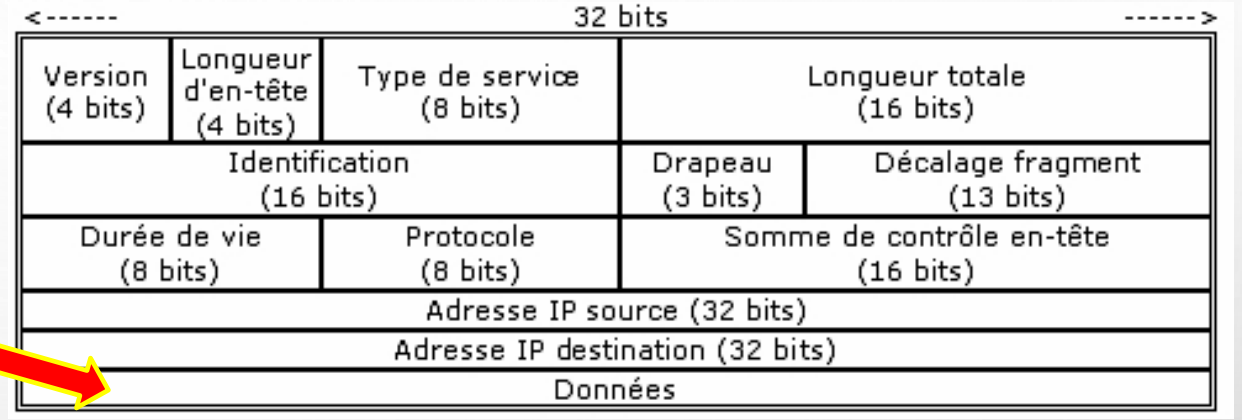
- Protocole basique de la couche transport
- Messages = datagrammes (d'où UDP = User Datagram Protocol)
- Spécifie le port source/destination
  - Peu de fiabilité garantie en dehors d'une checksum

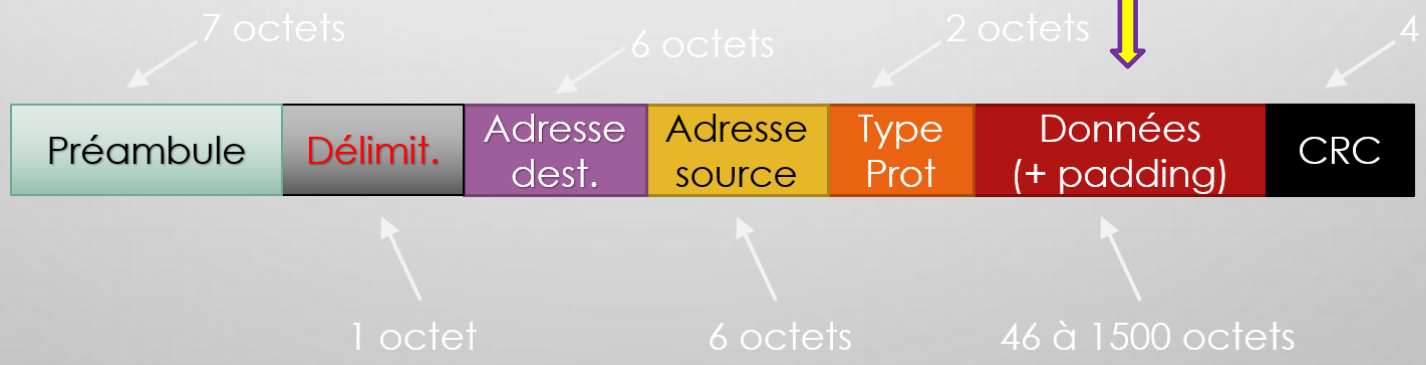


ENCAPSULATION

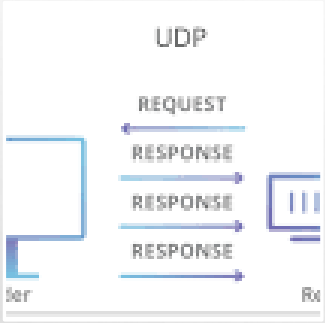


Message





LA TRANSMISSION FIABLE : TCP

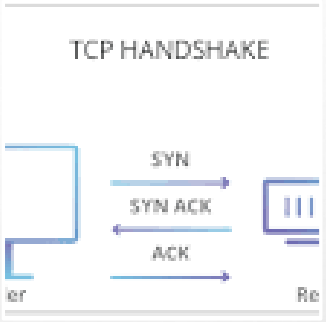


UDP permet :

Séparation messages par service : ports

Messages encapsulés : IP, puis Ethernet

Pas de transmission fiable



TCP permet :

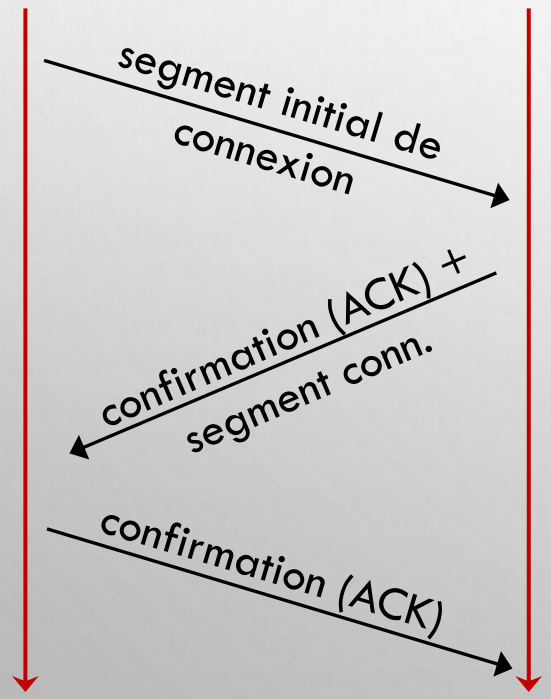
Transmission ordonnée des flux

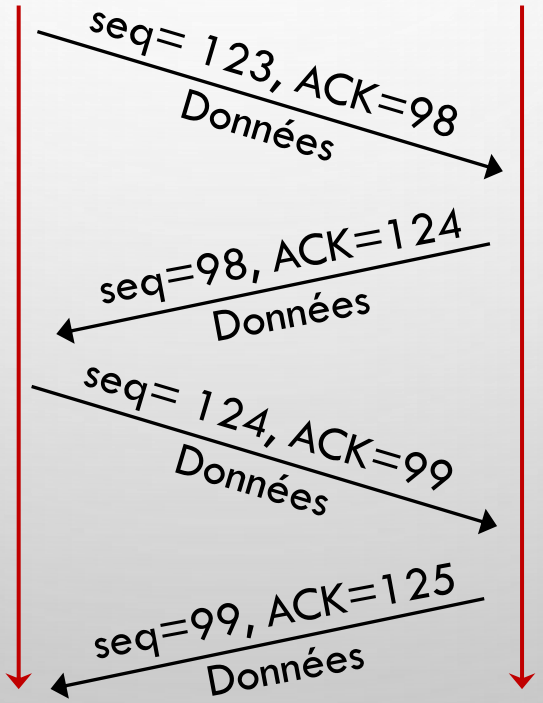
Correction d'erreurs

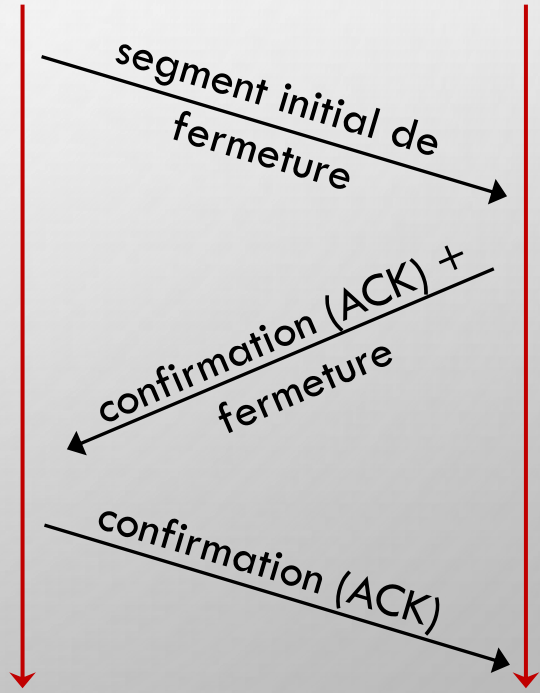
Transmission confirmée de messages

CONNEXION TCP

- LA CONNEXION S'OUVRE AVEC UNE SUITE DE MESSAGES
  - PUIS LES DEUX MACHINE ÉCHANGENT DES MESSAGES ET CONFIRMENT LEUR RÉCEPTION







PROTOCOLES CLIENT-SERVEUR

(COUCHE 7)

CLIENT, SERVEUR

- PROTOCOLE CLIENT-SERVEUR :
  - ❖ Les messages passent entre une machine client et une machine serveur
- MACHINE SERVEUR :
  - ❖ Toujours à l'écoute, souvent sur un port standard
  - ❖ Peut se connecter à un client
  - ❖ Peut jouer également le rôle d'un client
- MACHINE CLIENT :
  - ❖ Contacte le serveur (typiquement port non-standard)

LA COMMANDE SS

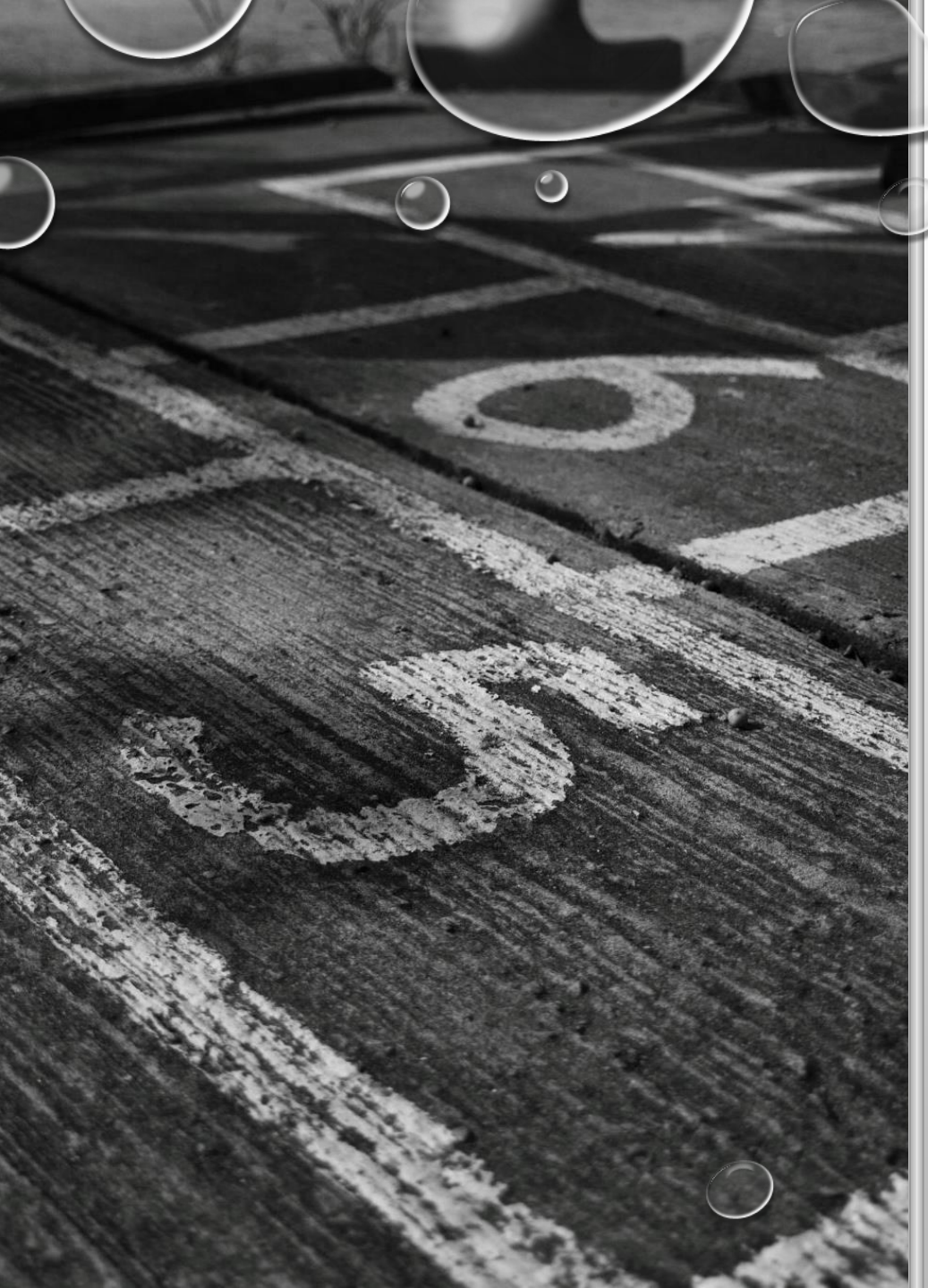
Les connexions d'une machine :

la commande ss

- Connexions liées à des protocoles différents (applications différentes)
- Statut : à l'écoute, en cours, en cours de finir...

Paramétrisation

- ss -ln ports écoute (en format numérique)
- ss -uan toute connexion sur udp, format numérique
- ss -tn connexions sur TCP, format numérique



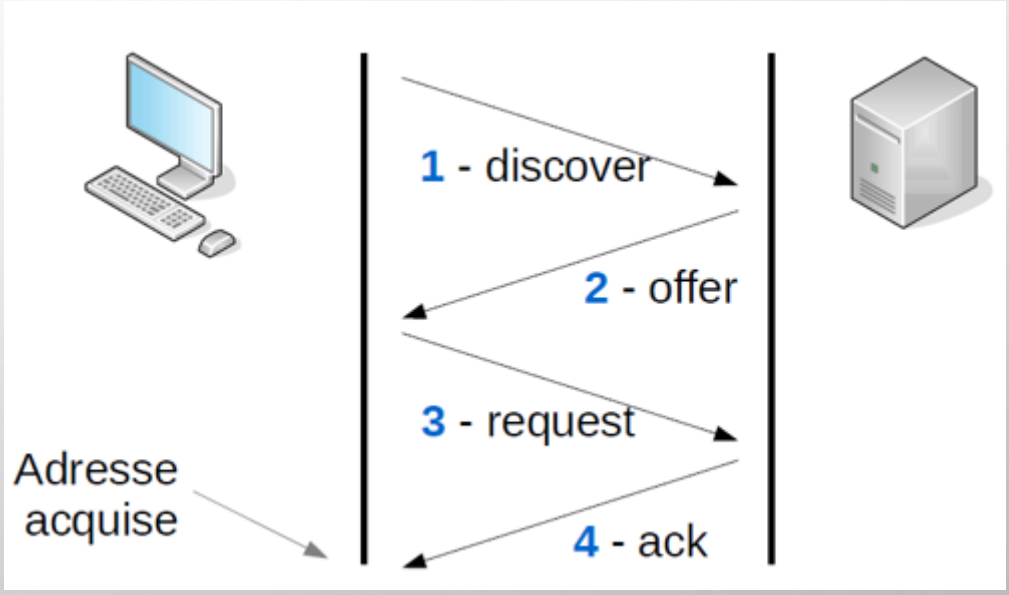
LE PROTOCOLE DHCP

PROBLÉMATIQUE

- On arrive dans un nouvel environnement (campus, hôtel, maison d'un ami…)
- On veut se connecter à son réseau
  - Une fois connectés, nous allons faire partie de son réseau
  - L'adresse physique ne change pas, mais on a une nouvelle adresse IP
- Pour la plupart on choisit de le faire automatiquement

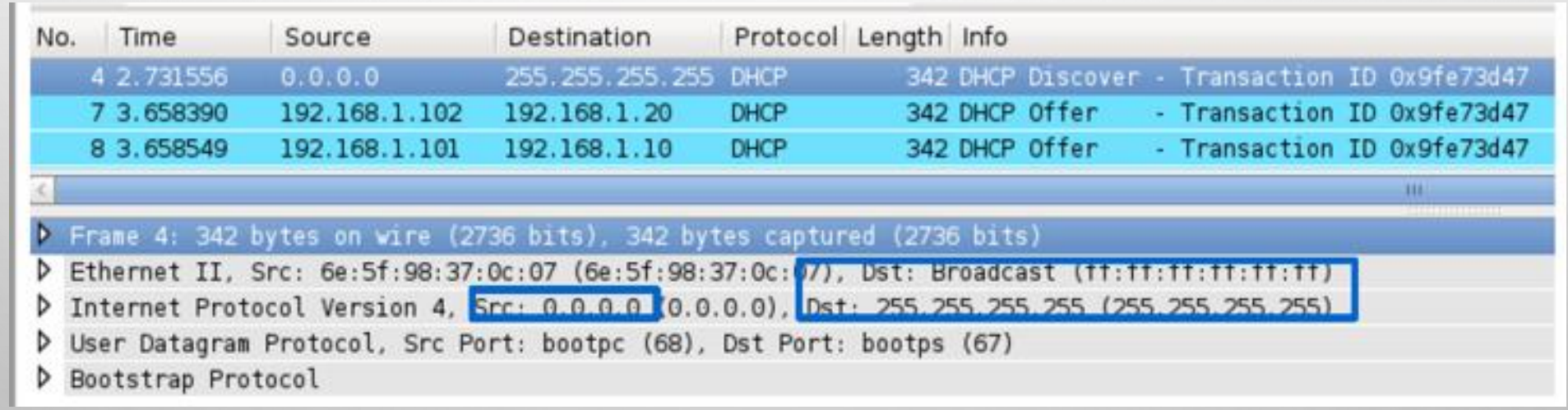
DHCP

- UN SERVEUR QUI DISTRIBUE AUTOMATIQUEMENT LA CONFIGURATION IP :
  - ADRESSE IP + MASQUE RÉSEAU
  - PASSERELLE PAR DÉFAUT (ROUTAGE)
  - MISE À JOUR DU SERVEUR DNS
- C'EST L'ISP QUI S'EN OCCUPE
  - GESTION CENTRALISÉE DU RÉSEAU
- 4 ÉTAPES



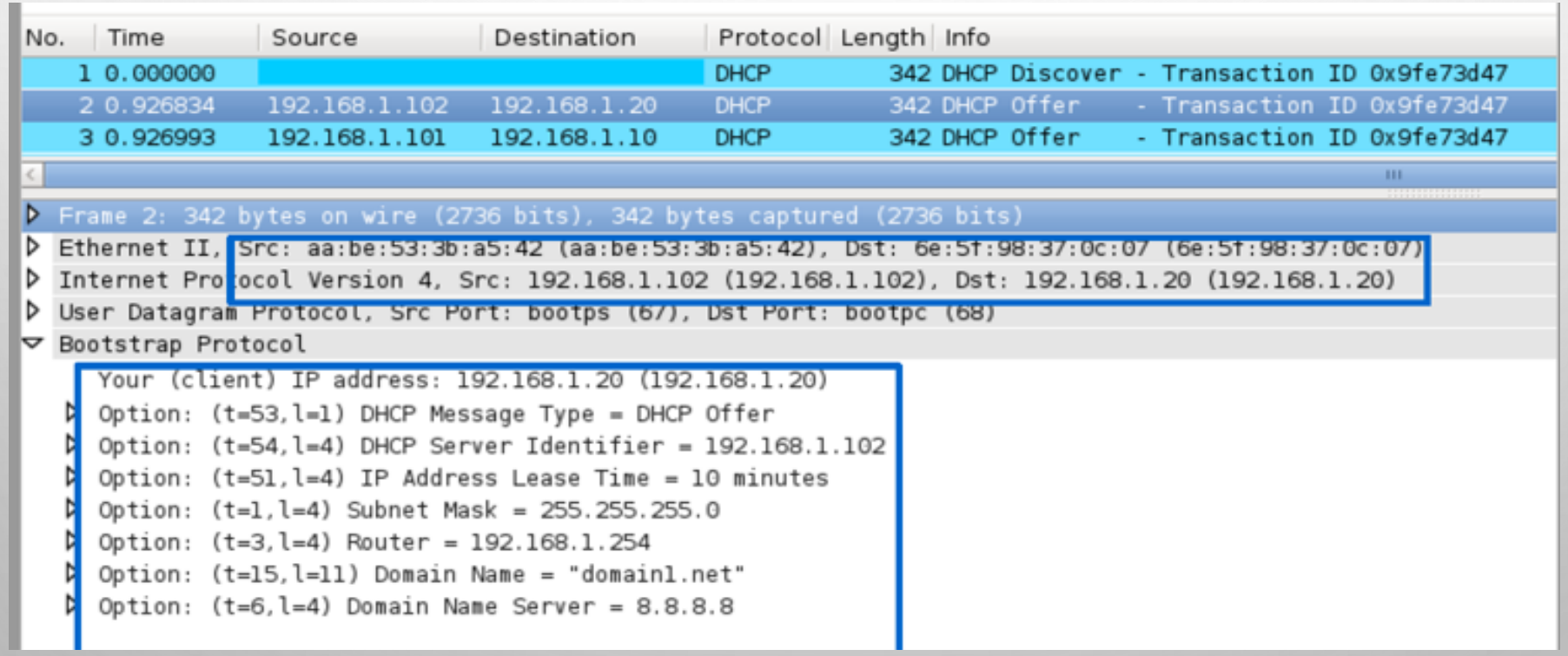
ETAPE 1 : DISCOVER

- LE NOUVEAU ORDINATEUR VEUT RECEVOIR UNE ADRESSE IP
  - IL NE CONNAÎT PAS SON ADRESSE IP
  - IL FAIT UN BROADCAST



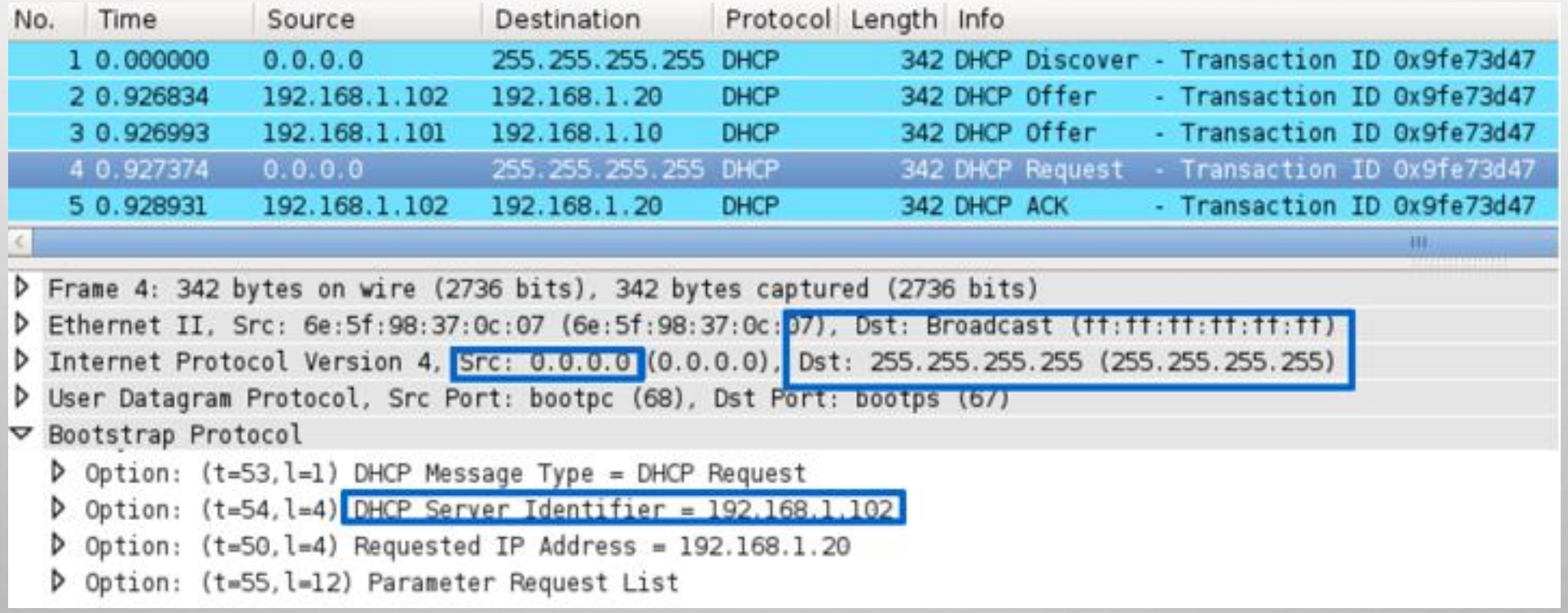
ETAPE 2 : OFFER

- LE SERVEUR DHCP A REÇU SON MESSAGE ET LUI PROPOSE UNE ADRESSE IP
  - CETTE OFFERTE EST LIMITÉE POUR UN NOMBRE DE MINUTES INITIELLEMENT



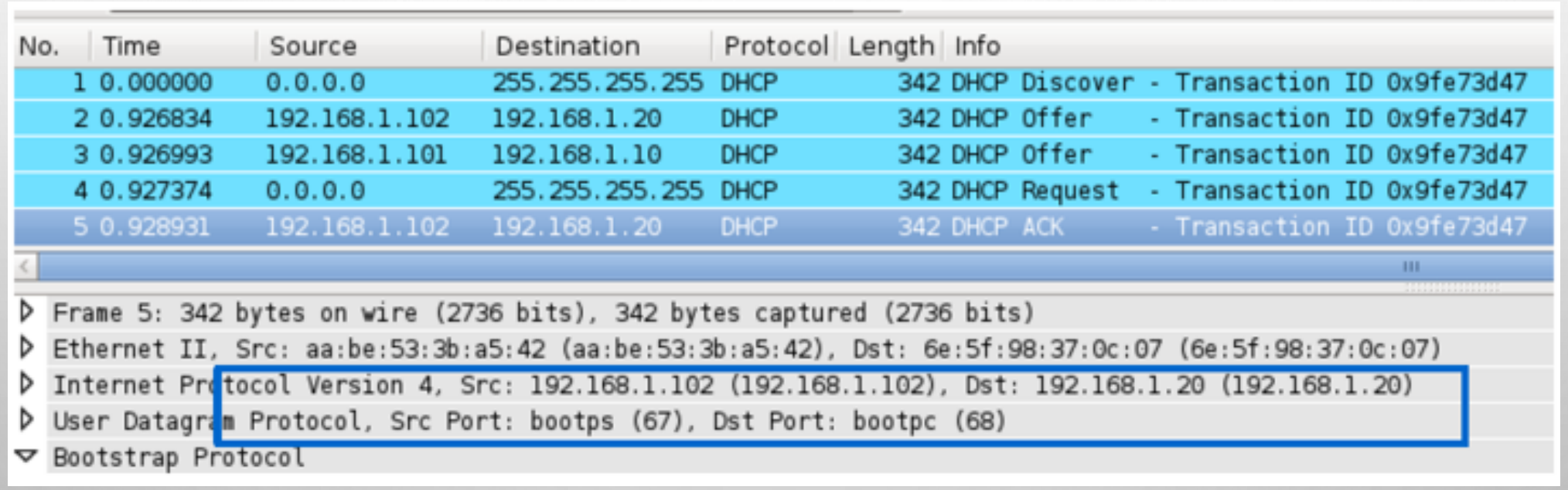
ETAPE 3 : REQUEST

- LE NOUVEL ORDINATEUR CHOISIT DE DEMANDER L'ADRESSE QU'ON LUI A DONNÉ
  - L'ADRESSE NE VA PLUS EXPIRER



ETAPE 4 : ACKNOWLEDGEMENT

- LE SERVEUR CONFIRME L'ALLOCATION DE LA NOUVELLE CONFIG. IP



CONFIGURATION DHCP

- CÔTÉ SERVEUR :
  - ❖ Configuration fichier dhcpd.conf
    - ❖ Spécifier sur quelle interface on fiat DHCP
    - ❖ Spécifier : plage d'adresses à utiliser, optionnellement quel routeur à utiliser, DNS...

- CÔTÉ CLIENT :
  - ❖ Obtenir une adresse dynamiquement : dhclient eth0
  - ❖ Configuration perenne : fichier /etc/network/interfaces

FICHIER INTERFACES

- CONFIGURATION PERENNE DES INTERFACES D'UNE MACHINE

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.56.11
netmask 255.255.255.0
gateway 192.168.56.1

auto eth1
iface eth1 inet dhcp
```

Loopback, interface lo

Config. statique, eth0

Config. par DHCP

DIFFÉRENCES DHCP, INTERFACES

- LE FICHIER /ETC/NETWORK/INTERFACES INDIQUE :
  - ❖ L'adresse IP + routeur de la machine elle-même
  - ❖ Pas de possibilité de configuration DNS

- LE FICHIER /ETC/DHCP/DHCPD.CONF INDIQUE :
  - ❖ Une plage d'adresses IP qui seront données à une machine cherchant une configuration dynamique
  - ❖ Un routeur que les machines prenant une config. dynamique peuvent utiliser
  - ❖ Un serveur DNS pour ces machines aussi ...

INTRODUCTION À KATHARÁ

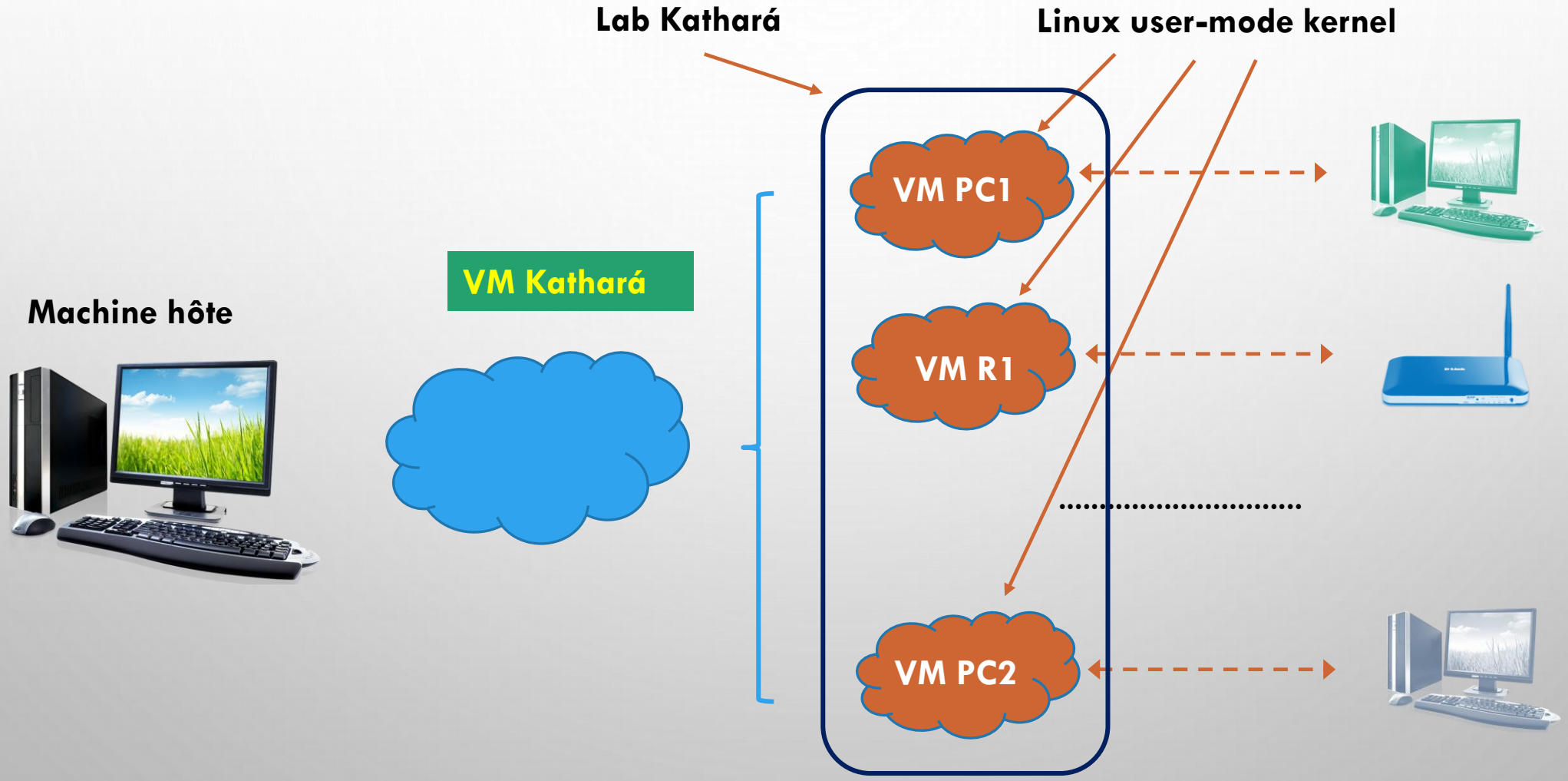
L'OUTIL KATHARÁ

- PERMET DE SIMULER LE FONCTIONNEMENT DE TOUT UN RESEAU SUR UNE MACHINE

- UN LOGICIEL QUI PERMET DE CRÉER DES CONTAINEURS SUR UNE **MACHINE HÔTE**
  - Chaque containeur est une VM Linux
  - Chaque VM joue le rôle d'un élément d'un réseau : un PC, un routeur, un serveur...
  - Toutes ces VMs sont dans un seul VM dans lequel on exécute Kathará
  - On peut manipuler VM par VM, ou faire toute la simulation fonctionner à partir de Kathará

- CECI NOUS PERMET DE SIMULER LE FONCTIONNEMENT D'UN RÉSEAU EN TOUTE SÉCURITÉ

L'INFRASTRUCTURE AVEC KATHARÁ



LES LABS KATHARÁ

- 1 LAB = 1 REPERTOIRE (CRÉÉ EN DÉBUT DU TP)

- LAB KATHARÁ : TOPOLOGIE SPÉCIFIÉE DANS UN FICHIER LAB.CONF
  - Lab.conf indique l'interconnexion physique entre les machines
  - Les machines partageant un réseau/médium physique forment 1 domaine de collision

- 1 MACHINE = 1 MACHINE VIRTUELLE
- DES FICHIERS POUR DÉCRIRE LE COMPORTEMENT DES MACHINES AU START ET À L'ARRÊT

LA CONFIGURATION D'UN LAB : LAB.CONF

<nom_machine>[<#>] : interface eth<#> de la machine <nom_machine>

net0, net1 : deux domaines de collision distincts, domain0, domain1

Dessinez la topologie indiquée par ce script

Démarrage du lab une fois lab.conf fini :

kathara lstart

```
>> sudo nano lab.conf

pca[0]=net0
pcb[0]=net0
pcb[1]=net1
pcc[0]=net1
```

LES FICHIERS .STARTUP

- CHAQUE MACHINE PEUT AVOIR UN FICHIER <NOM MACHINE>.STARTUP
  - UNE INSTRUCTION TYPIQUE DANS UN TEL FICHIER POURAIT ÊTRE

IP ADDRESS ADD 192.168.1.2/24 DEV ETH1

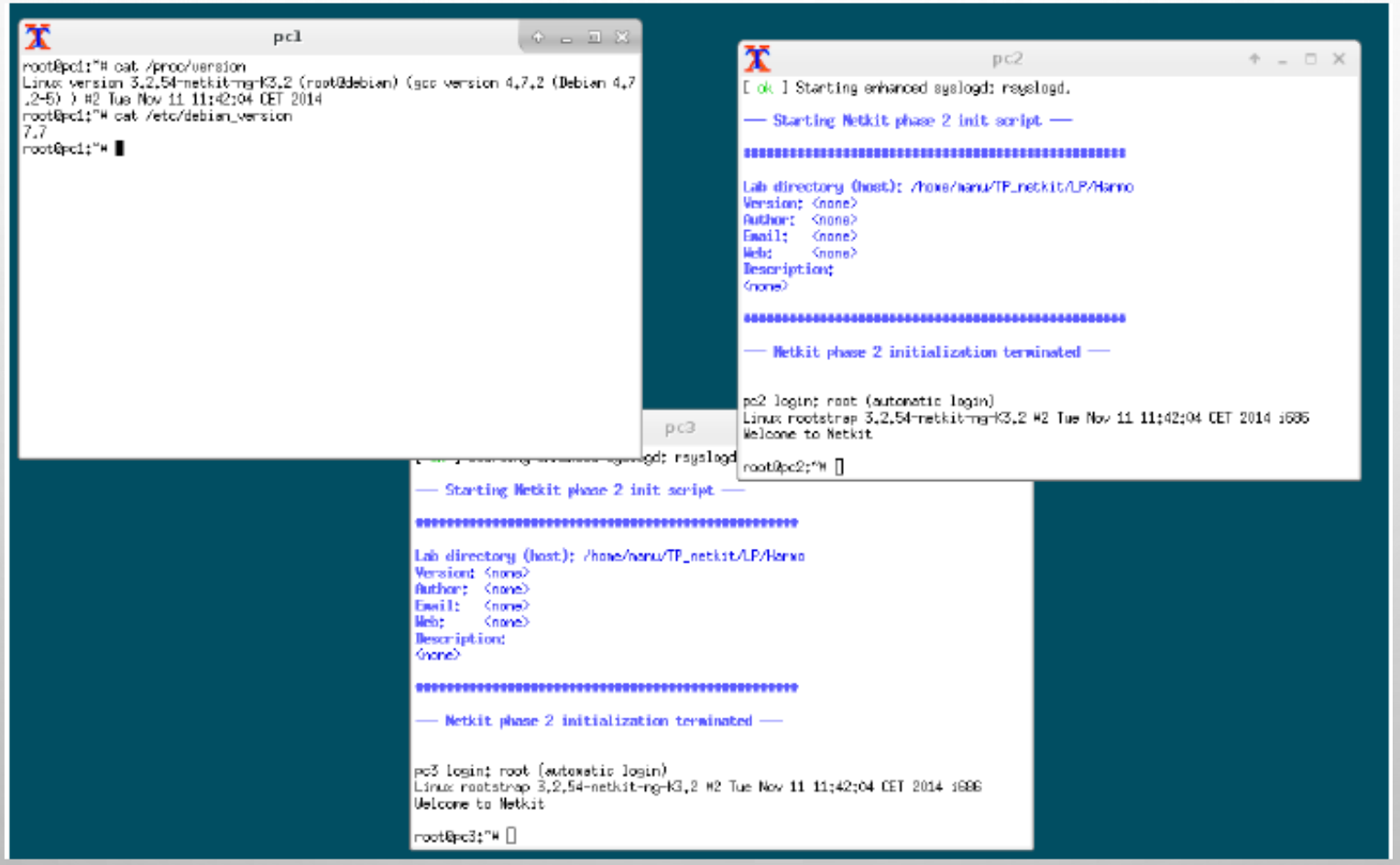
IP LINK SET DEV ETH1 UP

- ARRÊT PROPRE D'UN LAB :

  kathara.lclean

  kathara.wipe

LES TROIS VMS INITIALISÉES



UNE CONNEXION INTERNET

- Le logiciel Kathará sert à simuler des configurations de réseau avant de les mettre en pratique
  - Pour anticiper des crashes, des difficultés concernant le routage…
- Parfois, même pour une simulation il nous faut une connexion Internet
- Faisable en Kathará en utilisant des bridges :
  - Le pont crée sur PCC une nouvelle interface (eth1) avec une adresse de classe B, connectée à une nouvelle interface créée sur la machine hôte

```
>> sudo nano lab.conf

pca[0]=net0
pcb[0]=net0
pcb[1]=net1
pcc[0]=net1
pcc[bridged]=true
```