Concepts MAN

IP networks

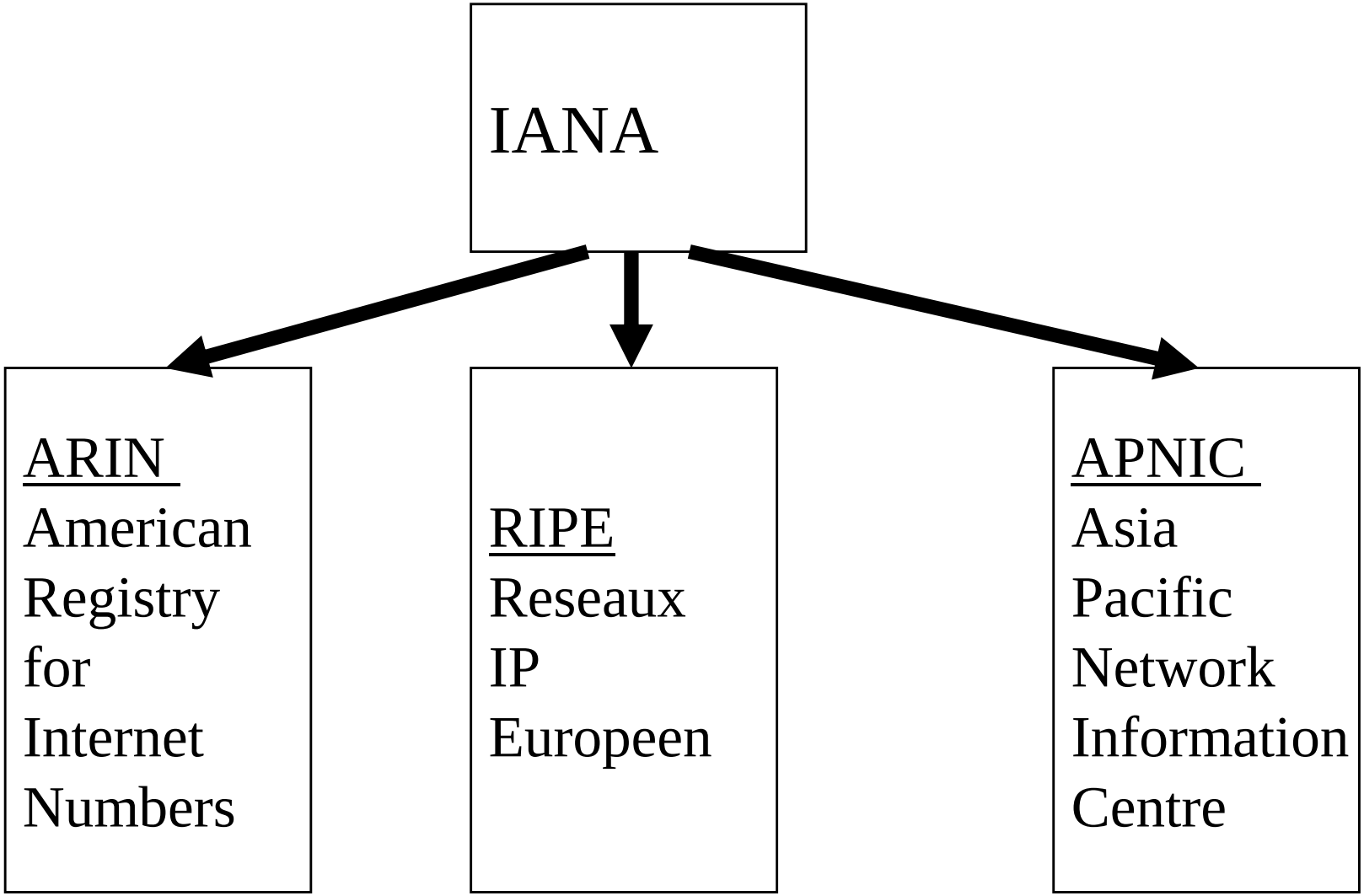
Eric Gaillard - 2015

EPITA - MAJEURES SRS & TCOM

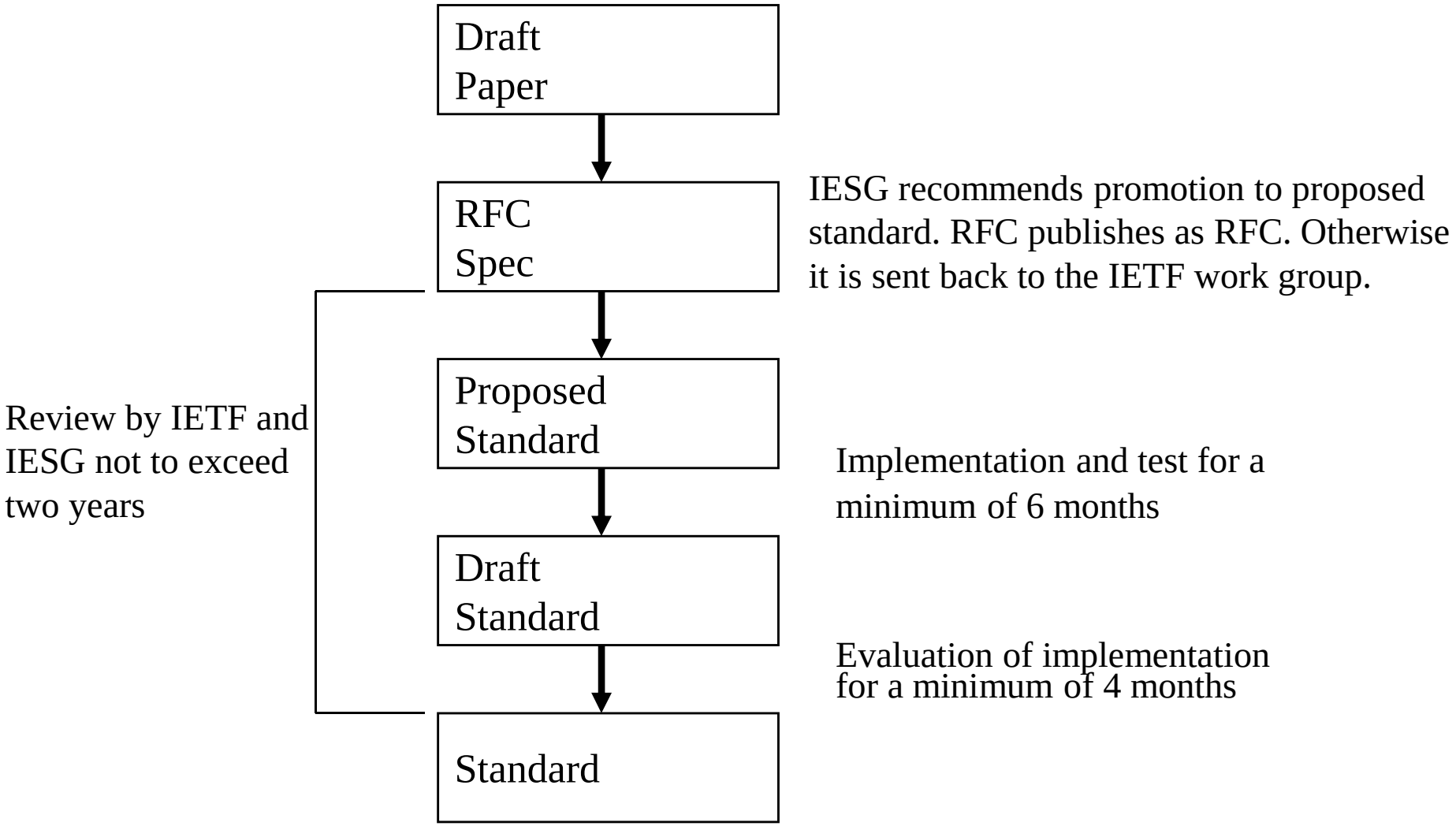
TCP/IP : pourquoi ?

- **Arguments**
  - Absence de norme ou lenteur d'émergence mal ressentie
  - Volonté de faire dialoguer des systèmes hétérogènes
  - Les spécifications de TCP/IP sont du domaine public
  - TCP/IP n ’est pas lié à un constructeur
  - TCP/IP est robuste et éprouvé
  - Support du réseau Internet
  - Support de la communauté universitaire
  - IP est adapté à l'interconnexion de réseau
  - TCP/IP est indépendant des réseaux physiques
  - TCP/IP était natif sur UNIX BSD
  - Les sources logicielles de TCP/IP sont facilement accessibles
- **Devenu un standard de facto**

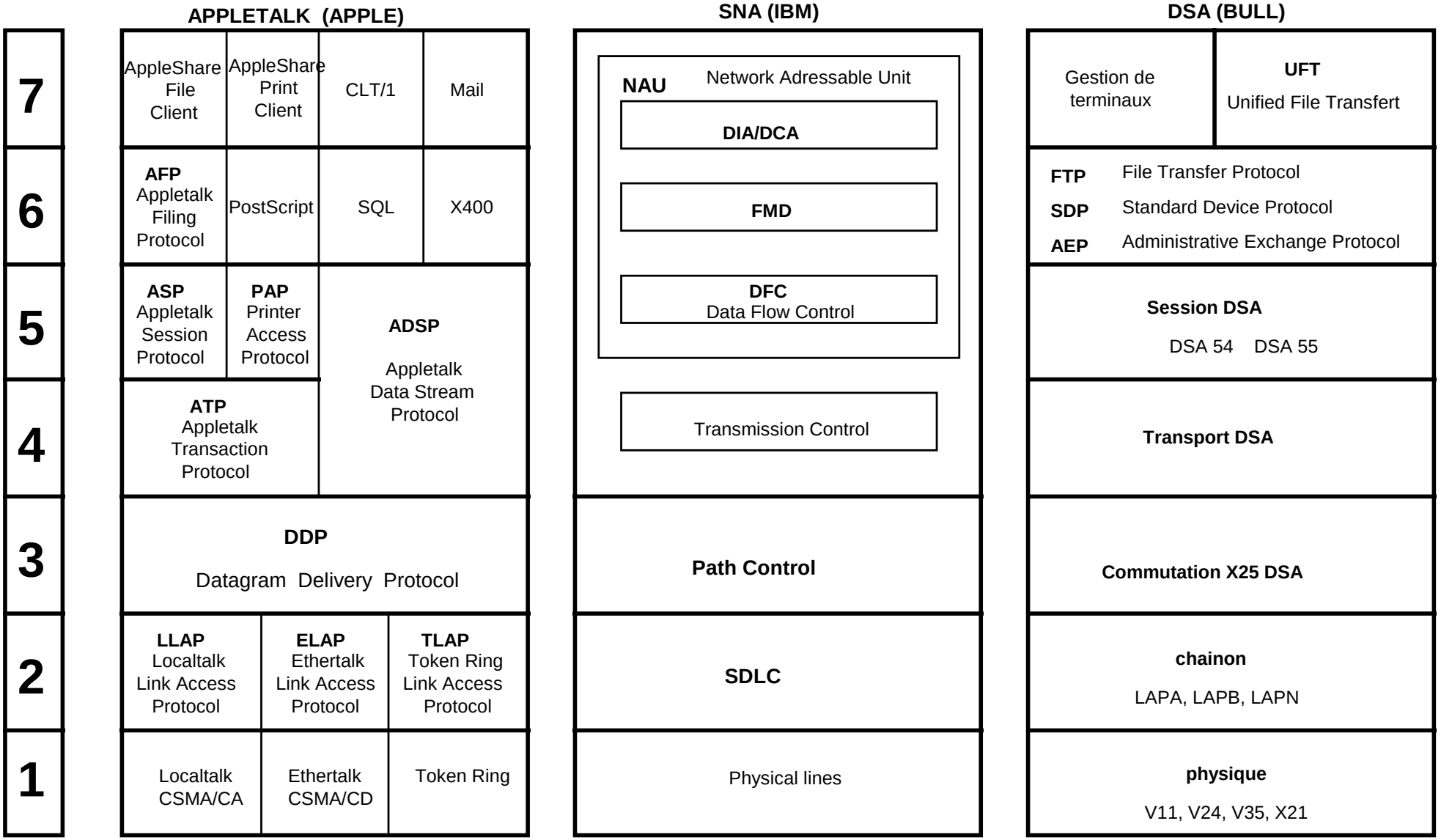
Internet Assigned Number Authority (IANA)



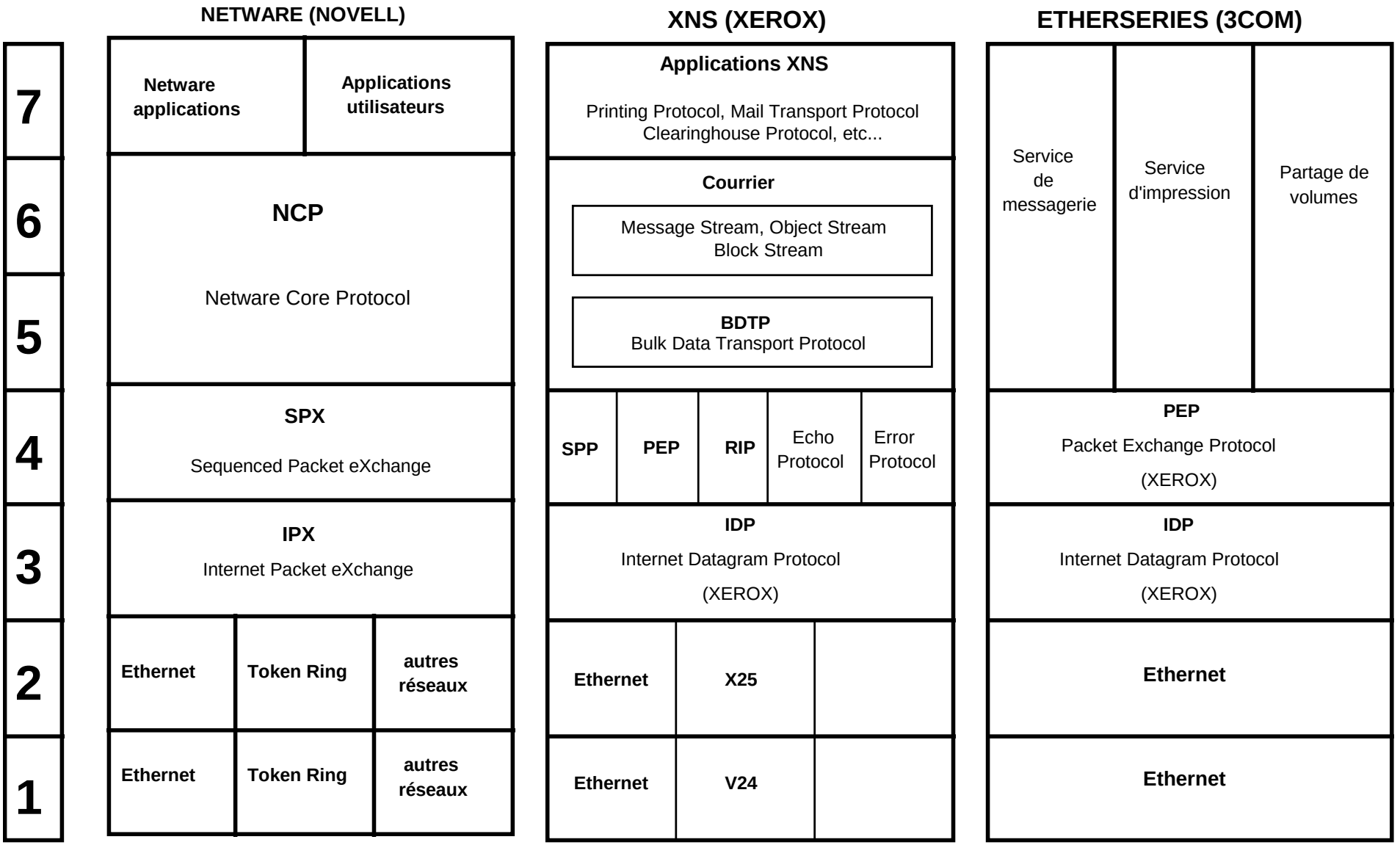
Processus de standardisation



Généralités: les architectures protocolaires



Généralités: les architectures protocolaires



Généralités: les architectures protocolaires

OSI (ISO)

FTAM ISO 8571	**MHS** ISO 8505	**ODA** ISO 8613	**VT** ISO 9040/41	**JTM** ISO 8831/32
ISO 8822 services **ISO 8823** protocole				
ISO 8326 services **ISO 8327** protocole				
ISO 8072 services **ISO 8073 classe 0,1,2,3 et 4**				
ISO 8348 services **ISO 8473** protocole "internet"				
ISO 8802/2				
ISO 8802/3 CSMA/CD (Ethernet)	**ISO 8802/4** Bus à jeton	**ISO 8802/5** Anneau à jeton (Token Ring)		
ISO 8802/3	**ISO 8802/3**	**ISO 8802/3**		

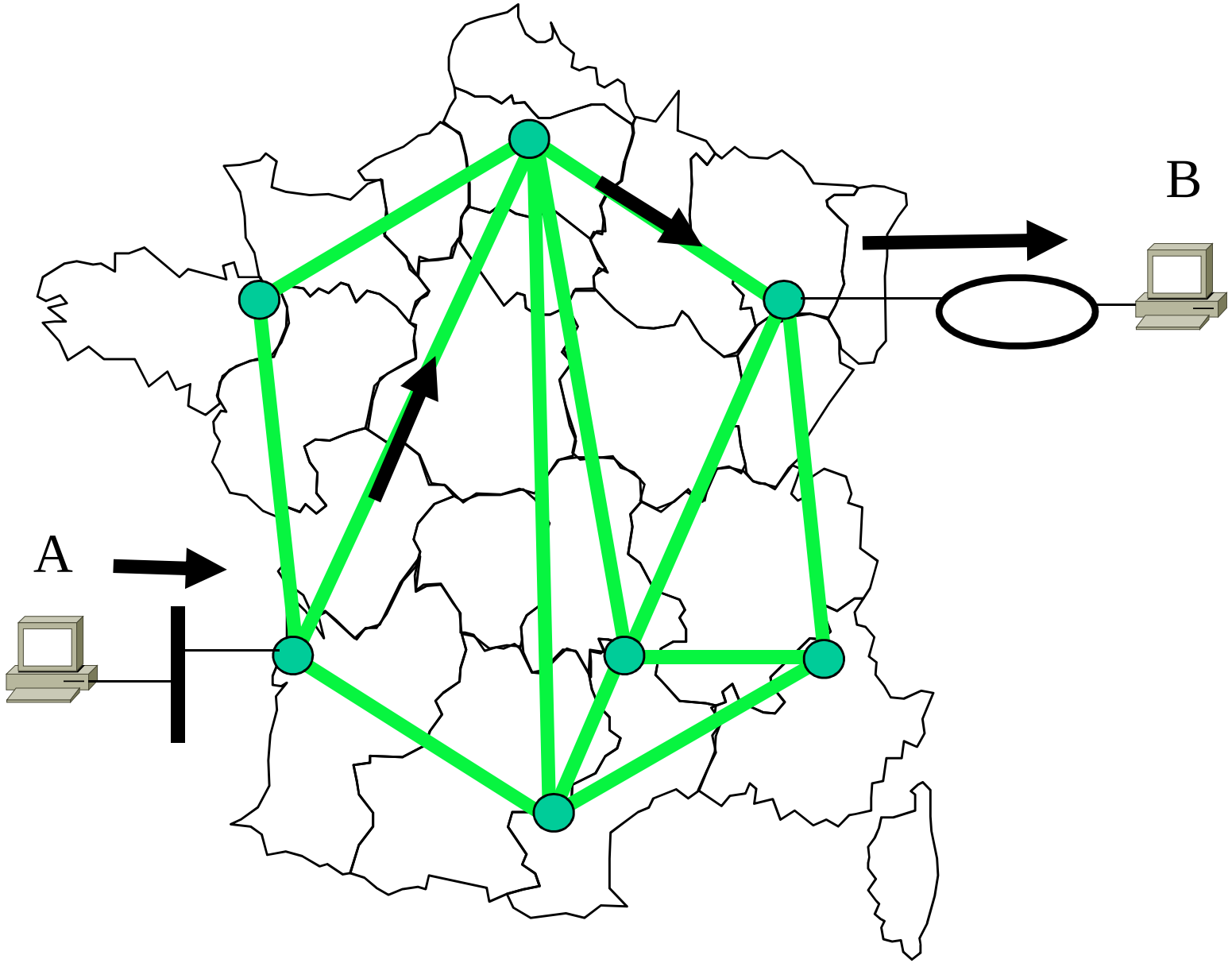
Architecture TCP/IP

TFTP **SNMP**	**Telnet**	**FTP**	**SMTP**
UDP User Datagram Protocol	**TCP** Transmission Control Protocol		
IP Internet Protocol			
Ethernet	**Token-Ring**	**FDDI**	
Ethernet	**Token-Ring**	**FDDI**	

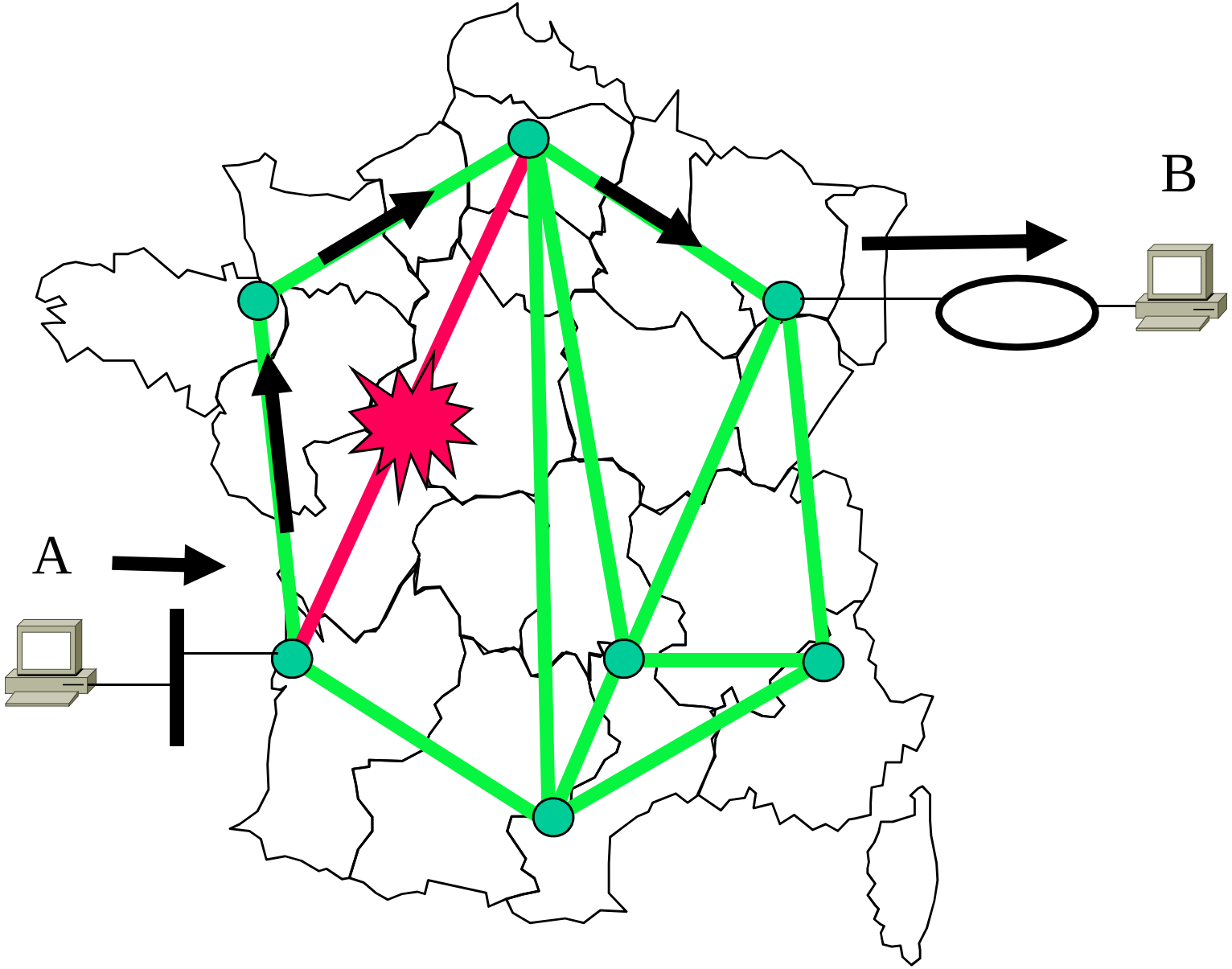
TCP/IP : historique

- Conception au milieu des années 70
- DARPA (Defense Advanced Projects Research Agency)
- DoD (Departement Of Defense)
- TCP/IP sur ARPANET au début des années 80
- BBN (Bolt Beranek & Newman) : TCP/IP sous UNIX
- Université de Berkeley
- 60000 noeuds interconnectés
- RFC

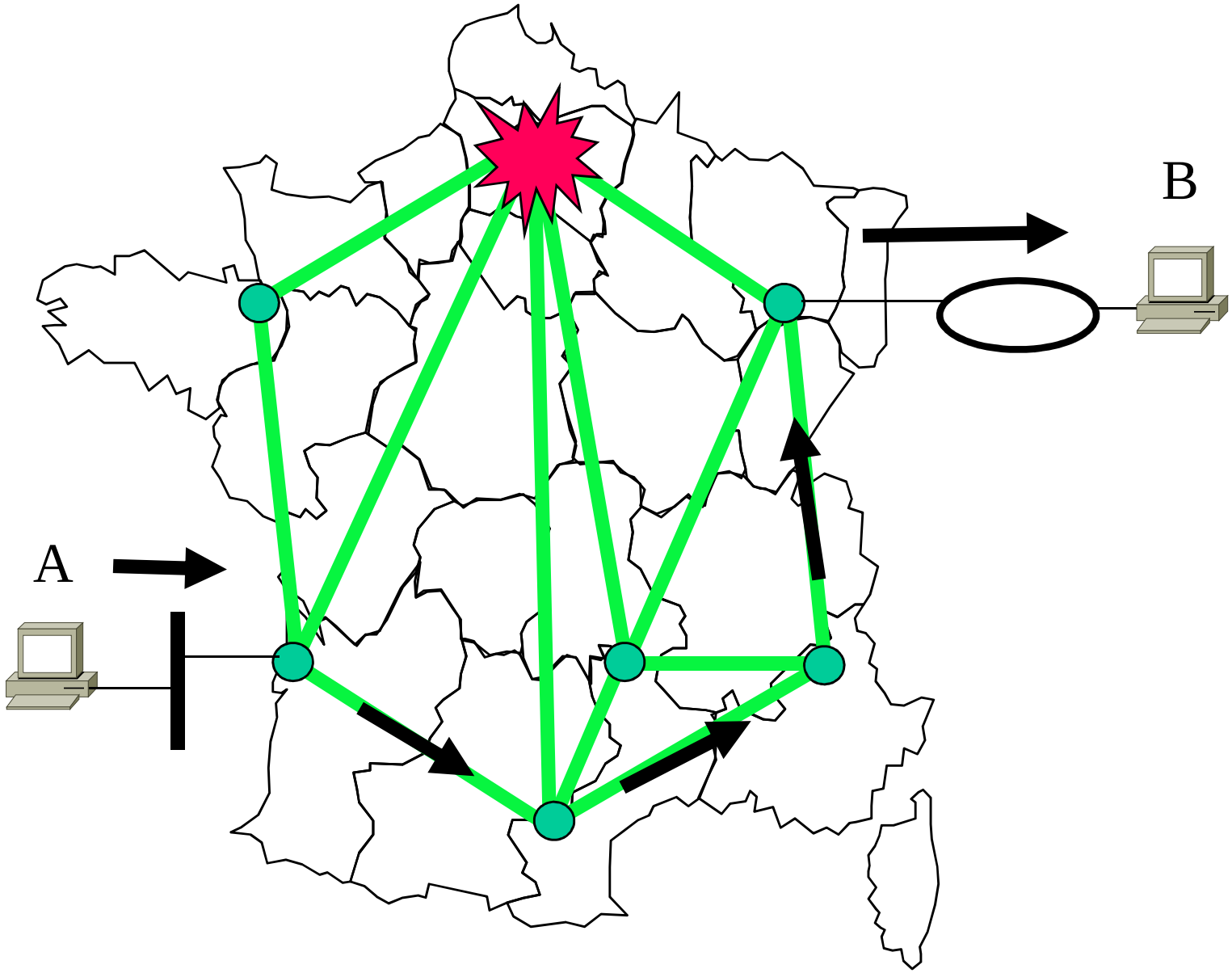
TCP/IP : généralités



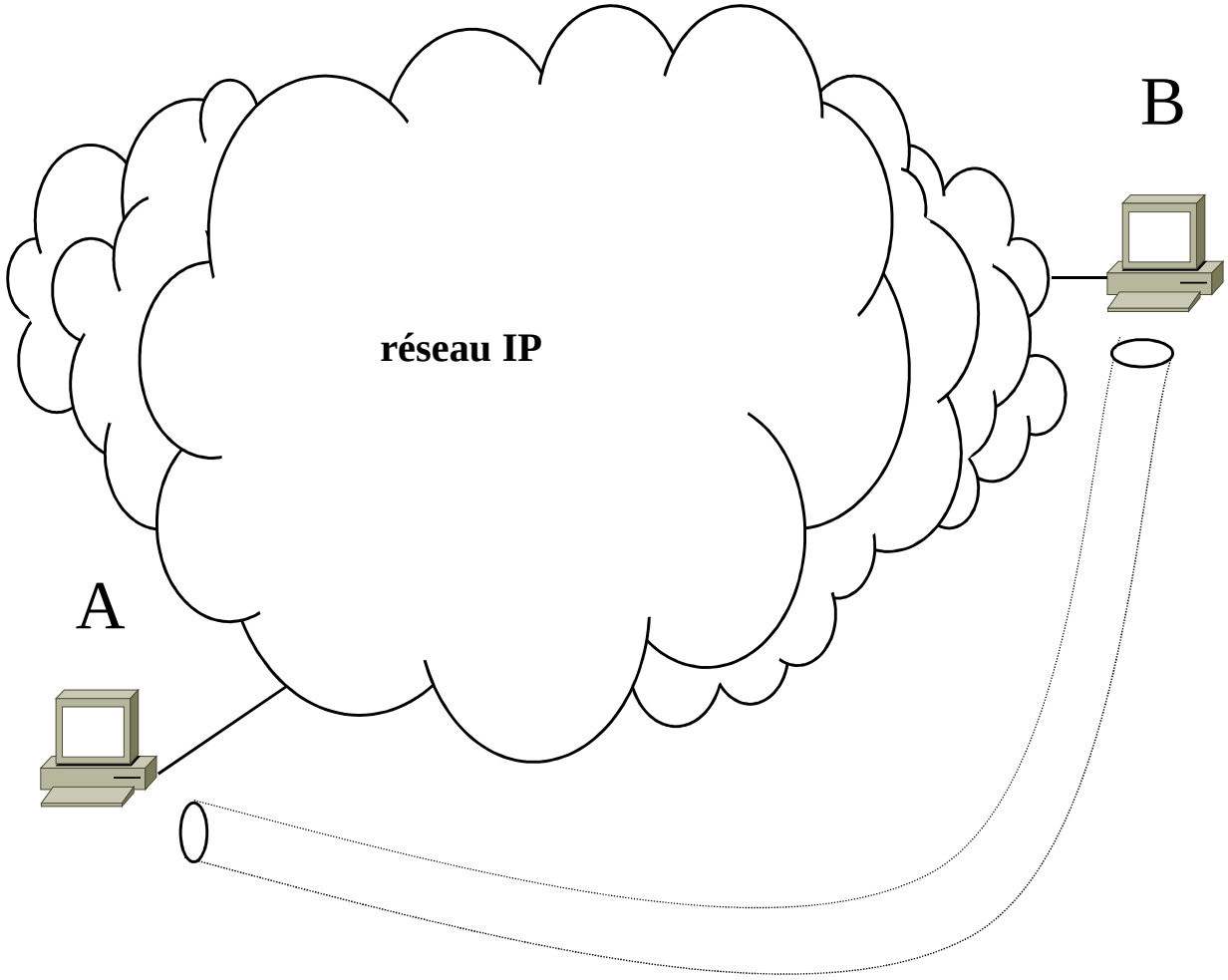
TCP/IP : généralités



TCP/IP : généralités



TCP/IP : généralités

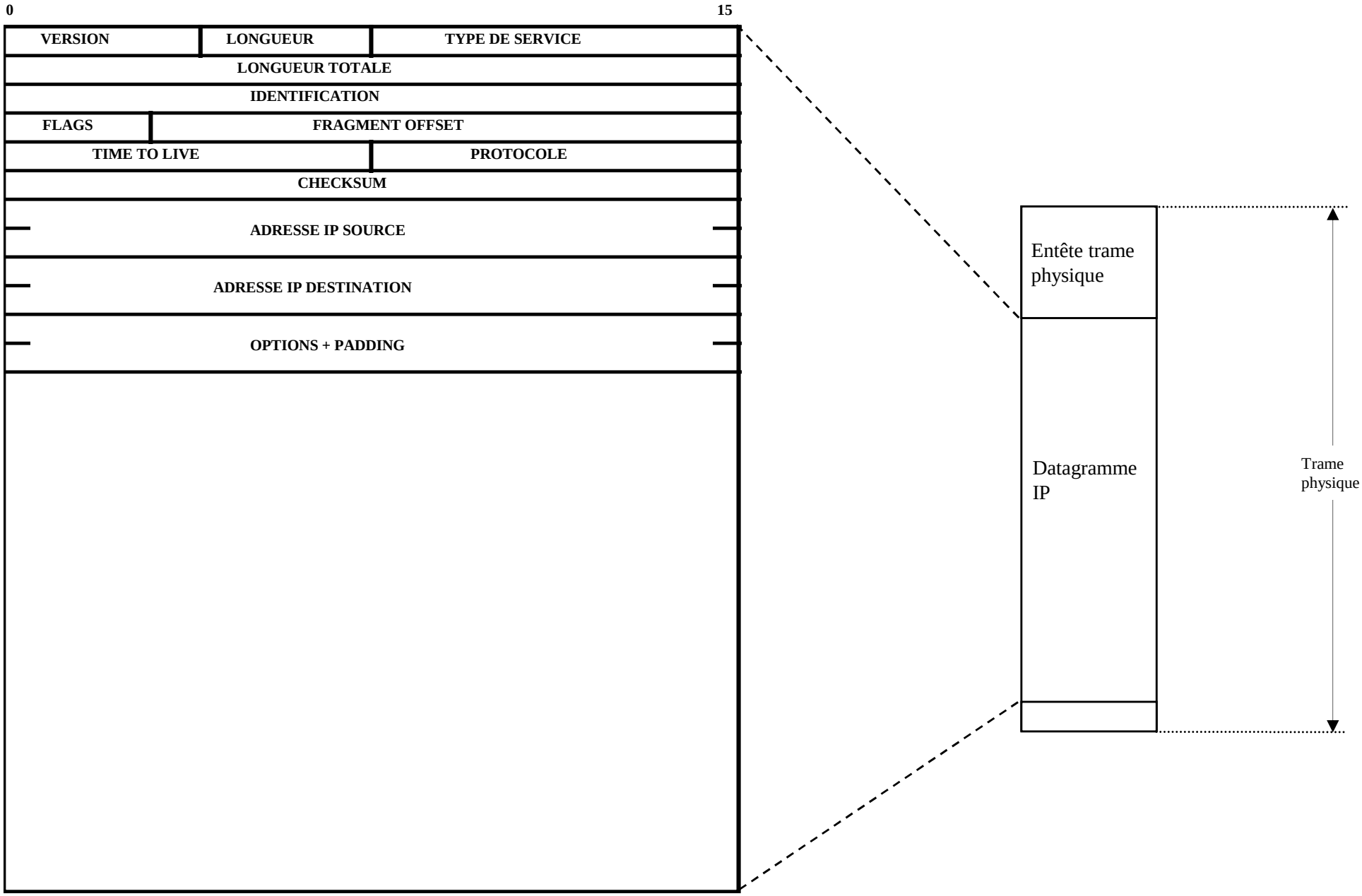


Connexion TCP

IPv4

- Internet Protocol
- RFC 791 / MIL-STD-1777
- IP est un protocole de niveau réseau qui fonctionne en mode à datagramme
- Il offre des services d ’adressage, de routage et de fragmentation
- IP est indépendant des réseaux « physiques »
- Les datagrammes IP peuvent être acheminés sur X.25, liaison modem, RNIS, Frame relay, ATM,

IP : format des datagrammes



IP : format des datagrammes

- **VERSION** **4 bits**
  - Numéro de version du protocole utilisé
  - Version 4 = version en cours = IPv4
  - Version 6 = nouvelle version = IPv6
- **LONGUEUR** **4 bits**
  - Longueur totale de l'entête IP exprimée en mots de 32 bits
  - Longueur par défaut (si aucune option) = Longueur Min = 5
  - Longueur Max = 15
- **TYPE DE SERVICE** **8 bits**
  - Priorité (3 bits) 111= Network control / 110 = Internetwork Control / 101 = CriticECP / 100 = Flash override/ 011 = Flash / 010 = Immediate / 001 = Priority / 000 = Routine
  - Delay 0 = Normal / 1 = bas
  - Troughput 0 = Normal / 1 = haut
  - Reliability 0 = Normal / 1 = haut
  - 2 derniers bits réservés

IP : format des datagrammes

- **LONGUEUR TOTALE 16 bits**
  - Longueur totale du datagramme IP (entête + données) exprimée en octet
  - Longueur par défaut = Longueur Min = 20
- **IDENTIFICATION 16 bits**
  - Identifie le paquet IP, utilisé pour reconstruire le datagramme après fragmentation
  - Chaque fragment a le même Id
- **FLAGS 3 bits**
  - 0 DF MF
  - Don ’t Fragment = 1 : fragmentation interdite / DF = 0 : fragmentation autorisée
  - More Fragment = 1 : d ’autres fragments suivents / MF = 0 : dernier fragment
- **FRAGMENT OFFSET 13 bits**
  - Position des données du fragment dans le datagramme initiale
  - Valeur par défaut = Valeur Min = 0
  - valeur Max = 8191

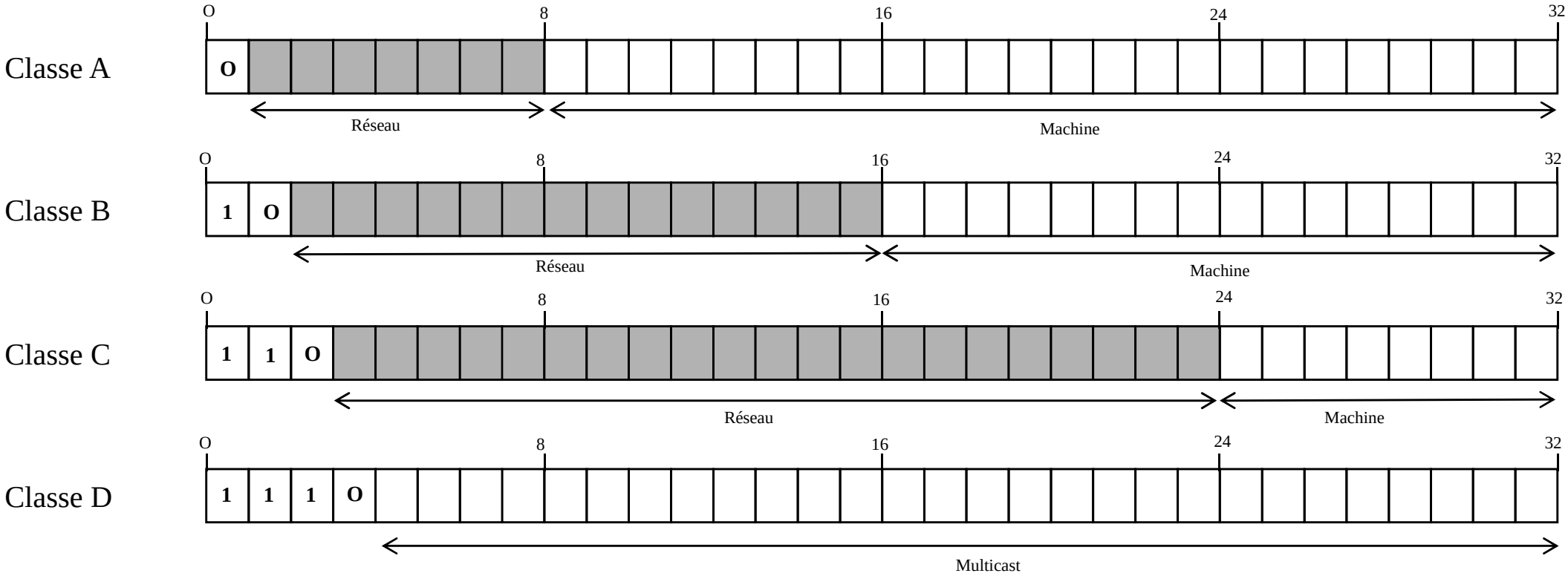
IP : format des datagrammes

- **TIME TO LIVE 8 bits**
  - Durée de vie du datagramme IP exprimée en secondes
  - TTL par défaut = 15 / TTL Min = 0 / TTL Max = 255
- **PROTOCOL 8 bits**
  - Identification du protocole de niveau supérieur (TCP=6 / UDP=17 / ICMP=1)
- **CHECKSUM 16 bits**
  - Vérification de l'intégrité du datagramme
  - Le complément à un sur 16 bits de la somme des compléments à un du datagramme IP
  - Le checksum est calculé à l ’émission (checksum = 0) et est vérifié à la réception
- **ADRESSE SOURCE 32 bits**
  - Adresse IP de la machine émettrice
- **ADRESSE DESTINATION 32 bits**
  - Adresse IP de la machine destinatrice
- **OPTIONS variable**
  - Options liées au protocole IP

IP : adressage

- **Une adresse IP est attribuée à toute interface physique connectée à un réseau IP**
- **On distingue 4 classes d'adresses IP : A,B, C et D**
- **Les adresses IP sont codées sur 32 bits**
- **Représentation des adresses en décimal**
  - xxx.xxx.xxx.xxx
  - 10001100 10001011 10000010 10011111 = 140.131.130.159
  - 0.0.0.0 à 255.255.255.255
- **Les adresses IP sont découpées en deux champs dont la taille est variable suivant la classe d'adresse: le champs réseau et le champ local**
- **Adressage public ou référencé**
  - Attribué au plan international par le IANA
- **Adressage privé**
  - inconnu des instances internationales

IP : adressage



IP : adressage

- **Réseau de classe A**
  - 7 bits pour la partie réseau soit 2^7 - 2 = 126 réseaux
  - 24 bits pour la partie machine soit 2^{24} - 2 = 16 777 214 machines
  - xxx.0.0.0/8
  - Exemple : 12.0.0.0/8

- **Réseau de classe B**
  - 14 bits pour la partie réseau soit 2^{14} - 2 = 16 382 réseaux
  - 16 bits pour la partie machine 2^{16} - 2 = 65 534 machines
  - xxx.xxx.0.0/16
  - Exemple : 128.196.0.0/16
- **Réseau de classe C**
  - 24 bits pour la partie réseau soit 2^{24} - 2 = 2 097 152 réseaux
  - 8 bits pour la partie machine soit 2^8 - 2 = 254 machines
  - xxx.xxx.xxx.0/32
  - Exemple : 197.242.123.0/24

IP : adressage

000.000.000.000	Réservé
000.xxx.xxx.xxx	Réservé
001.xxx.xxx.xxx - 126.xxx.xxx.xxx	Classe A
127.xxx.xxx.xxx	Loopback
128.000.xxx.xxx	Réservé
128.001.xxx.xxx - 191.254.xxx.xxx	Classe B
191.255.xxx.xxx	Réservé
192.000.000.xxx	Réservé
192.000.001.xxx - 223.255.254.xxx	Classe C
223.255.255.xxx	Réservé
224.xxx.xxx.xxx - 255.255.255.254	Classe D
255.255.255.255	Diffusion générale

IP : adressage

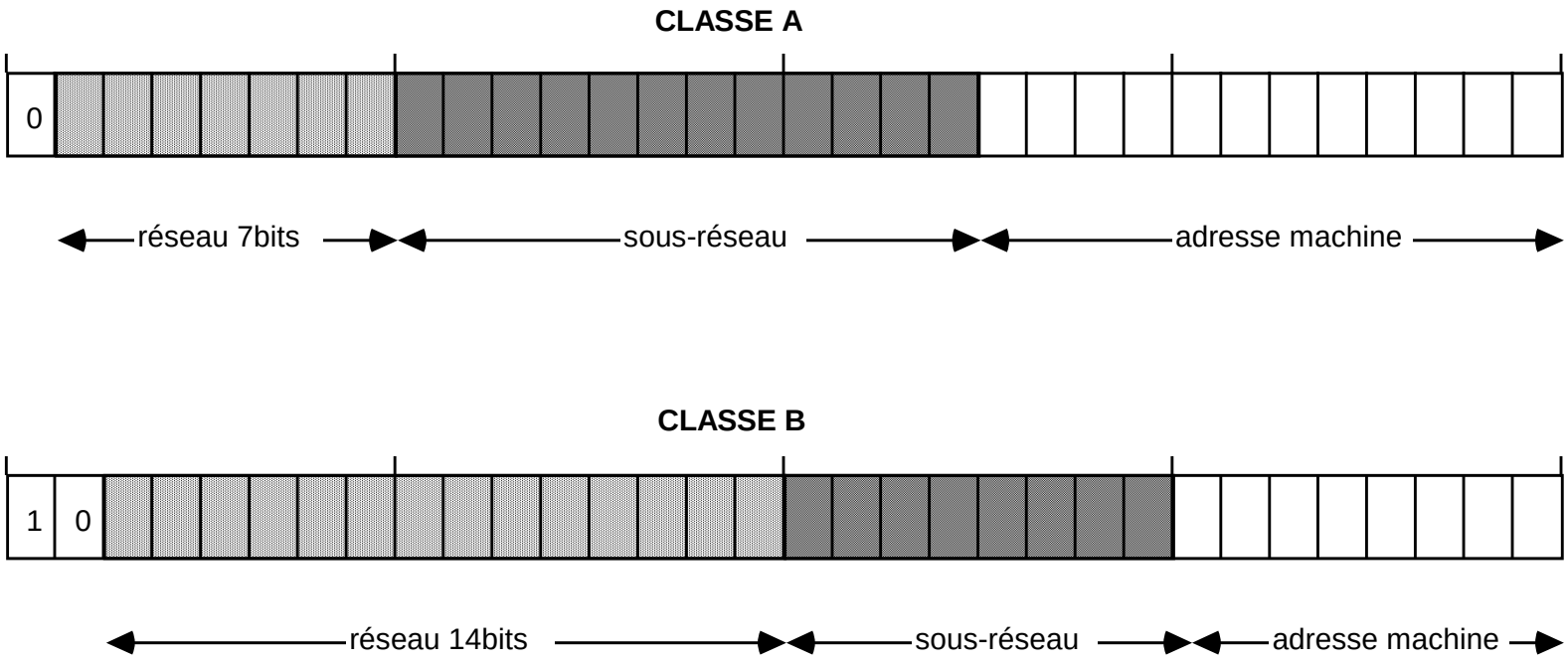
- **Le champ machine à 0 est utilisé au démarrage des machines pour connaître une adresse IP par le biais du protocole RARP**
  - Exemple : 195.10.200.0 ou 145.10.0.0
- **155.100.255.255 est l 'adresse de diffusion localisée au réseau de classe B 155.100.0.0**
- **Par convention de notation**
  - 0.0.0.1 sur le réseau 192.9.100 désigne la machine 192.9.100.1 du réseau de classe C 192.9.100.0
  - .1 désigne la machine 0.0.0.1 du réseau de classe C courant
  - .0.1 désigne la machine 0.0.0.1 du réseau de classe B courant
  - .0.0.1 désigne la machine 0.0.0.1 du réseau de classe A courant
  - 113.1.100.23 désigne l 'adresse 113.001.100.023
  - 113. désigne le réseau de classe A 113.0.0.0
  - 195.10.200. désigne le réseau de classe C 195.10.200.0

IP : sous-réseau

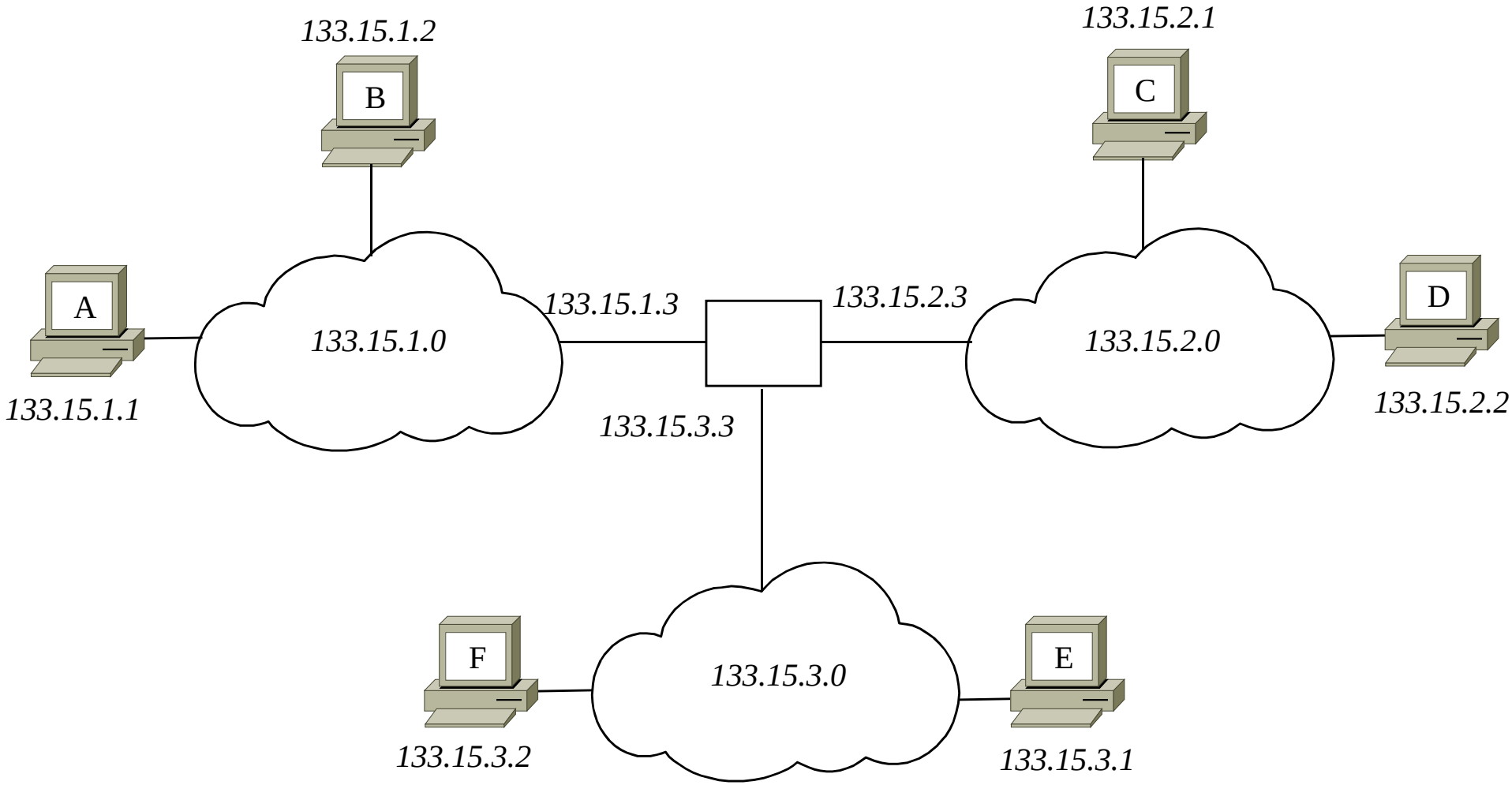
- **Sous-réseau IP**
  - RFC 915 « Internet subnets »
  - RFC 922 « Broadcasting Internet datagrams in the presence of subnets »
  - RFC 932 « A subnetwork addressing scheme »
  - RFC 936 « Another internet subnet addressing scheme »
  - RFC 950 « Internet standard subnetting procedure »
- **Le « subnetting » est possible avec les trois classes d ’adresses A, B et C**
- **La partie machine de l ’adresse IP est découpée en deux champs : sous-réseau et machine**
- **Le nombre de bits alloués au champ sous-réseau est choisi en fonction du nombre de sous-réseaux souhaités**
- **Masque de sous-réseau (subnet mask)**
  - Tous les bits des champs réseau et sous-réseau à 1
  - Tous les bits du champ machine à 0
- **Toute machine sur un sous-réseau doit connaître son masque de sous-réseau**

IP : sous-réseau

- **Exemples de subnetting**



IP : sous-réseau



	A	B	C	D	E	F
IP	133.15.1.1	133.15.1.2	133.15.2.1	133.15.2.2	133.15.3.1	133.15.3.2
SN	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
GW	133.15.1.3	133.15.1.3	133.15.2.3	133.15.2.3	133.15.3.3	133.15.3.3

IP : principe du routage

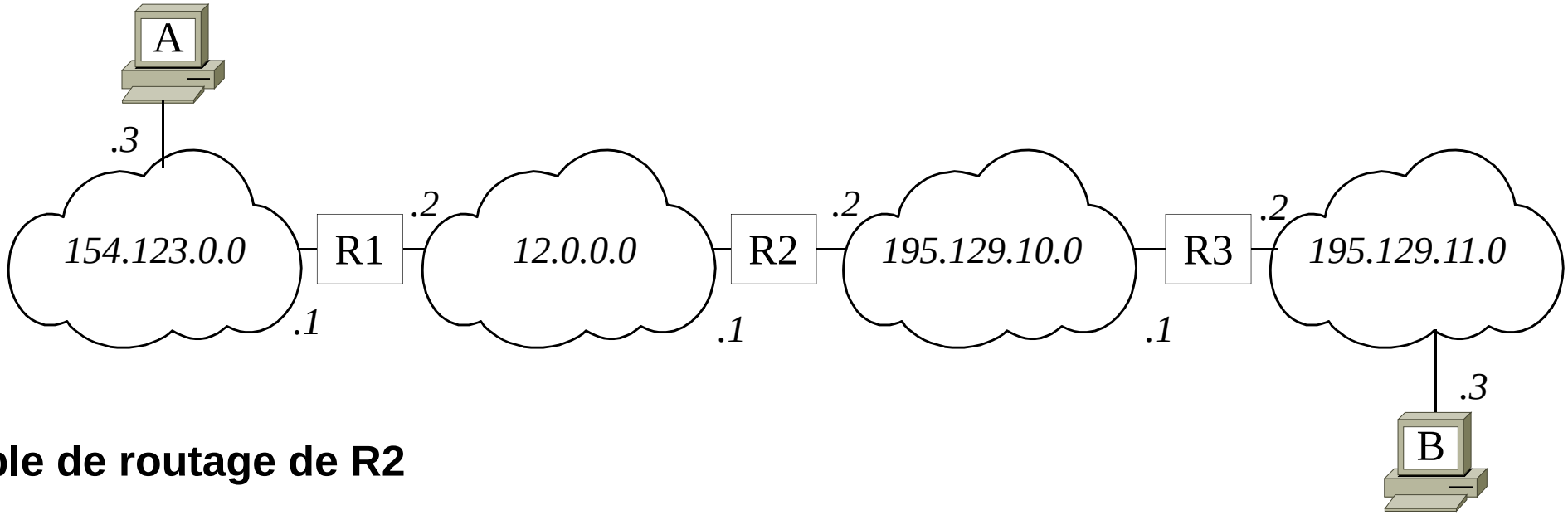


Table de routage de R2

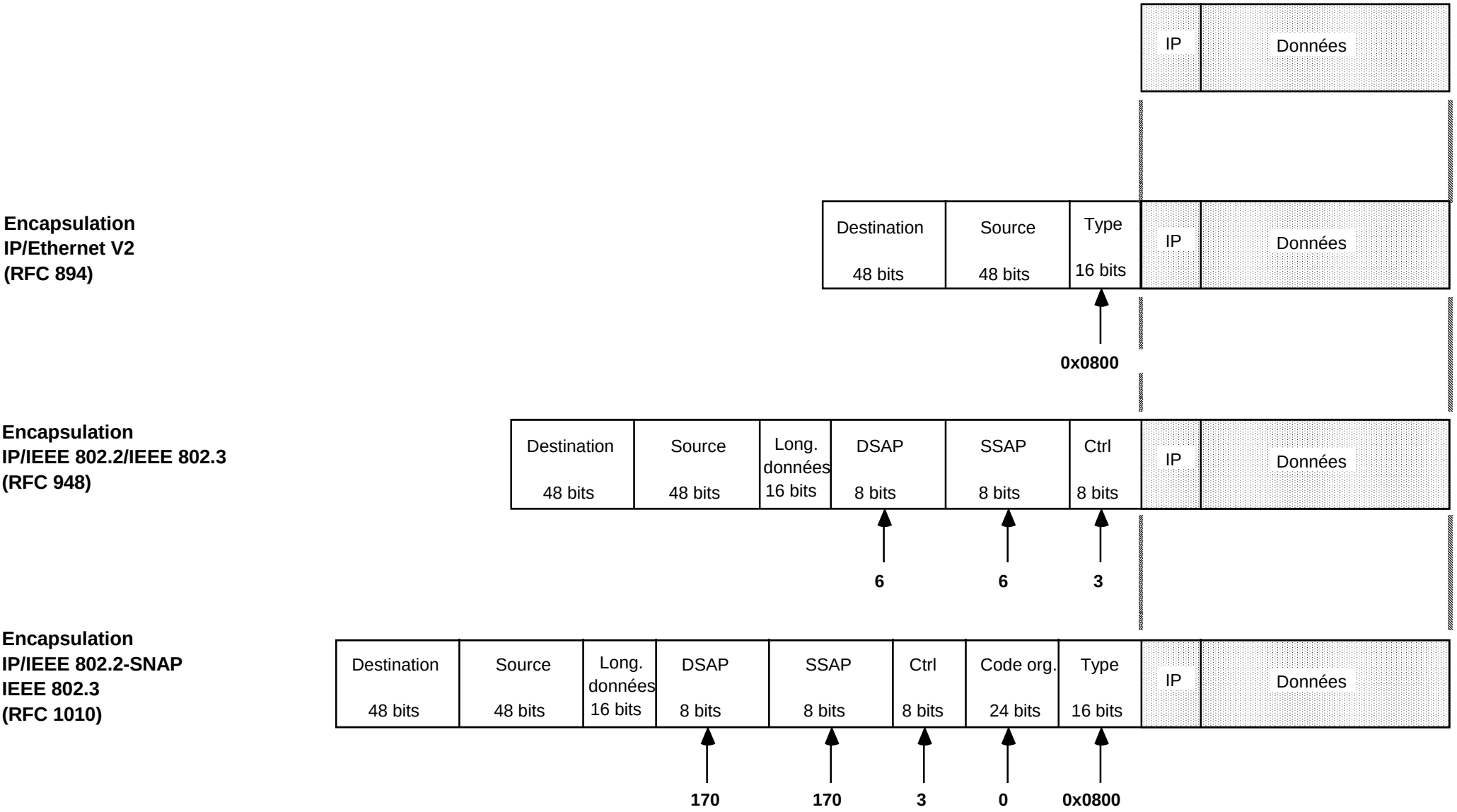
Réseau connu	Next Gateway
12.0.0.0	Direct
154.123.0.0	12.0.0.2
195.129.10.0	Direct
195.129.11.0	195.129.10.1

IP : exemple de plan d ’adressage

- **Hypothèse : adressage privé de classe A**

	Réseau	**sous-réseaux**			**local**
Taille	8 bits	12 bits			12 bits
Subnet Mask	255.	255.240.			0
Attribution	Société	Région	Site	Réseau	Machine
Taille	8 bits	4 bits	4 bits	4 bits	12 bits

IP : encapsulation Ethernet et 802.3



CIDR : généralités

- **Nombre d ’adresses IP disponibles en théorie : 2^{32} = 4 294 967 296**
- **Nombre d ’adresses IP avec l ’utilisation des classes d ’ adresses : 3 720 183 560**

  Classe A / 2 113 928 964

+ Classe B / 1 073 577 988

+ Classe C / 532 676 608

- **L ’utilisation des classes d ’adresses induit une perte d ’environ 14% d ’adresses**

CIDR : généralités

- **Classless Inter Domain Routing**
- **RFC 1519**
- **Objectifs**
  - pallier le manque d'adresse
  - limiter la taille des tables de routage
- **Allocation de réseaux sans classe**
- **Allocation de réseaux de classe C contigus**
- **Supernetting**
  - Agrégation des entrées dans les tables de routage
- **Notion de préfixe**
  - groupement par région, par opérateurs, etc …

CIDR : règles

- **Les “1” du masque doivent être contigus**
  - 254
  - 252
  - 248
  - ...
- **Deux réseaux IP**
  - 195.125.100.0 / 24 => Masque par défaut 255.255.255.0
  - 195.125.101.0 / 24 => Masque par défaut 255.255.255.0
  - 2 entrées dans les tables de routage
- **Agrégation des deux réseaux**
  - 195.125.100.0 / 23
  - Masque 255.255.254.0
  - 195.125.100.0 = préfixe
  - 1 entrée dans les tables de routage

CIDR : règles

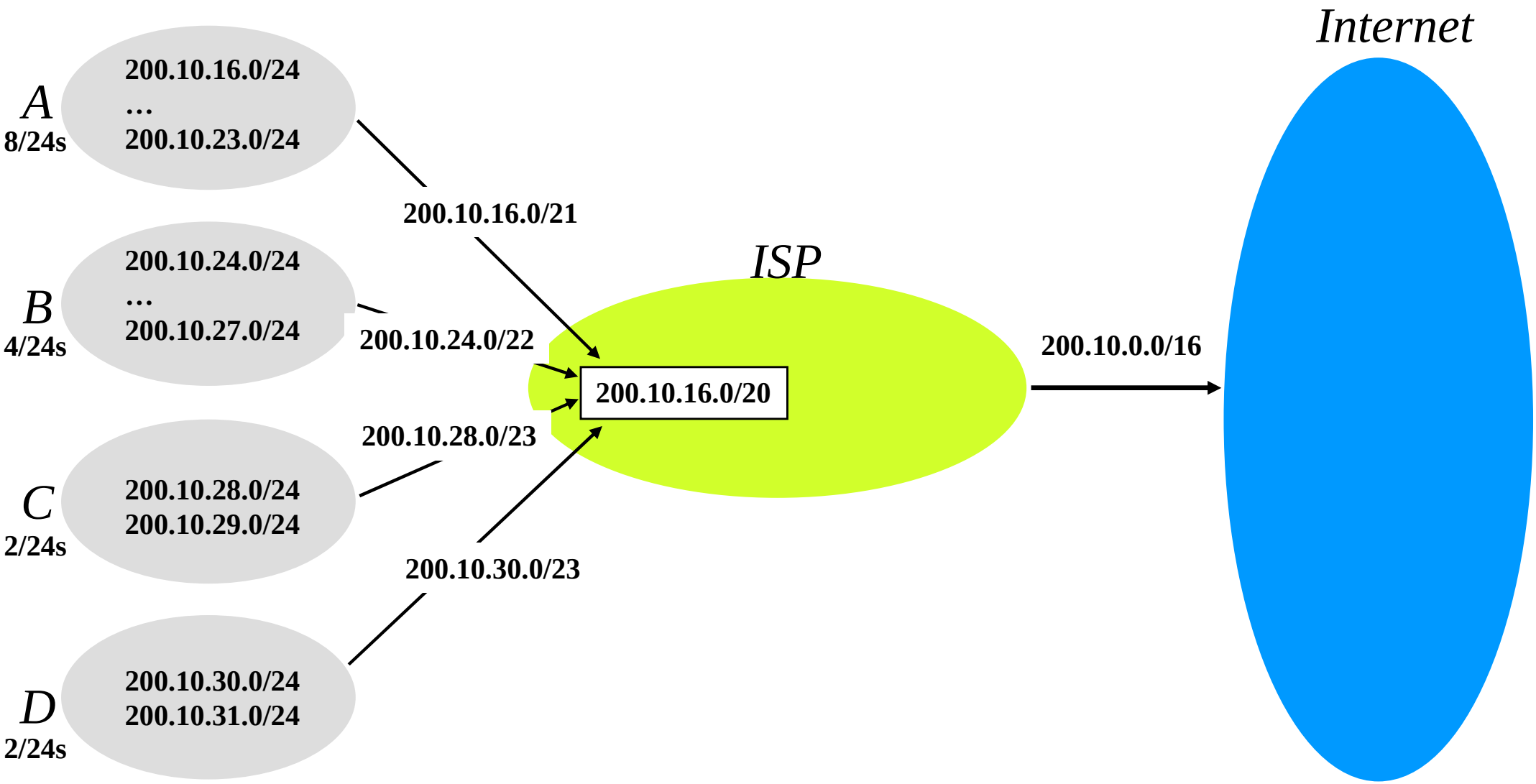
- **SuperNet**
  - /24 255 1
  - /23 254 2
  - /22 252 4
  - /21 248 8
  - /20 240 16
  - …… ...

CIDR : protocoles de routage

- **Protocoles de routages supportant CIDR**

EGP	Non
BGP4	Oui
RIPv1	Non
RIPv2	Oui
IGRP	Non
EIGRP	Oui
IS-IS	Oui
OSPFv2	Oui

CIDR : exemple



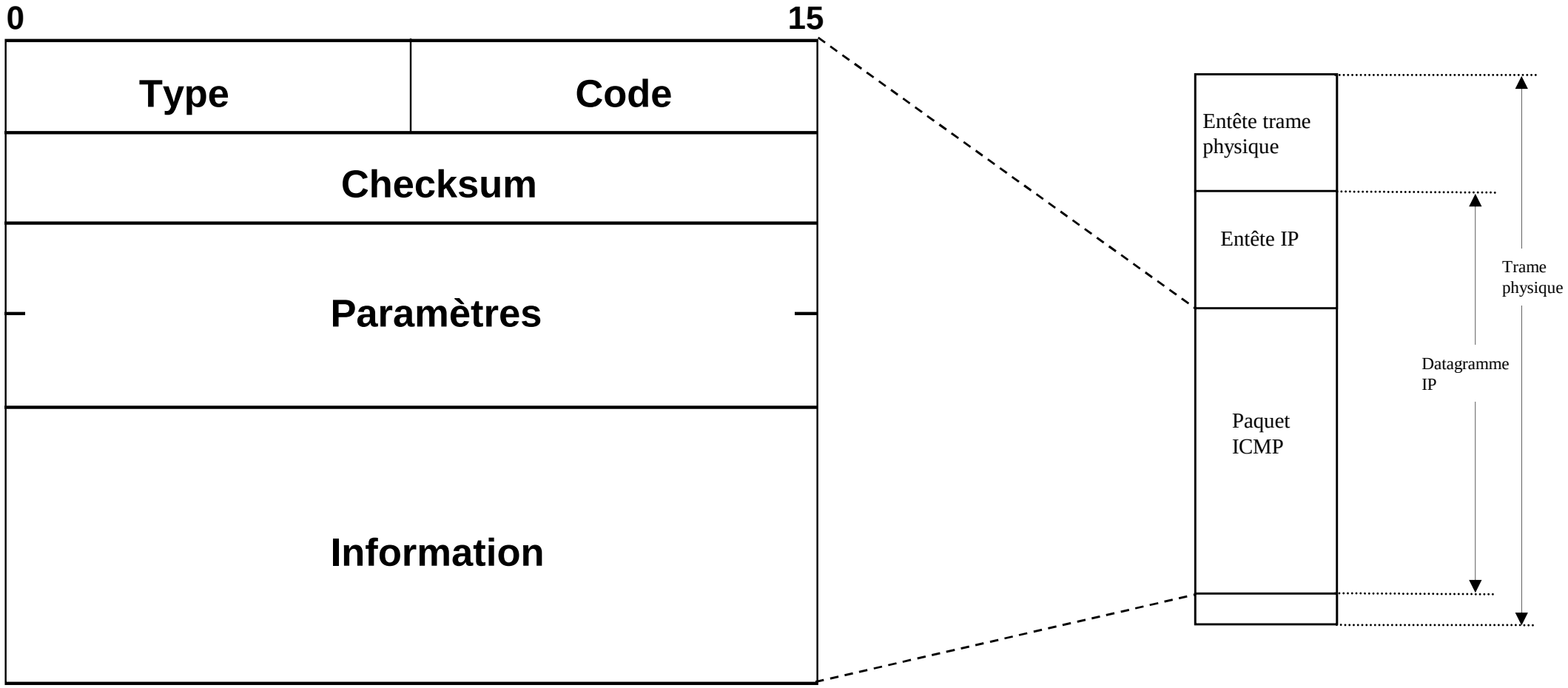
NAT

- **Network Address Translation**
- **Private IP network is an IP network that is not directly connected to the Internet**
- **IP addresses in a private network can be assigned arbitrarily or respecting RFC 1918**
  - Arbitrarily : not registered and not guaranteed to be globally unique
- **Range of private IP¨addresses (RFC 1918)**
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.255.255
  - 192.168.0.0 – 192.168.255.255

ICMP

- **Internet Control Message Protocol**
- **RFC 792**
- **ICMP est un protocole de contrôle et de report d'erreurs dans l'environnement IP**
- **ICMP fonctionne en mode non-connecté, il utilise les services du protocole IP**
  - TOS=0, Protocol=1
- **ICMP est obligatoire dans toutes les implémentations logicielles de TCP/IP**
- **ICMP ne traite pas les erreurs de paquets. .. ICMP**
- **Si un routeur détecte un problème sur un datagramme IP, il le détruit et émet un message ICMP pour informer l'émetteur sur la nature de l 'incident**
- **Le protocole ICMP intervient dans le routage IP**

ICMP : format des messages



ICMP : format des messages

- **TYPE 8 bits**
  - Indique la nature du message et le format du paquet ICMP
    - *0* *Echo Reply*
    - *3* *Destination Unreachable*
    - *4* *Source Quench*
    - *5* *Redirect*
    - *8* *Echo Request*
    - *11* *Time Exceeded for a Datagram*
    - *12* *Parameter Problem*
    - *13* *Timestamp Request*
    - *14* *Timestamp Reply*
    - *15* *Information Request*
    - *16* *Information Reply*
    - *17* *Adress Mask Request*
    - *18* *Adress Mask Reply*

ICMP : format des messages

- **CODE 8 bits**
  - Code d'erreur
    - *0* *Network Unreachable*
    - *1* *Host Unreachable*
    - *2* *Protocol Unreachable*
    - *3* *Port Unreachable*
    - *4* *Fragmentation Needed and DF set*
    - *5* *Source Route Failed*
    - *6* *Destination Network Unknown*
    - *7* *Destination Host Unknown*
    - *8* *Source Host Isolated*
    - *9* *Communication with desination network administratively prohibited*
    - *10* *Communication with desination host administratively prohibited*
    - *11* *Network Unreachable for type of Service*
    - *12* *Host Unreachable for type of Service*

ICMP : format des messages

- **CHECKSUM 16 bits**
  - Vérification de l'intégrité du paquet ICMP
  - Le complément à un sur 16 bits de la somme des compléments à un du message ICMP. Lors du calcul du Checksum, le champ destiné à recevoir ce Checksum sera laissé à zéro.
- **PARAMETRES**
  - Paramètres éventuels suivant le type de message
- **INFORMATION**
  - Informations relatives au message ICMP (par exemple une partie du paquet IP à l'origine du message d'erreur)

ICMP : inaccessibilité du destinataire

- **Type = 3**
- **Code**
  - 0 = réseau inaccessible
  - 1 = hôte inaccessible
  - 2 = protocole non disponible
  - 3 = port non accessible
  - 4 = fragmentation nécessaire mais interdite
  - 5 = échec d'acheminement source
- **Checksum**
  - Le complément à un sur 16 bits de la somme des compléments à un du message ICMP. Lors du calcul du Checksum, le champ destiné à recevoir ce Checksum sera laissé à zéro
- **Datagramme avec en-tête + 64 bits de données**
  - extrait du datagramme original pour renseigner le destinataire du message
- **Les codes 0, 1, 4, et 5 proviendront de routeurs.**
- **Les codes 2 et 3 proviendront d'hôtes.**

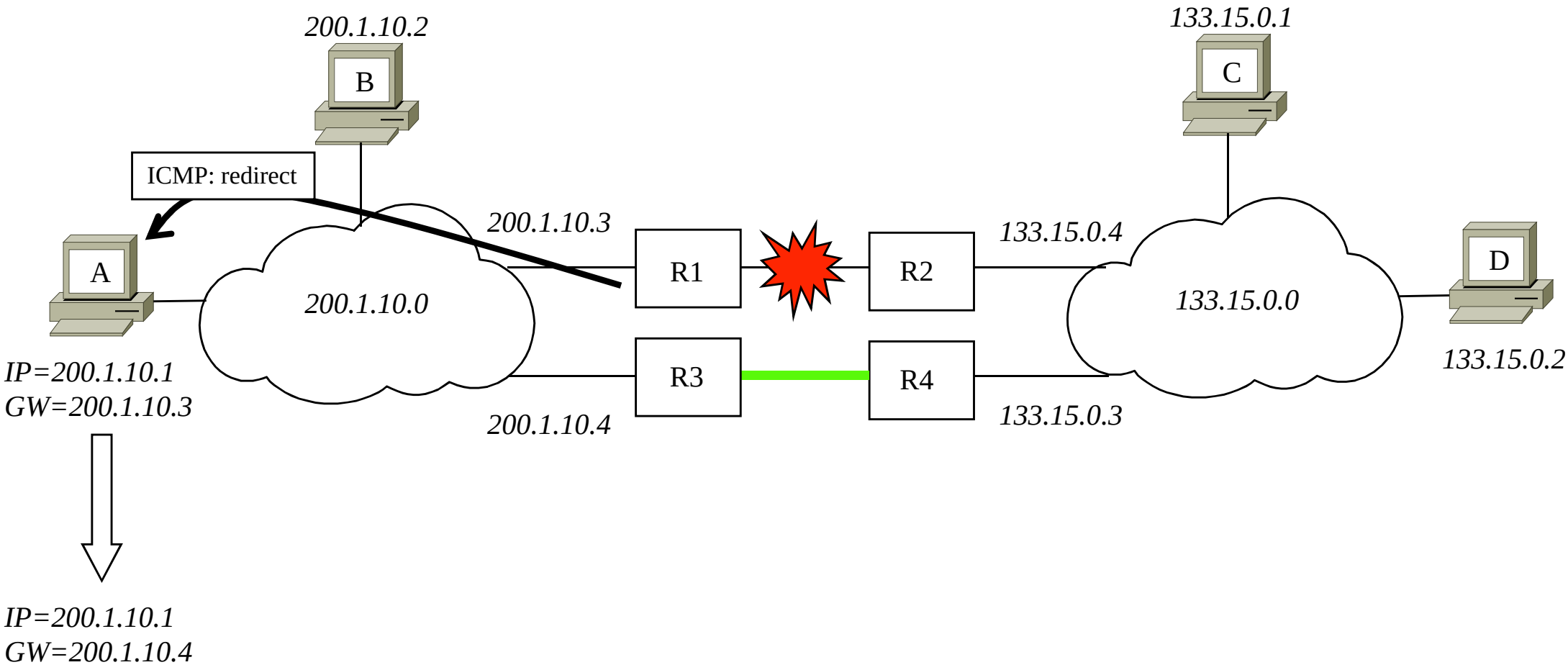
ICMP : gestion des congestions

- **Type = 4**
- **Code**
  - 0
- **Checksum**
  - Le complément à un sur 16 bits de la somme des compléments à un du message ICMP. Lors du calcul du Checksum, le champ destiné à recevoir ce Checksum sera laissé à zéro.
- **Datagramme avec en-tête + 64 bits de données**
  - extrait du datagramme original pour renseigner le destinataire du message
- **Les messages de code 0 proviendront d'un hôte ou d'un routeur.**

ICMP : redirection

- **Type = 5**
- **Code**
  - 0 = Redirection de datagramme sur la base du réseau
  - 1 = Redirection de datagramme sur la base de l'adresse d'hôte
  - 2 = Redirection de datagramme sur la base du réseau et du Type de Service
  - 3 = Redirection de datagramme sur la base de l'hôte et du Type de Service
- **Checksum**
  - Le complément à un sur 16 bits de la somme des compléments à un du message ICMP. Lors du calcul du Checksum, le champ destiné à recevoir ce Checksum sera laissé à zéro.
- **Adresse IP de routeur**
  - Adresse du routeur auquel le trafic, à destination du réseau spécifié dans le champ de destination de l'en-tête IP du datagramme original, doit être envoyé.
- **Datagramme avec en-tête + 64 bits de données**
  - extrait du datagramme original pour renseigner le destinataire du message
- **Les messages de codes 0, 1, 2, et 3 proviendront d'un routeur**

ICMP : redirection



ICMP : expiration de TTL

- **Type = 11**
- **Code**
  - 0 = durée de vie écoulée avant arrivée à destination;
  - 1 = temps limite de réassemblage du fragment dépassé.
- **Checksum**
  - Le complément à un sur 16 bits de la somme des compléments à un du message ICMP. Lors du calcul du Checksum, le champ destiné à recevoir ce Checksum sera laissé à zéro.
- **Datagramme avec en-tête + 64 bits de données**
  - extrait du datagramme original pour renseigner le destinataire du message
- **Un message de code 0 proviendra d'un routeur. Un message de code 1 proviendra d'un hôte.**

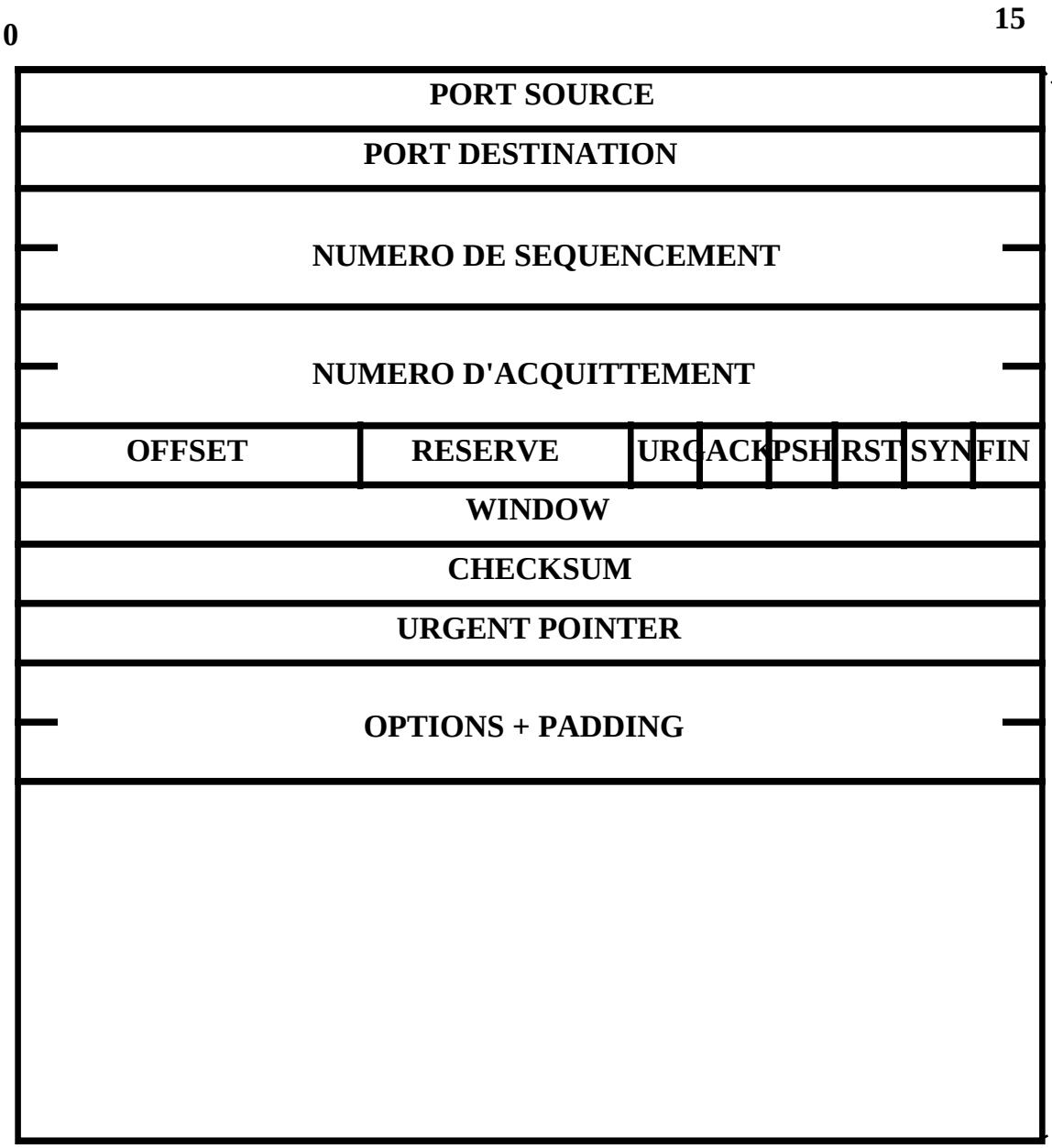
ICMP : message d ’erreur

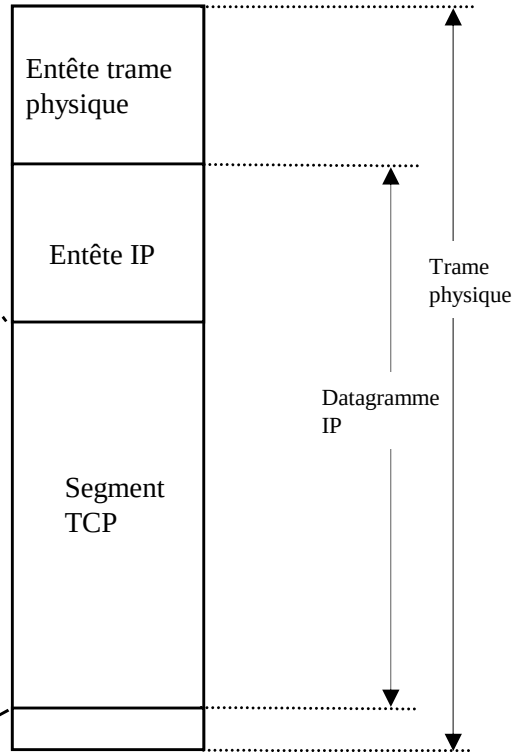
- **Type = 12**
- **Code**
  - 0 = l'erreur est indiquée par le pointeur.
- **Checksum**
  - Le complément à un sur 16 bits de la somme des compléments à un du message ICMP. Lors du calcul du Checksum, le champ destiné à recevoir ce Checksum sera laissé à zéro.
- **Pointer**
  - Si code = 0, identifie l'octet où l'erreur a été détectée.
- **Datagramme avec en-tête + 64 bits de données**
  - extrait du datagramme original pour renseigner le destinataire du message
- **Un message de code 0 pourra provenir d'un routeur ou d'un hôte.**

TCP

- **Transmission Control Protocol**
- **RFC 793 / MIL-STD-1778**
- **TCP est un protocole de transport orienté «flot d'octets»**
  - Les données transmises à TCP constituent un flot d'octets de longueur variable.
  - TCP divise ce flot de données en segments
- **TCP utilise les services de IP**
  - protocole = 6
- **Fiabilité de l'acheminement par séquencement, acquittement et retransmission des segments**
- **Le contrôle de flux est effectué par un système de fenêtres coulissantes**
- **TCP gère les congestions du réseau en utilisant un algorithme de retransmission adaptative**

TCP : format des segments





TCP : format de segments

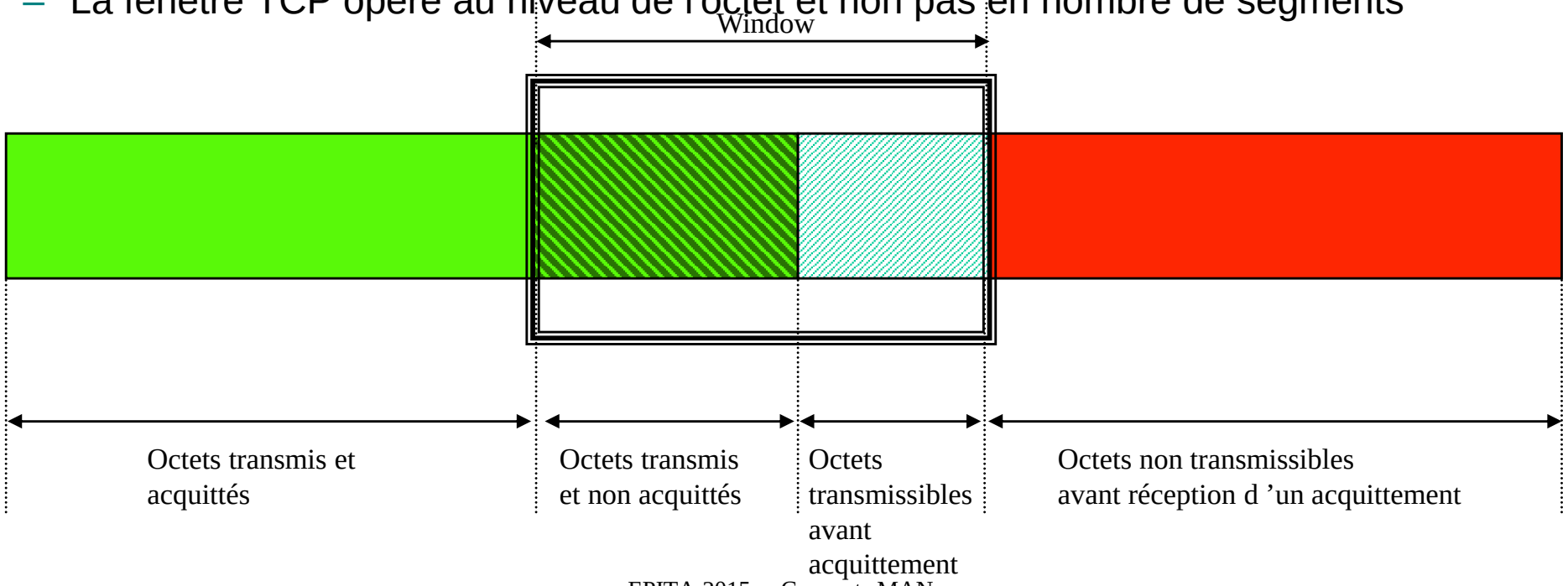
- **PORT SOURCE** **16 bits**
  - Port TCP de l'application émettrice du segment
- **PORT DESTINATAIRE** **16 bits**
  - Port TCP de l'application destinataire du segment
- **NUMERO DE SEQUENCE** **32 bits**
  - référence les octets transmis dans le segment
- **NUMERO D'ACQUITTEMENT** **32 bits**
  - le prochain numéro de séquencement attendu par l'émetteur de cet acquittement: tous les octets précédents cumulés sont implicitement acquittés
  - Si un segment a un numéro de séquencement supérieur au numéro de séquence attendu (bien que dans la fenêtre), le segment est conservé mais l'acquittement référence toujours le numéro de séquence attendu
  - Pour tout segment émis, TCP s'attend à recevoir un acquittement

TCP : format des segments

- **OFFSET**
  - Indique la position des données dans le segment à partir du début de l'entête. exprimé en nombre de mots de 32 bits
- **URG** **1 bit**
  - le segment contient des données urgentes à la position précisée dans le champ OFFSET
- **ACK** **1 bit**
  - Indique que le segment contient un acquittement
- **PSH** **1 bit**
  - Un segment TCP transmis avec le flag PSH indique au récepteur TCP qu'il doit remettre les données reçues à l 'application
- **RST** **1 bit**
  - réinitialisation de connexion
- **SYN** **1 bit**
  - indique une initialisation de connexion
- **FIN** **1 bit**
  - indique une libération de connexion

TCP : format des segments

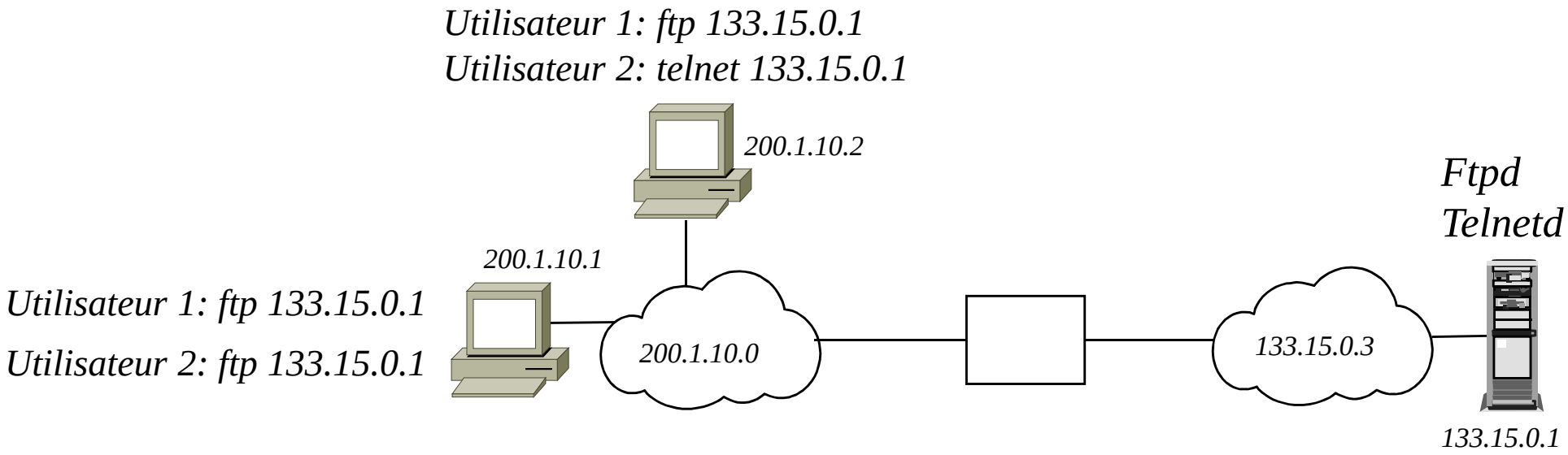
- **WINDOW** **16 bits**
  - Fenêtre « coulissante » dynamique
  - nombre d'octets pouvant être transmis par l ’émetteur du segment avant réception d ’un acquittement
  - contrôle du flux entre les deux extrémités de la connexion TCP
  - le destinataire des segments demande une régulation du débit de l ’émetteur en fonction des ressources dont il dispose
  - La fenêtre TCP opère au niveau de l'octet et non pas en nombre de segments



TCP : format des segments

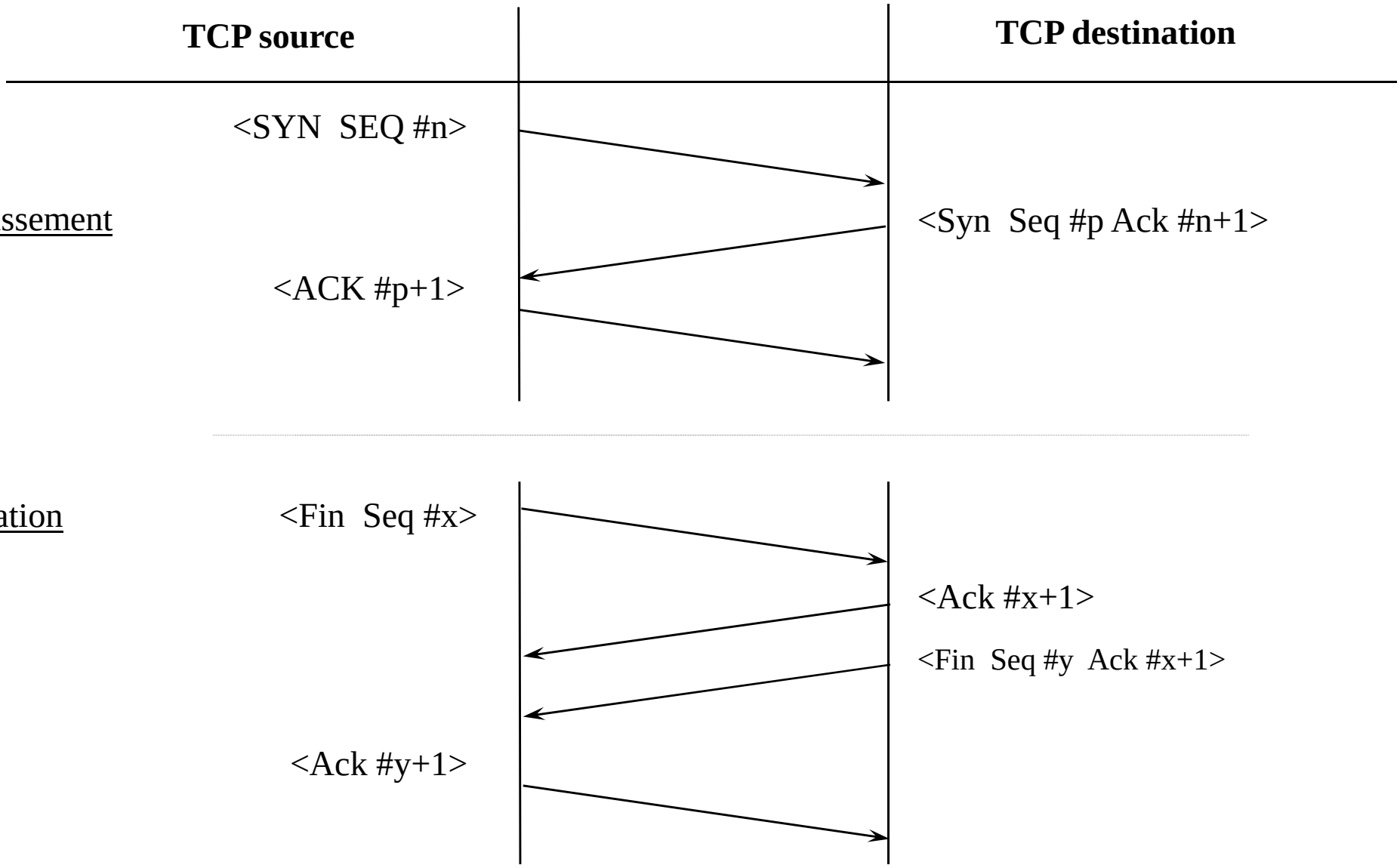
- **CHECKSUM** **16 bits**
  - Vérification de l'intégrité du segment TCP
  - Le calcul du CRC utilise un pseudo-header : @IP(source +destination) + protocole + longueur TCP
  - Le complément à un sur 16 bits de la somme des compléments à un du pseudo entête
- **POINTEUR URGENT** **16 bits**
  - Positionnement des données urgentes dans le segment (si URG = 1)
- **OPTIONS** **variable**
  - Permet un échange d'informations optionnelles entre les modules TCP distants
  - Permet de négocier la taille maximale des segments échangés. Cette option n'est présente que dans les segments d'initialisation de connexion (avec bit SYN).
  - TCP calcule une taille maximale de segment de manière à ce que le datagramme IP résultant corresponde au MTU du réseau. La recommandation est de 536 octets.

TCP : les connexions

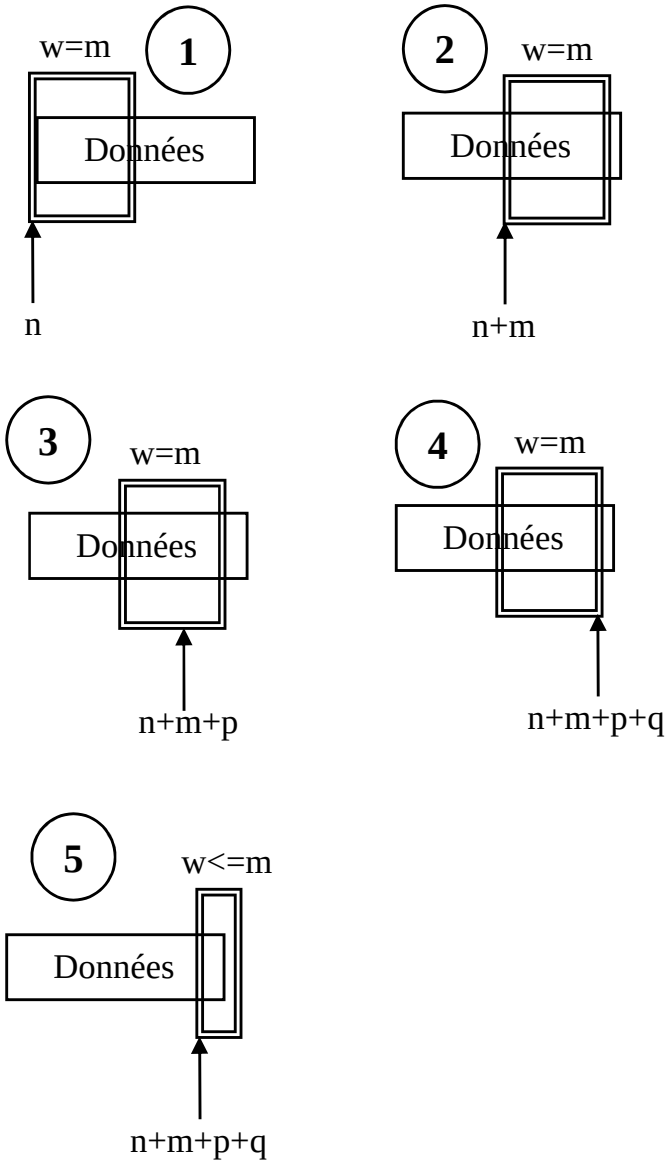


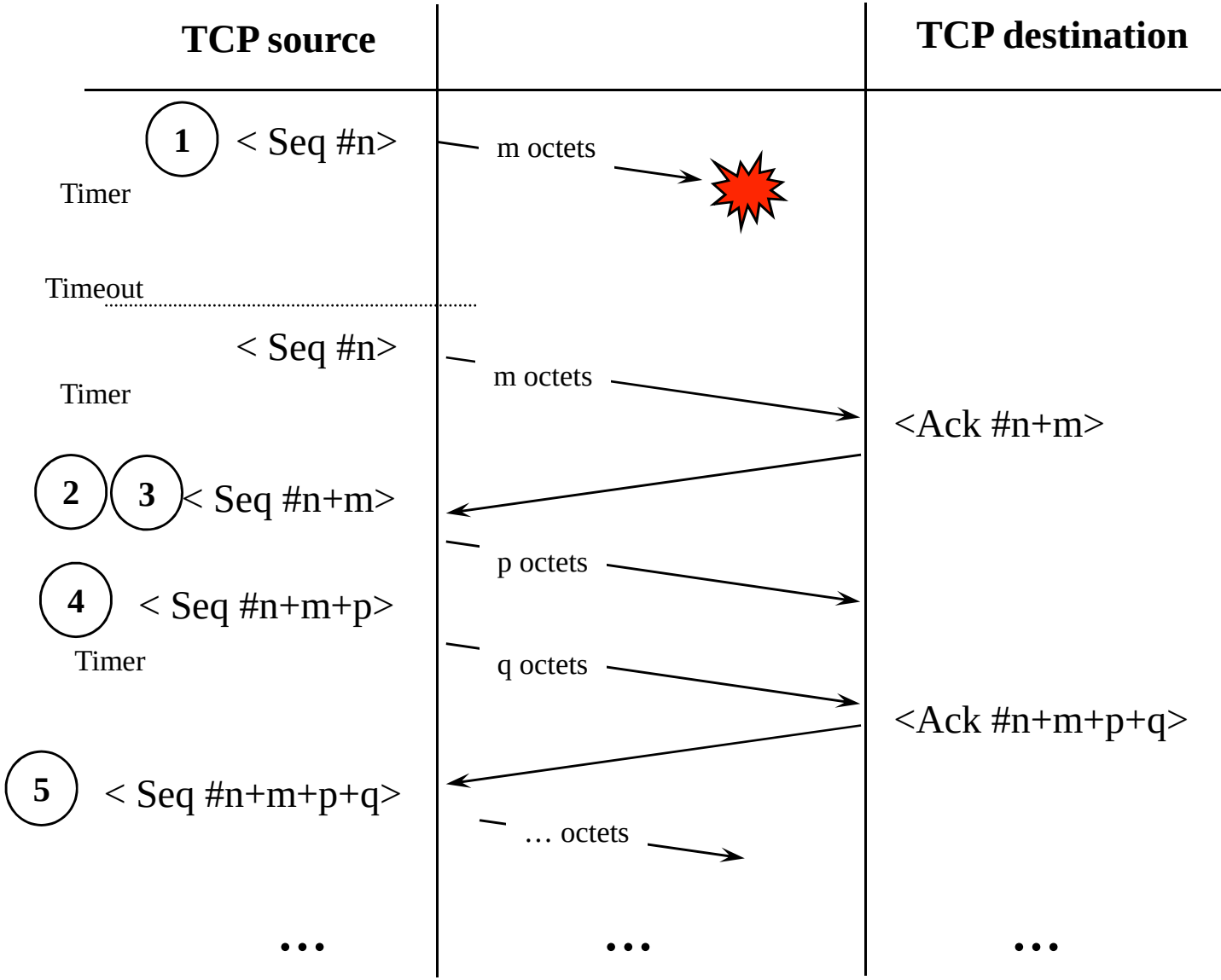
- **Une connexion TCP a deux extrémités !!**
- **Une extrémité de connexion TCP = couple (adresse IP, port TCP)**
- **Exemple**
  - Connexion n°1 = [(200.1.0.2, #n) , (133.15.0.1,21)]
  - Connexion n°2 = [(200.1.0.2, #q) , (133.15.0.1,23)]
  - Connexion n°3 = [(200.1.0.1, #p) , (133.15.0.1,21)]
  - Connexion n°4 = [(200.1.0.1, #n) , (133.15.0.1,21)]

TCP : établissement et libération de connexion



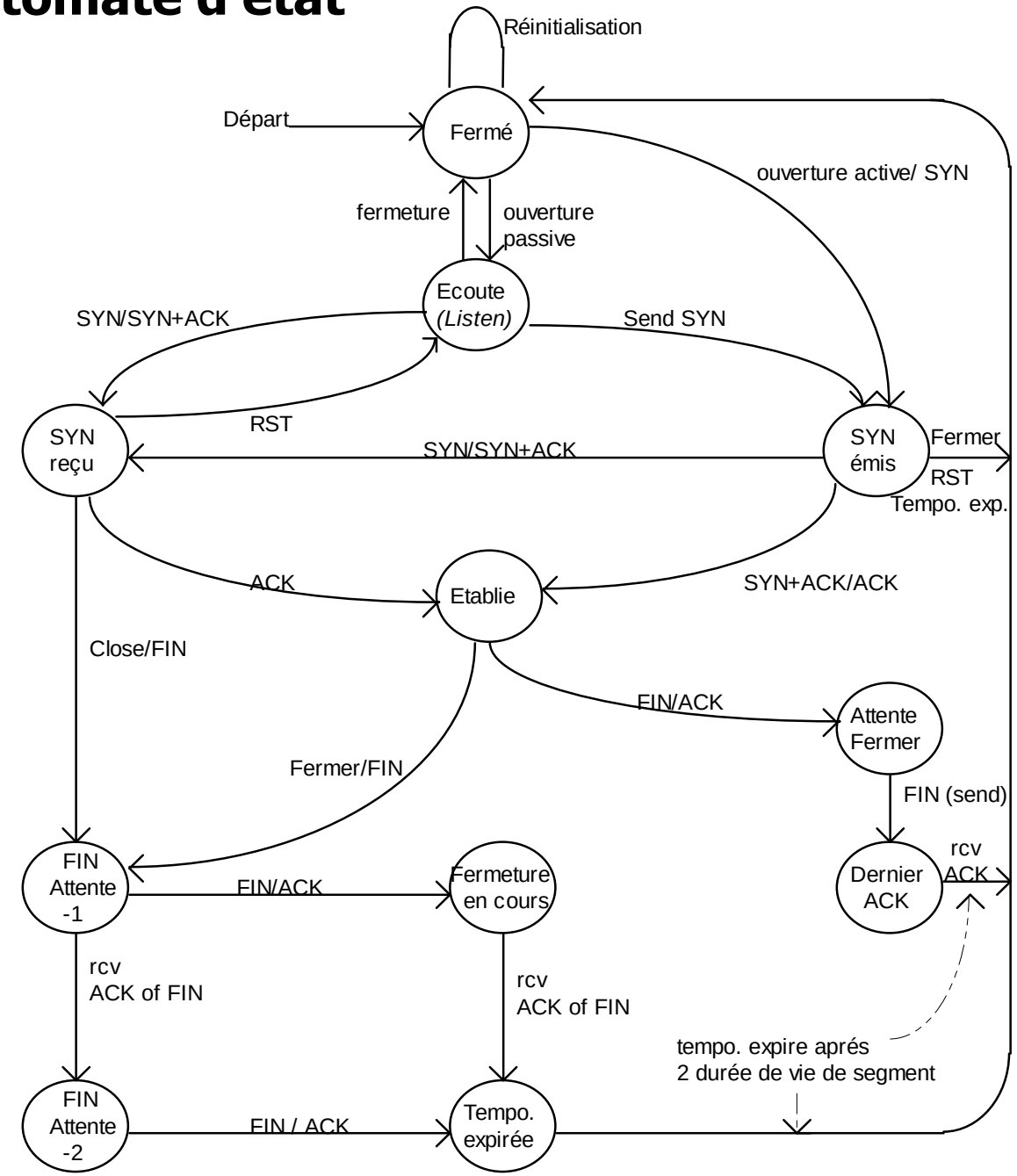
TCP : principe de fonctionnement





Hypothèses : window=m et p+q=m

TCP : automate d'état



TCP : changement d ’état

- **Une connexion TCP change d'état lorsque l'un des événements suivants survient**
- **APPELS DE L'APPLICATION**
  - CLOSE
  - OPEN
  - SEND
  - RECEIVE
  - ABORT
  - STATUS
- **ARRIVEE D'UN SEGMENT**
  - si l'un ou plusieurs des bits suivants est positionné : SYN, ACK, RST et FIN
- **TIMEOUT**
  - USER TIMEOUT
  - RETRANSMISSION TIMEOUT
  - TIME-WAIT TIMEOUT

TCP : le SRTT

- **Les délais de retransmission sont calculés dynamiquement dans TCP pour réagir aux perturbations du réseau. TCP base son mécanisme de retransmission sur le calcul du «Smooth Round-Trip Time».**
- **$SRTT_n$ = (alpha * $SRTT_{n-1}$) + ((alpha) * RTT)**
- **RTT: Temps écoulé entre l'envoi de données et la réception d'un acquittement.**
- **RTO: Retransmission Timeout**
- **RTO = min [UBOUND, max[LBOUND,(beta*SRTT)]]**
- **alpha: facteur d'atténuation entre 0.8 et 0.9**
  - alpha proche de 1 : RTT insensible aux variations brèves
  - alpha proche de 0 : RTT très sensible aux variations rapides
- **beta varie de 1.3 à 2.0**
- **UBOUND: limite sup. du timeout =1mn / LBOUND: lim. inf.=1s.**

TCP : gestion de la congestion

- **La congestion**
  - pertes de paquets
  - augmentation des délais d ’attente d ’acquittement au niveau de TCP
  - augmentation du nombre de retransmissions
  - Les extrémité ignorent tout de la congestion sauf les délais. Habituellement, les protocoles retransmettent les segments ce qui aggrave encore le phénomène.
- **Palliatifs**
  - ICMP source quench
  - fenêtre virtuelle de congestion au niveau TCP
- **TCP applique la fenêtre d’émission suivante**
  - Si absence de congestion: w_récepteur = w_congestion.
  - Si congestion: w_autorisée = min (w_récepteur, w_congestion).
  - Diminution dichotomique : à chaque segment perdu, la fenêtre de congestion est diminuée par 2 (minimum 1 segment)
  - la temporisation de retransmission est augmentée exponentiellement.

TCP : interface avec les applications

- **OPEN (Port local, port distant, mode d'ouverture, timeout, précédence, sécurité, options)**
  - Ouverture d ’une connexion TCP
- **SEND (identificateur de la connexion, adresse du buffer, nombre d'octets, flag PSH, flag URG, timeout**
  - Transmission de données
- **RECEIVE(identificateur de la connexion, adresse du buffer, nombre d'octets)**
  - Réception de données
- **CLOSE(identificateur de la connexion)**
  - Fermeture de connexion
- **STATUS (identificateur de la connexion):**
  - Donne des informations sur la connexion en cours.
- **ABORT (identificateur de la connexion)**
  - Arrêt d'une connexion.

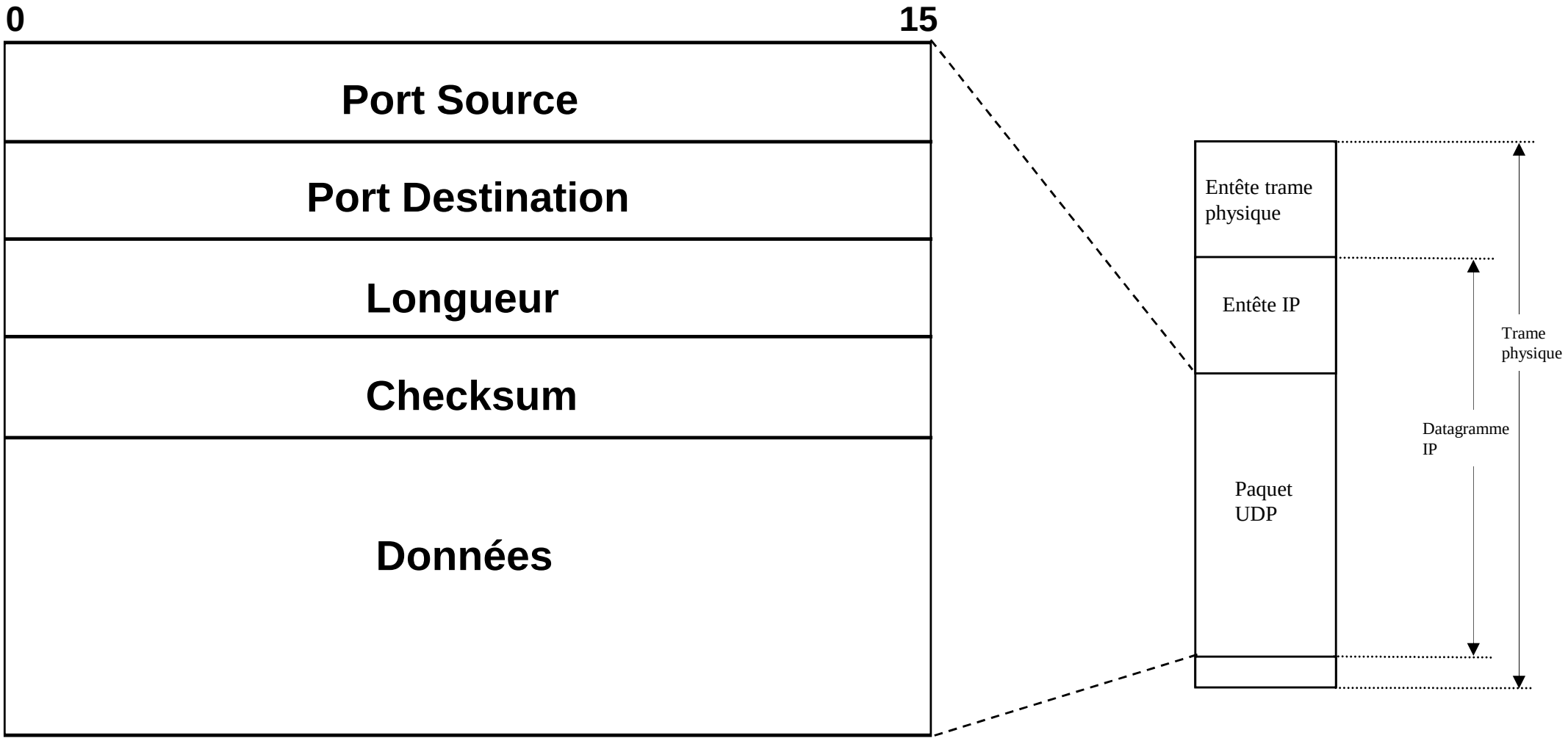
TCP : well-known ports

Port	Services	Description
20	FTP-DATA	File Transfer [Default Data]
21	FTP	File Transfer [Control]
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer
37		TIME Time
42	NAMESERVER	Host Name Server
43	NICNAME	Who Is
53	DOMAIN	Domain Name Server
79	FINGER	Finger
80		HTTP WWW
110	POP3	Post Office Protocol - Version 3
111	SUNRPC	SUN Remote Procedure Call

UDP

- User Datagram Protocol
- RFC 768
- UDP est un protocole de transport orienté datagramme
- Pas d ’acquittement et de séquencement
- Pas de contrôle de flux
- Service de multiplexage assuré (ports de communication)
- Utilisé par les applications NFS, TFTP, SNMP, ...

UDP : format des paquets



UDP : format des paquets

PORT SOURCE — **Port applicatif émetteur du datagramme**

PORT DESTINATION — **Port applicatif auquel est destiné le datagramme**

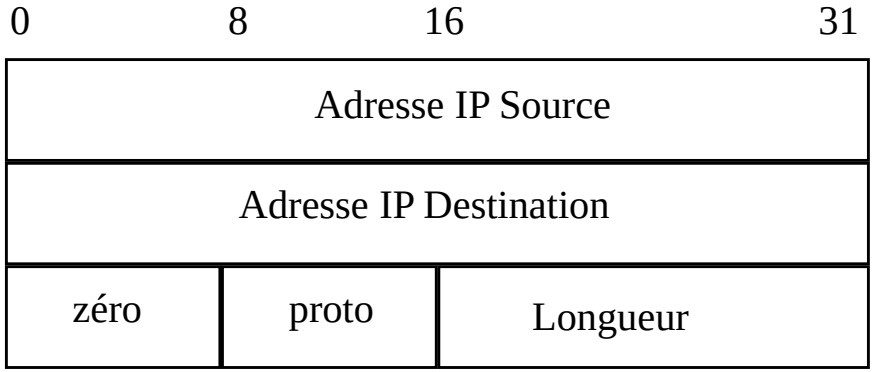
LONGUEUR — **Longueur totale (entête + données) en octets du datagramme UDP**

CHECKSUM — **Vérification de l'intégrité du datagramme UDP**

checksum

le calcul du Checksum est optionnel, la valeur 0 indique que le n'a pas été calculé

Le checksum est calculé sur un pseudo-entête UDP non transmis



PROTOCOL — Identificateur du protocole de transport (17= UDP)

LONGUEUR — Longueur du datagramme UDP sans le pseudo-en-tête.

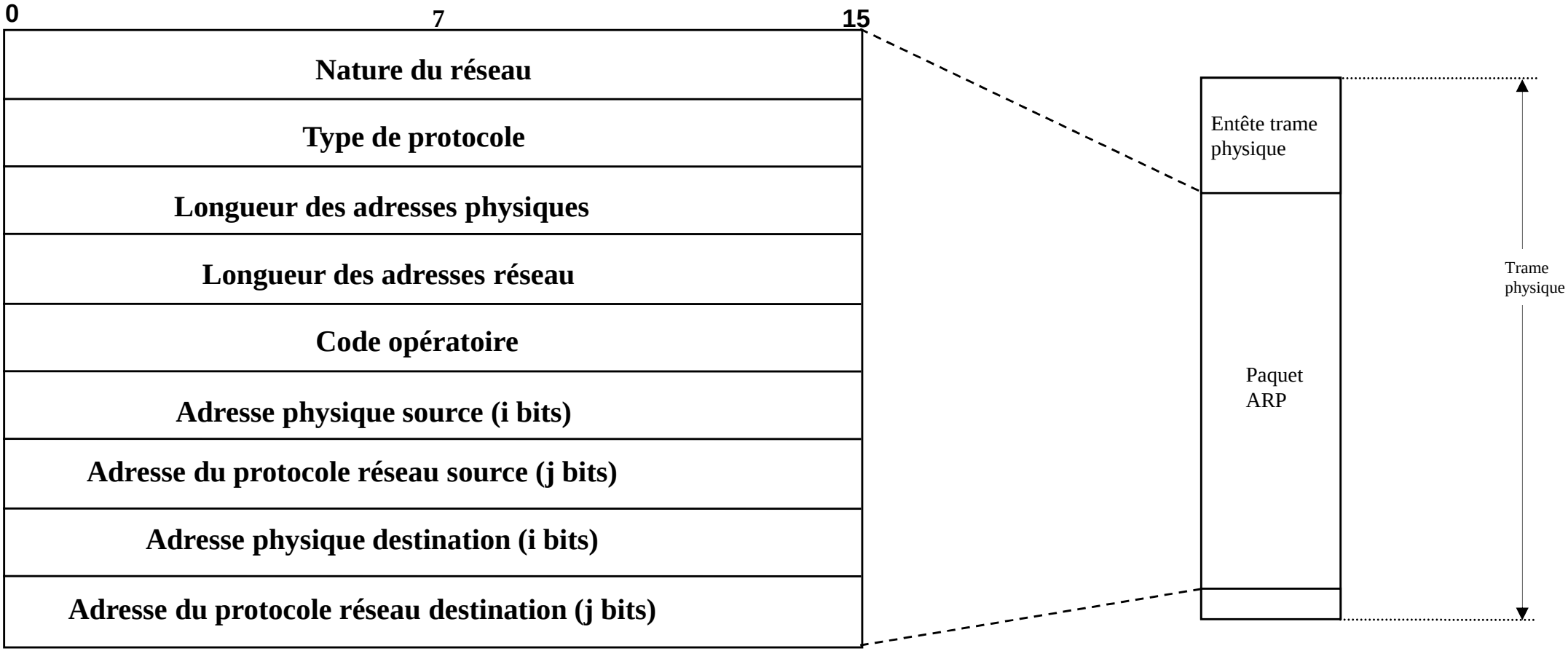
UDP : well known ports

Port	Services	Description
7	ECHO	Echo
11	USERS	Active Users
13	DAYTIME	Daytime
37	TIME	Time
42	NAMESERVER	Host Name Server
53	DOMAIN	Domain Name Server
67	BOOTPS	Boot protocol server
68	BOOTPC	Boot protocol client
69	TFTP	Trivial File transfert protocol
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management prot.

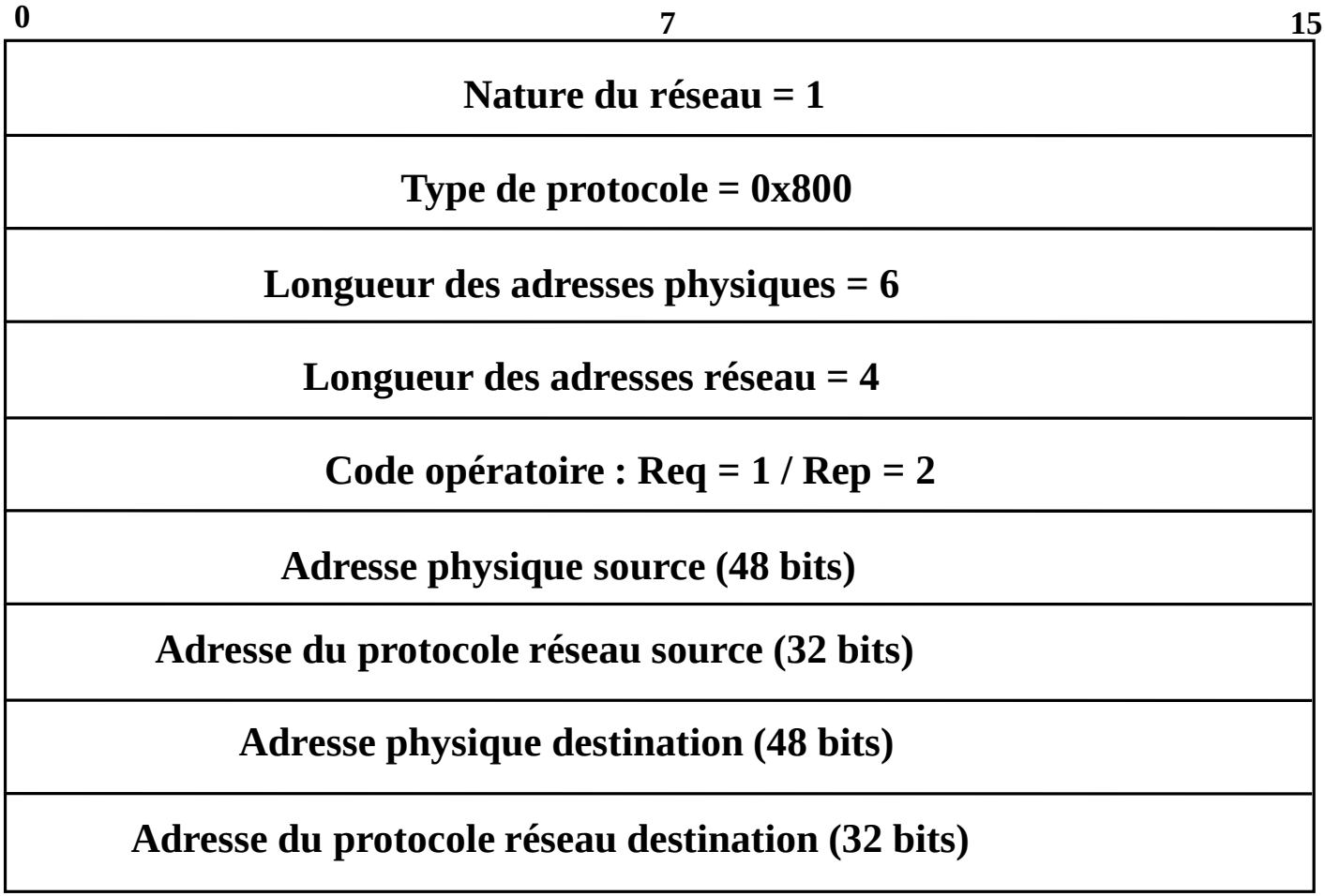
ARP

- **Address Resolution Protocol**
- **RFC 826**
- **Les paquets ARP (requêtes et réponses) sont directement encapsulés dans les trames de niveau MAC**
  - Exemple : type = 0x806 (ARP encapsulé dans Ethernet)
- **ARP permet de fournir à une machine l'adresse physique d'une autre machine située sur le même réseau physique à partir de l'adresse IP de la machine destinatrice**
  - Exemple : A souhaite dialoguer en IP avec B mais A ne connaît pas l 'adresse physique de B
  - A diffuse une requête ARP avec son adresse IP, son adresse physique et l 'adresse IP de B
  - Toutes les stations reçoivent la requête ARP
  - B répond avec son adresse physique
- **ARP ne peut opérer que sur des réseaux autorisant la diffusion**
- **Les paquets ARP ne « traversent » pas les routeurs**

ARP : format des paquets



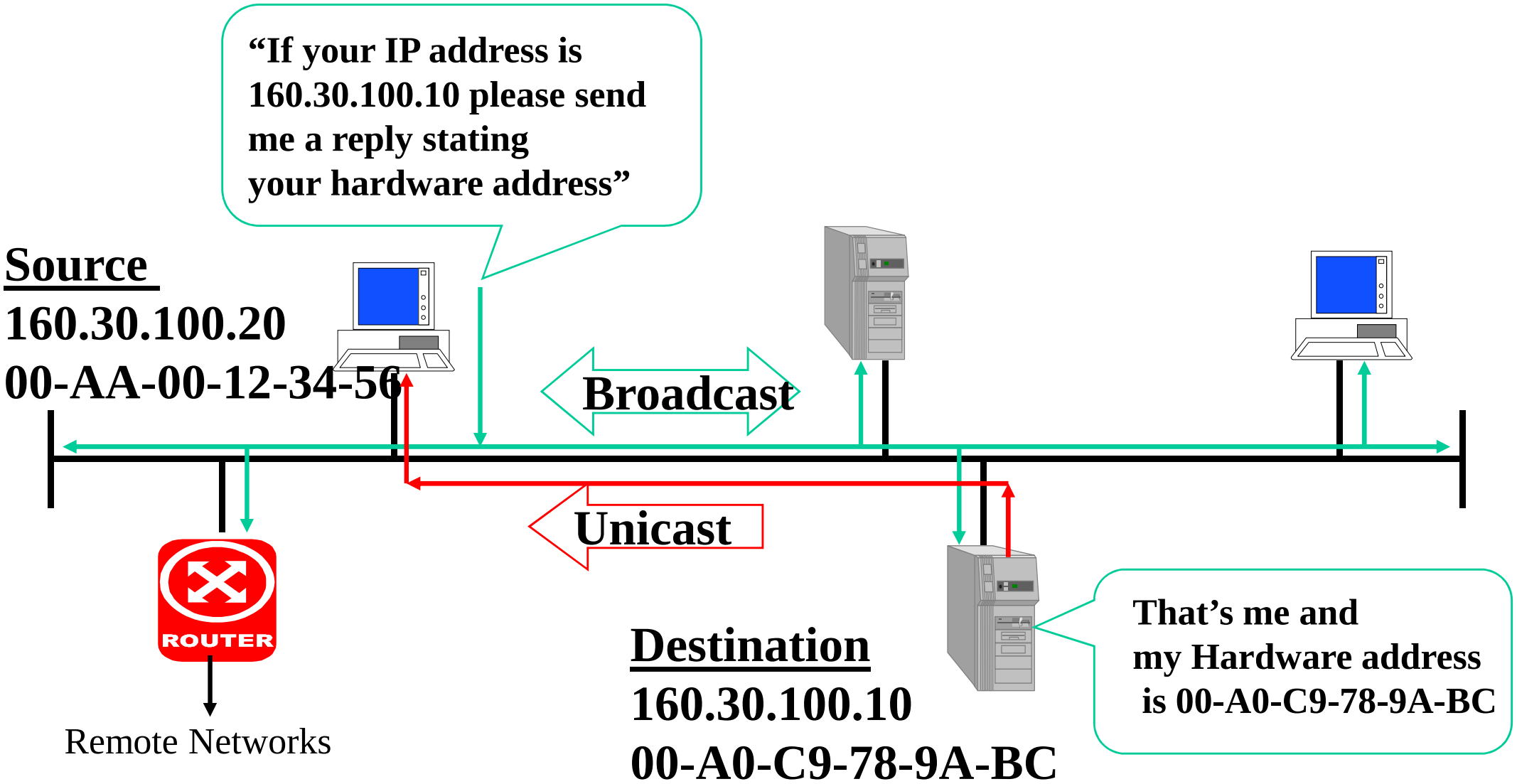
ARP : IP sur Ethernet



ARP : exemple de table

Protocole	Adresse	Adresse physique	Type
IP	131.122.1.2	02608C2EC381	Ethernet
IP	131.122.1.3	08002007F0FA	Ethernet
IP	131.123.1.1	02608C2ECC38	Ethernet
IP	131.122.1.2	550020003C5E	Token-Ring
IP	131.122.1.3	10004566FA12	Token-Ring
IP	28.2.10.10	02608C2EC380	Ethernet
IP	28.2.10.11	02608CFAFE12	Ethernet
IP	28.2.10.11	55002000E4E5	Token-Ring

ARP


“If your IP address is 160.30.100.10 please send me a reply stating your hardware address”
Source
160.30.100.20
00-AA-00-12-34-56
Broadcast
Unicast
ROUTER
Remote Networks
Destination
160.30.100.10
00-A0-C9-78-9A-BC
That’s me and my Hardware address is 00-A0-C9-78-9A-BC

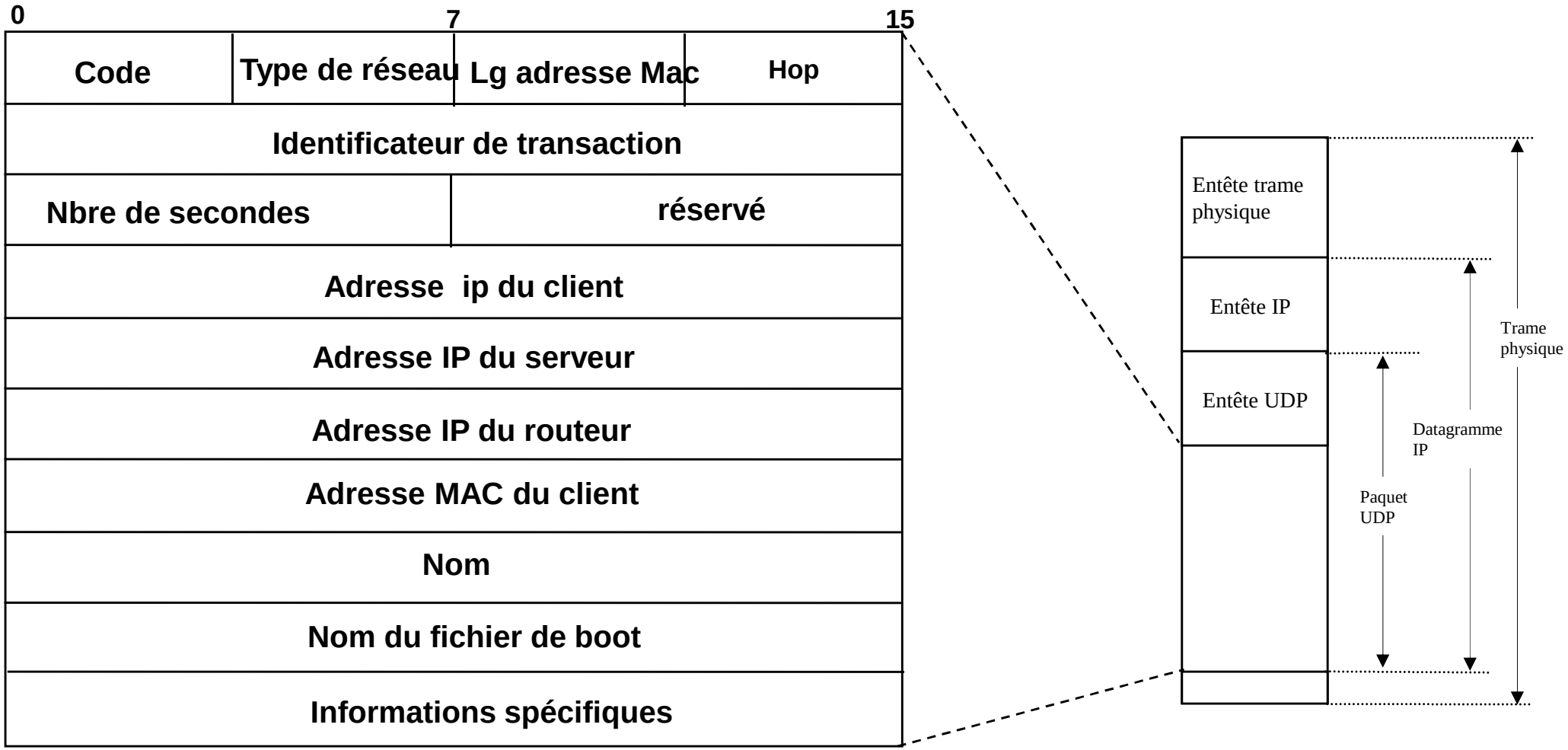
RARP

- **Reverse Address Resolution Protocol**
- **RFC 903**
- **Utilisé pour permettre à une station ne connaissant que son adresse physique de prendre connaissance de son adresse IP de manière dynamique.**
- **Le format des paquets RARP est le même que celui des paquets ARP (type = 0x8035)**
- **Un serveur primaire est affecté à chaque machine qui effectue une demande RARP, ainsi que des serveurs secondaires en cas d'indisponibilité du serveur primaire.**
- **Le serveur RARP maintient à jour une base d ’informations entre les adresses IP et les adresses physiques.**
- **les stations émettent une requête RARP en diffusion pour demander l'adresse IP qui est associée à leur adresse physique.**
- **Les serveurs RARP envoient à l’émetteur une réponse RARP contenant son adresse IP.**
- **Les paquets RARP ne « traversent » pas les routeurs.**

BOOTP

- **Bootsrap Protocol**
- **RFC 951**
- **Protocole de démarrage (terminaux X, stations sans disque, …)**
- **Analogue à RARP mais de plus haut niveau et plus riche en terme d ’informations envoyées**
- **Utilise les services de UDP**
  - Serveur : port 67 / Client : port 68
- **Serveur Bootp**
  - *ba : diffusion de réponses bootp pour tester les requêtes bootp*
  - *bf : localisation du fichier de boot*
  - *ds : adresse IP du serveur DNS*
  - *gw : adresse du proche routeur*
  - *ha : adresse MAC*
  - *hd : « home directory » du fichier de boot*
  - *hn : nom*
  - *ht : nature du réseau physique*
  - *ip : adresse IP*
  - *sm : masque de sous-réseau*
  - *...*

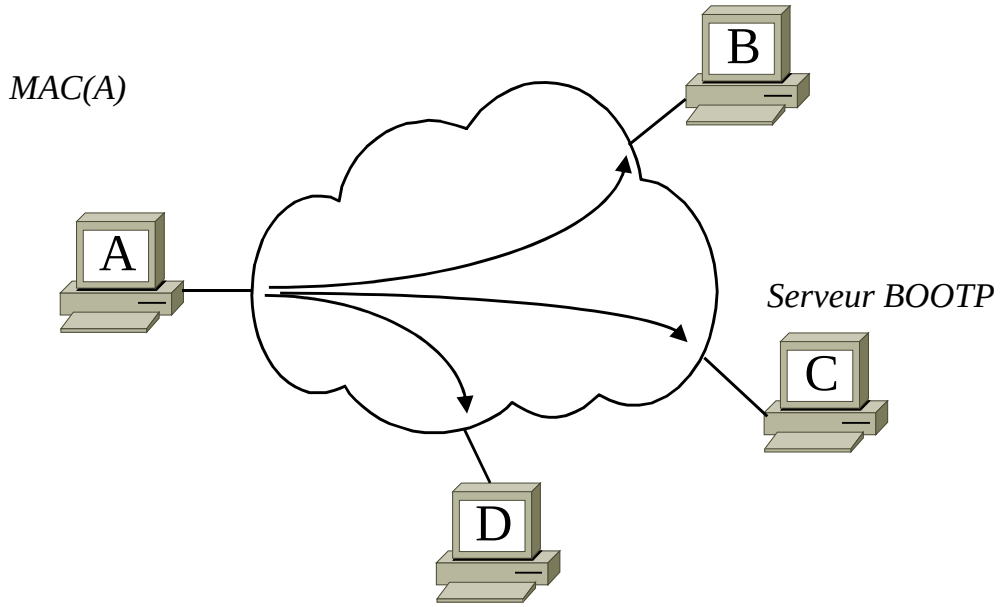
BOOTP : format des paquets



BOOTP : format des paquets

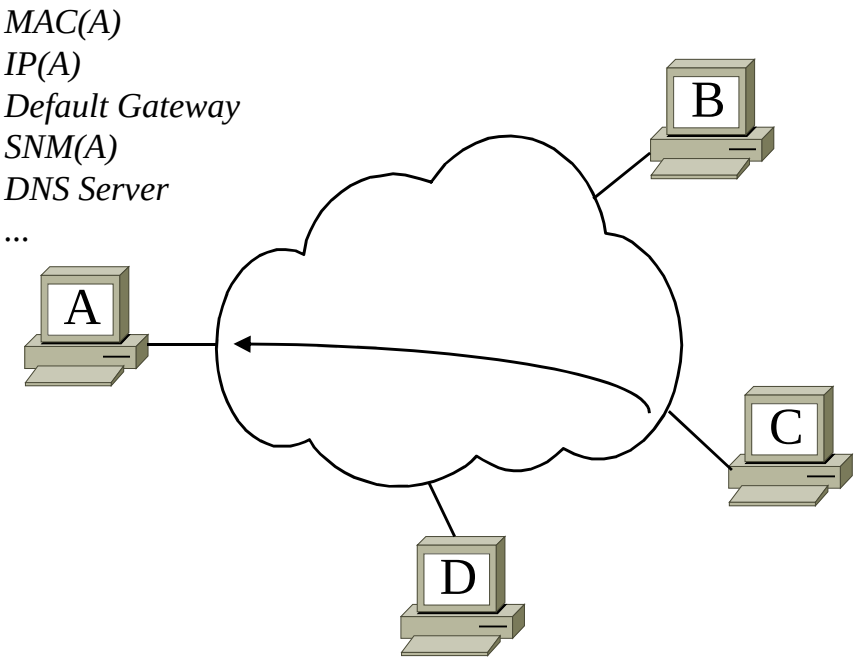
- **CODE**
  - 1 = requête / 2 = réponse
- **TYPE**
  - 1 = Ethernet
- **longueur de l'adresse MAC**
  - 6 = Ethernet
- **HOP COUNT**
  - 0 par défaut / Si le paquet transite par un routeur celui-ci l'incrémente de 1.
- **IDENTIFICATEUR DE TRANSACTION**
  - entier de 32 bits fixé aléatoirement qui sert à faire correspondre les réponses avec les requêtes.
- **NOMBRE DE SECONDES**
  - fixé par le client et sert à un serveur secondaire de délai d'attente avant qu'il ne réponde au cas où le serveur primaire serait en panne.

BOOTP : principe



Diffusion de la requête (255.255.255.255)

Réponse transmise en diffusion ou directement à l ’émetteur de la requête

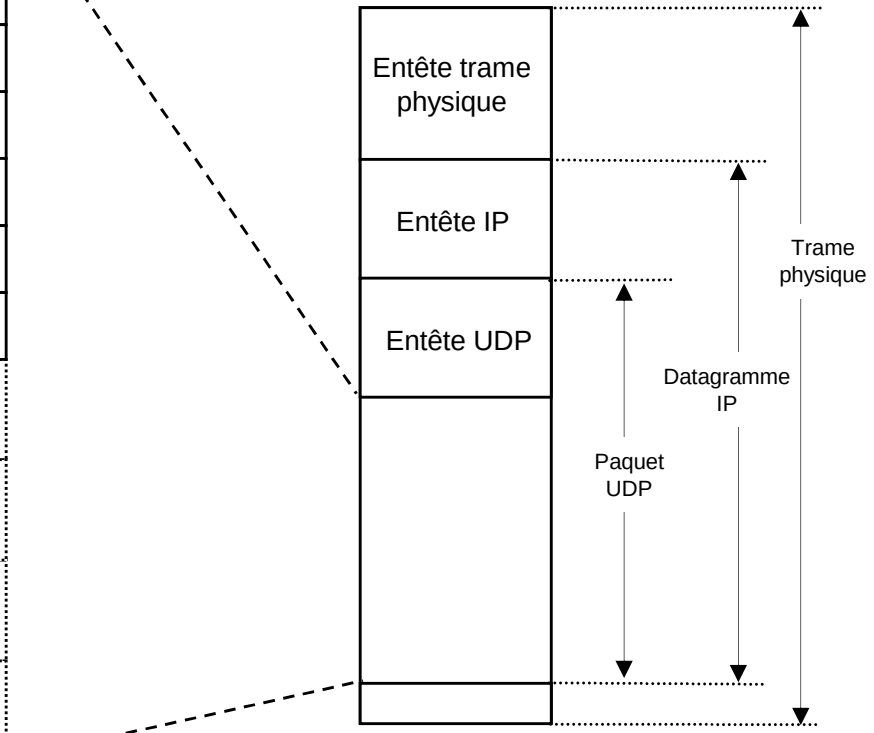


DHCP : principe général

- **Dynamic Host Configuration Protocol**
- **RFC 2131/2132**
- **Bootp ne peut pas affecter d 'adresses dynamiquement (ISP, portables, ….)**
  - DHCP = BOOTP ++
- **Mécanisme**
  - plug & play
  - d 'attribution dynamique d 'adresses IP
  - d 'attribution dynamique de paramètre de configuration des machines
    - *hostname, IP address, netmask, routers, name servers, time servers, log servers, boot file, boot server, vendor-specific information*
- **UDP port 67/68**
- **Machines qui ne doivent pas utiliser DHCP**
  - Routeurs
  - Serveurs DHCP
  - Certains serveurs applicatifs
- **Contrat entre le client et le serveur**
  - durée variable

DHCP : format des paquets

0	8	16	24	31
OP	HTYPE	HLEN	HOPS	
TRANSACTION ID				
SECONDS		**FLAGS**		
CLIENT IP ADDRESS				
YOUR IP ADDRESS				
SERVER IP ADDRESS				
ROUTER IP ADDRESS				
CLIENT HARDWARE ADDRESS (16 OCTETS)				
SERVER HOST NAME (64 OCTETS)				
BOOT FILE NAME (128 OCTETS0				
OPTIONS (VARIABLE)				



DHCP : format des paquets

- **OP CODE 8 bits**
  - 1 DHCP discover
  - 2 DHCP Offer
  - 3 DHCP Request
  - 4 DHCP Decline
  - 5 DHCP Ack
  - 6 DHCP Nack
  - 7 DHCP Release
- **HWTYPE 8 bits**
  - Nature du réseau
  - 1=Ethernet … 6=802.2 …
- **HLENGTH 8 bits**
  - Longueur en octets des adresses physique
  - 6=Ethernet …

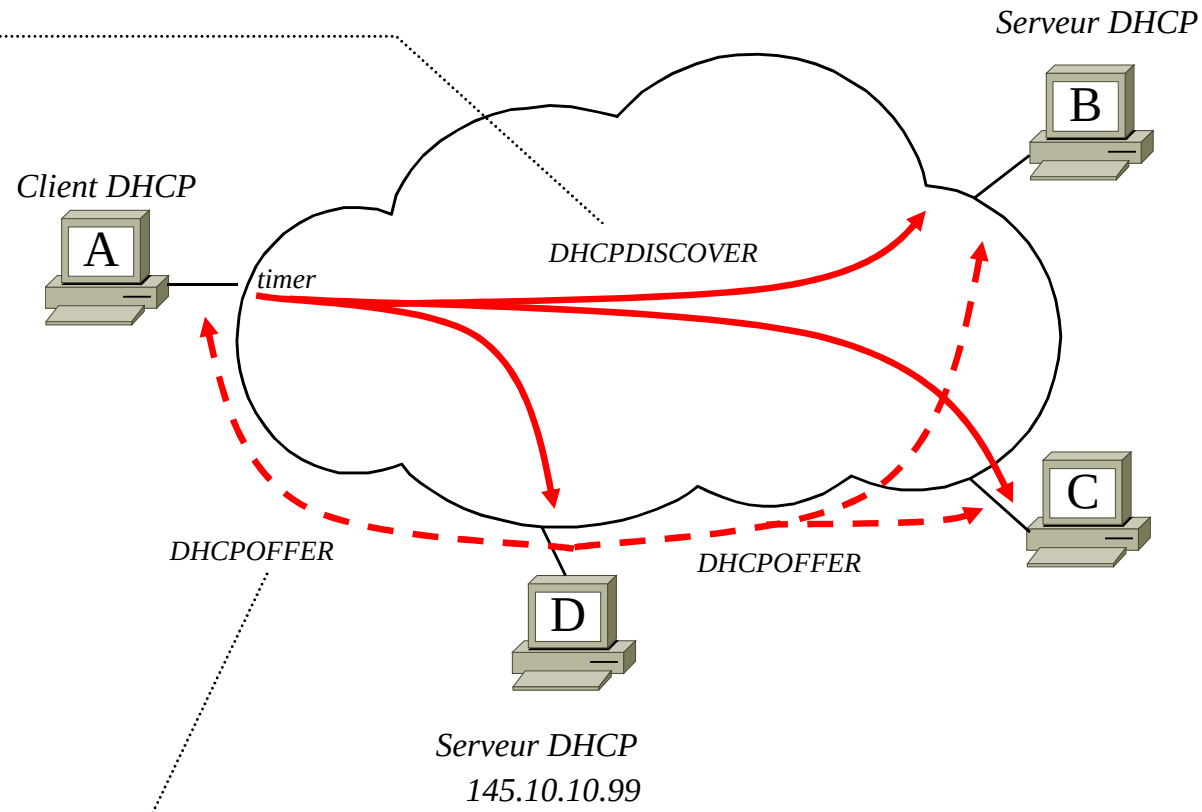
DHCP : format des paquets

- **HOPS**
  - Positionné à 0 par le client DHCP et incrémenté par chaque routeur
- **TRANSACTION ID**
- **Secondes**
  - Durée du contrat

DHCP : attribution dynamique

@IPs = 0.0.0.0
@IPd = 255.255.255.255
@HW = Mac(A)
Monnom

port 67

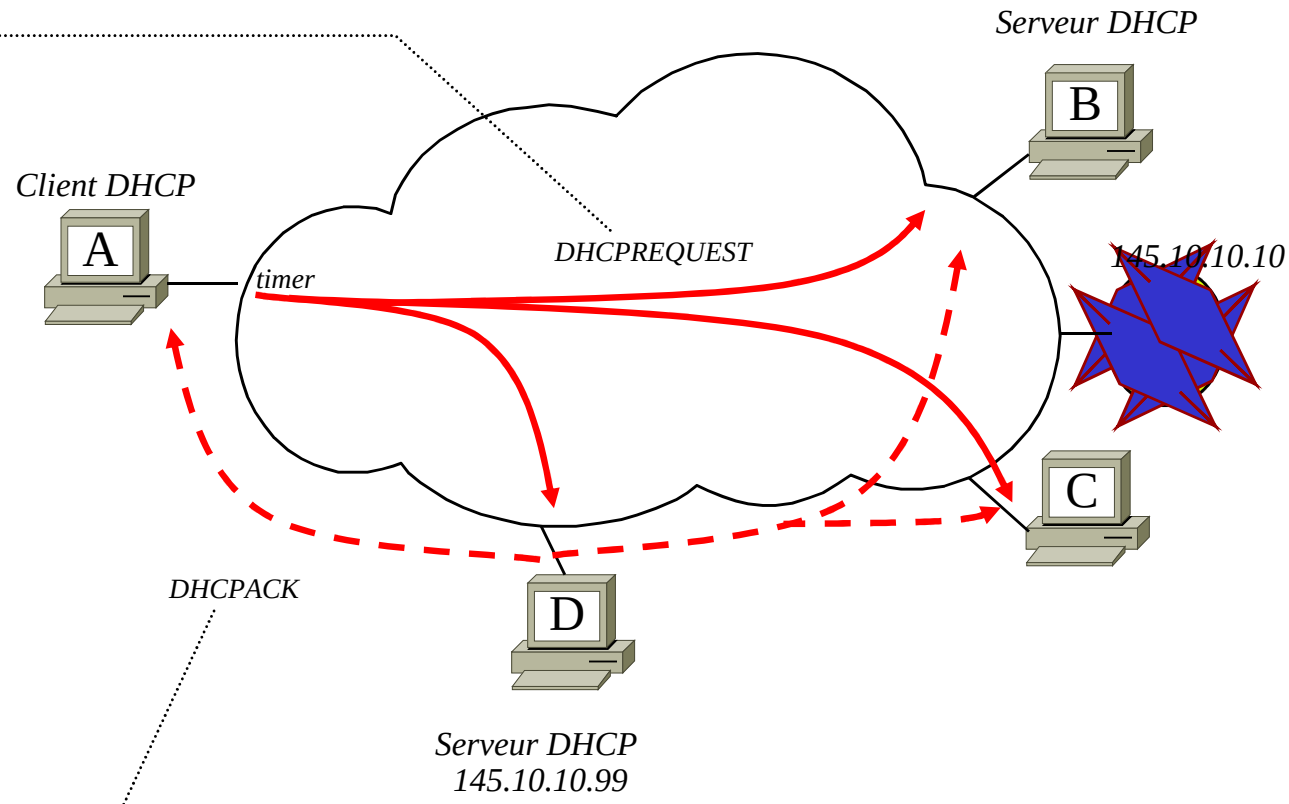


@IPs = 145.10.10.99
@IPd = 255.255.255.255
@IPoffered = 145.10.10.1
@HWc = Mac(A)
SNM = 255.255.255.0
Durée = 72 heures
Id serveur = 145.10.10.99

port 68

DHCP : attribution dynamique

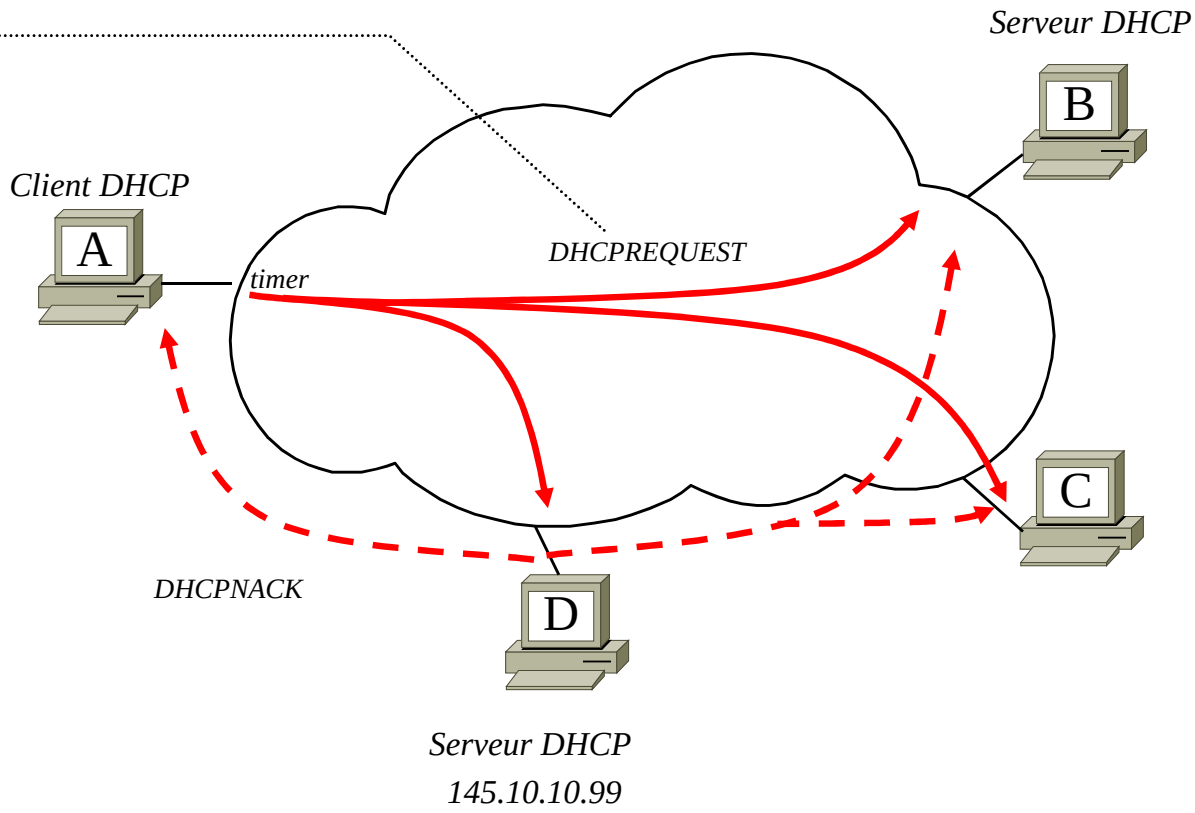
@IPs = 0.0.0.0
@IPd = 255.255.255.255
@HWc = Mac(A)
@IPrequested = 145.10.10.1
Server ID = 145.10.10.99



@IPs = 145.10.10.99
@IPd = 255.255.255.255
@IPoffered = 145.10.10.1
@HWc = Mac(A)
SNM = 255.255.255.0
Durée = 72 heures
Server ID = 145.10.10.99
Router = 145.10.10.10

DHCP : refus du contrat

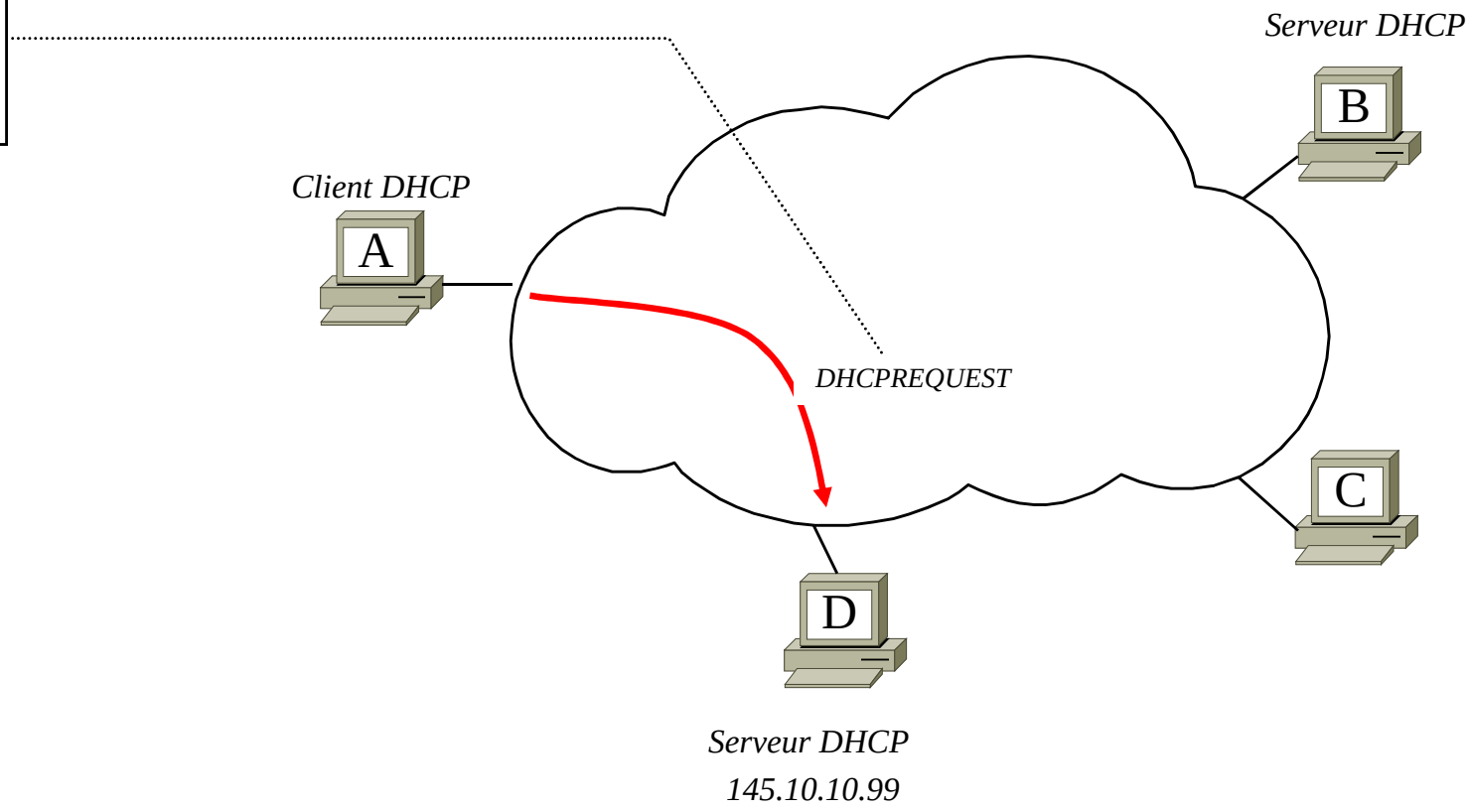
@IPs = 0.0.0.0
@IPd = 255.255.255.255
@HWc = Mac(A)
@IPrequested = 145.10.10.1
Server ID = 145.10.10.99



DHCP : renouvellement du contrat

Timer = 50% timerMax

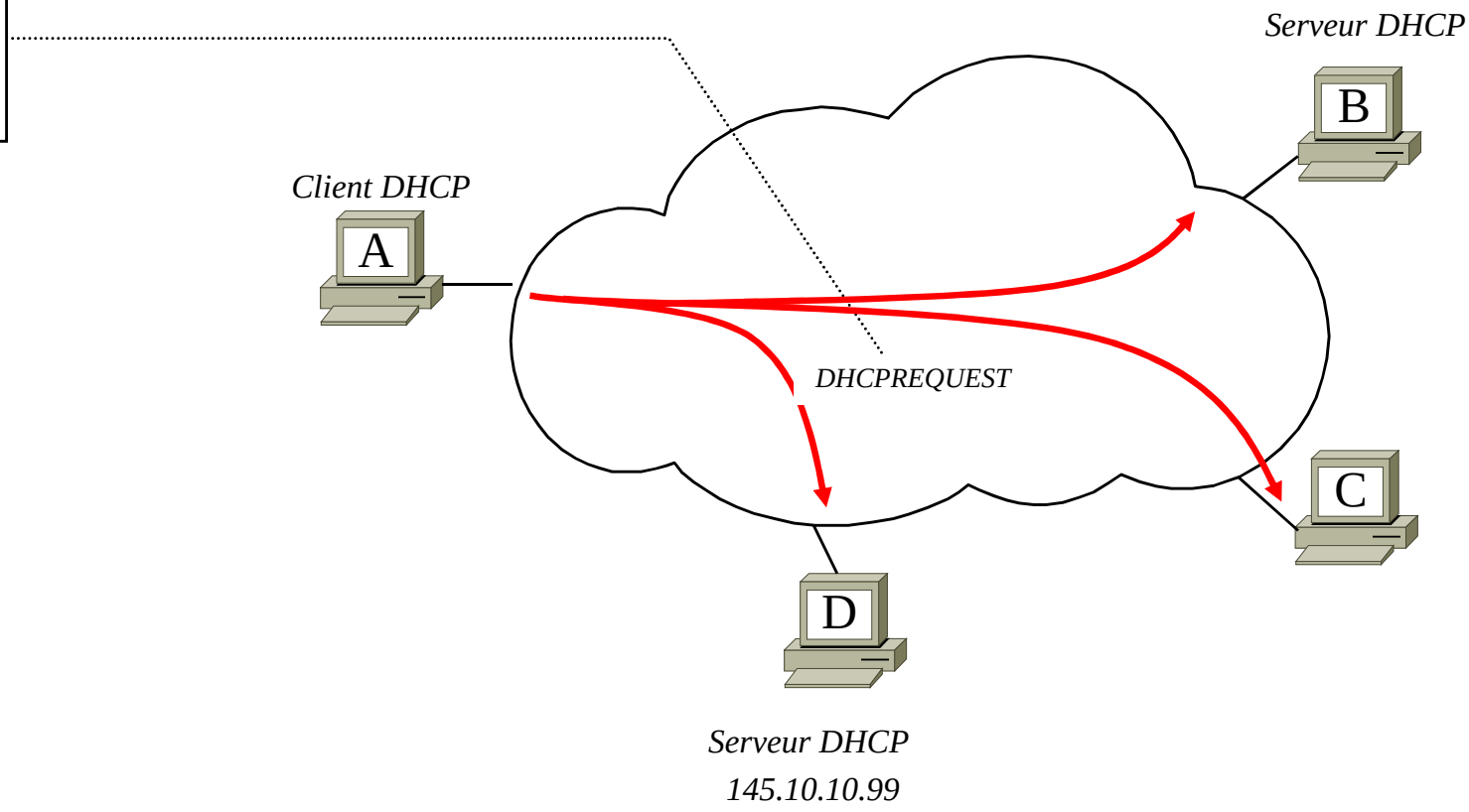
@IPs = 145.10.10.1
@IPd = 145.10.10.99
@HWc = Mac(A)
@IPrequested = 145.10.10.1
Server ID = 145.10.10.99



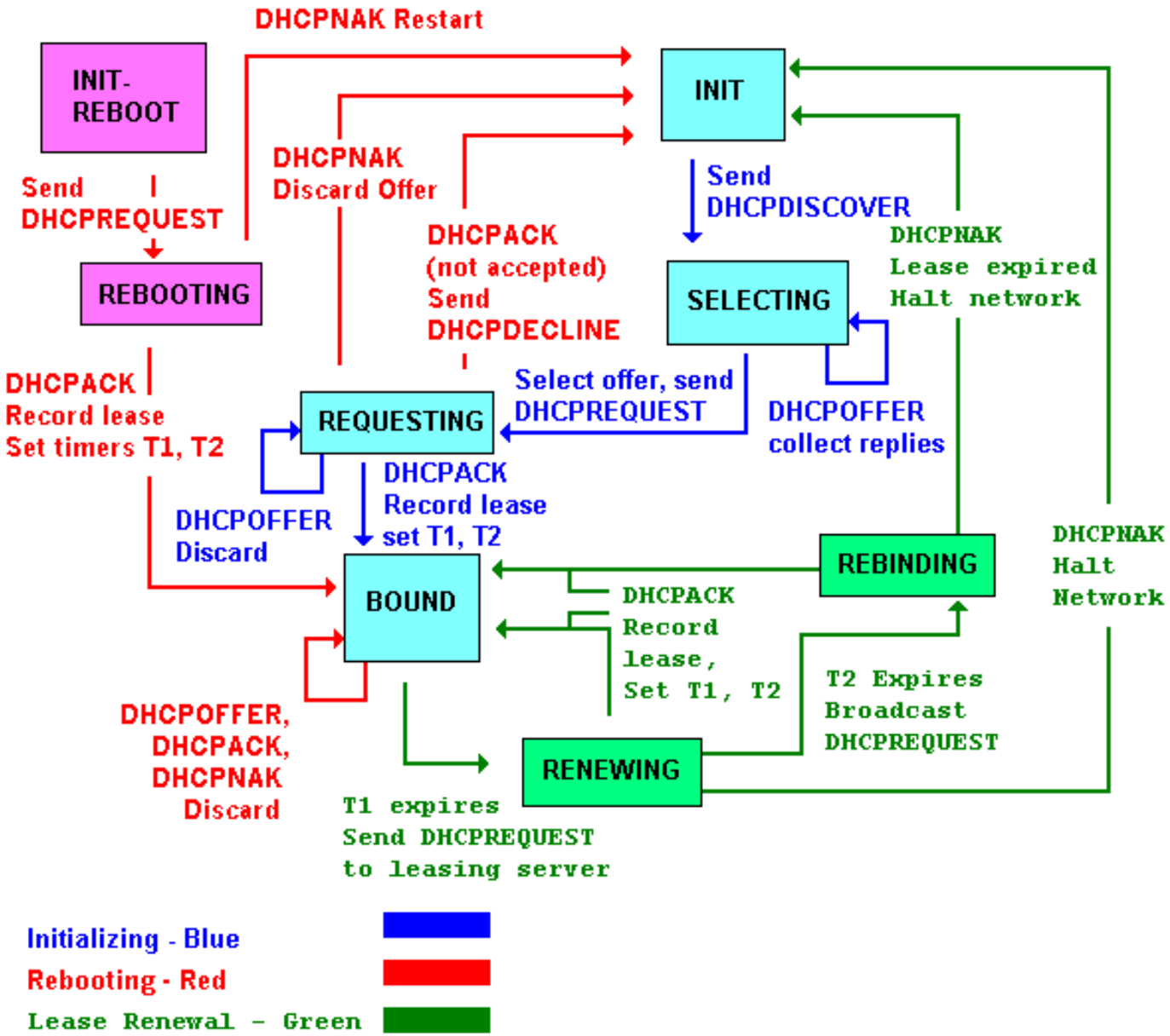
DHCP : renouvellement du contrat

Timer = 87,5% timerMax

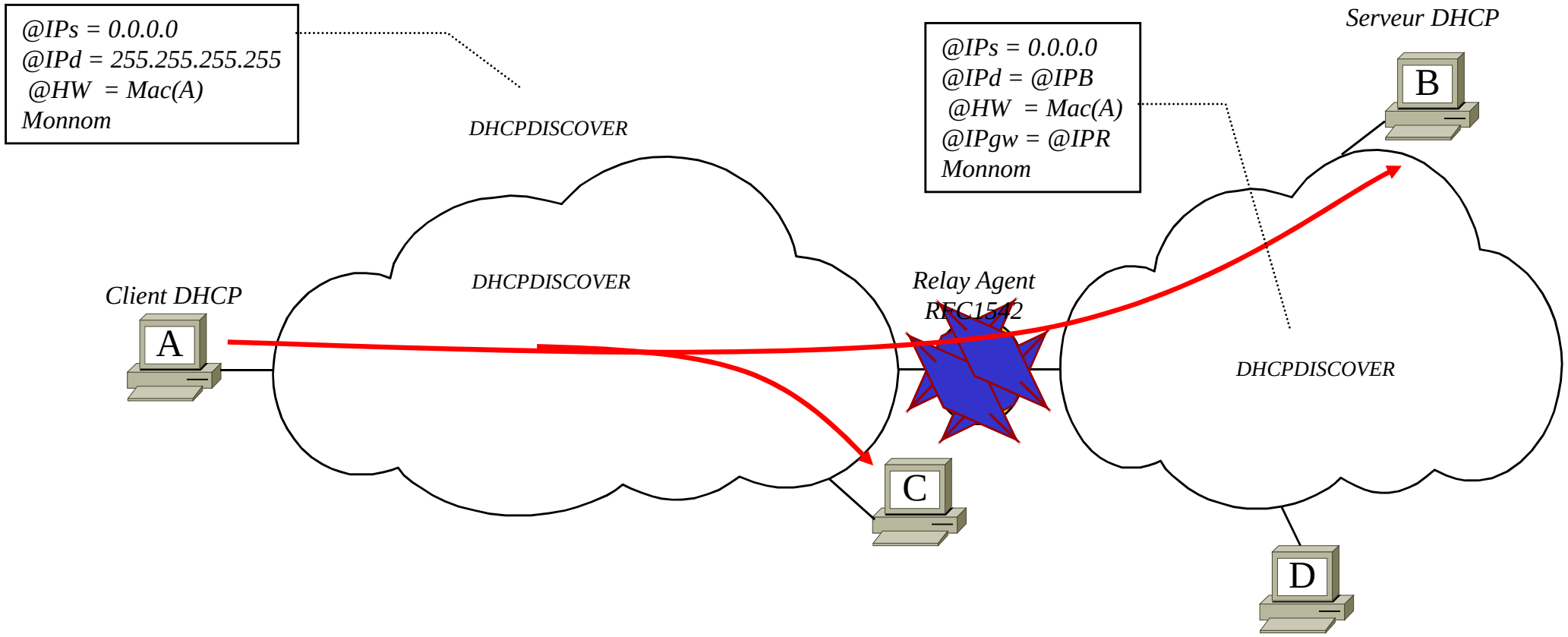
@IPs = 0.0.0.0
@IPd = 255.255.255.255
@HWc = Mac(A)
@IPrequested = 145.10.10.1
Server ID = 145.10.10.99



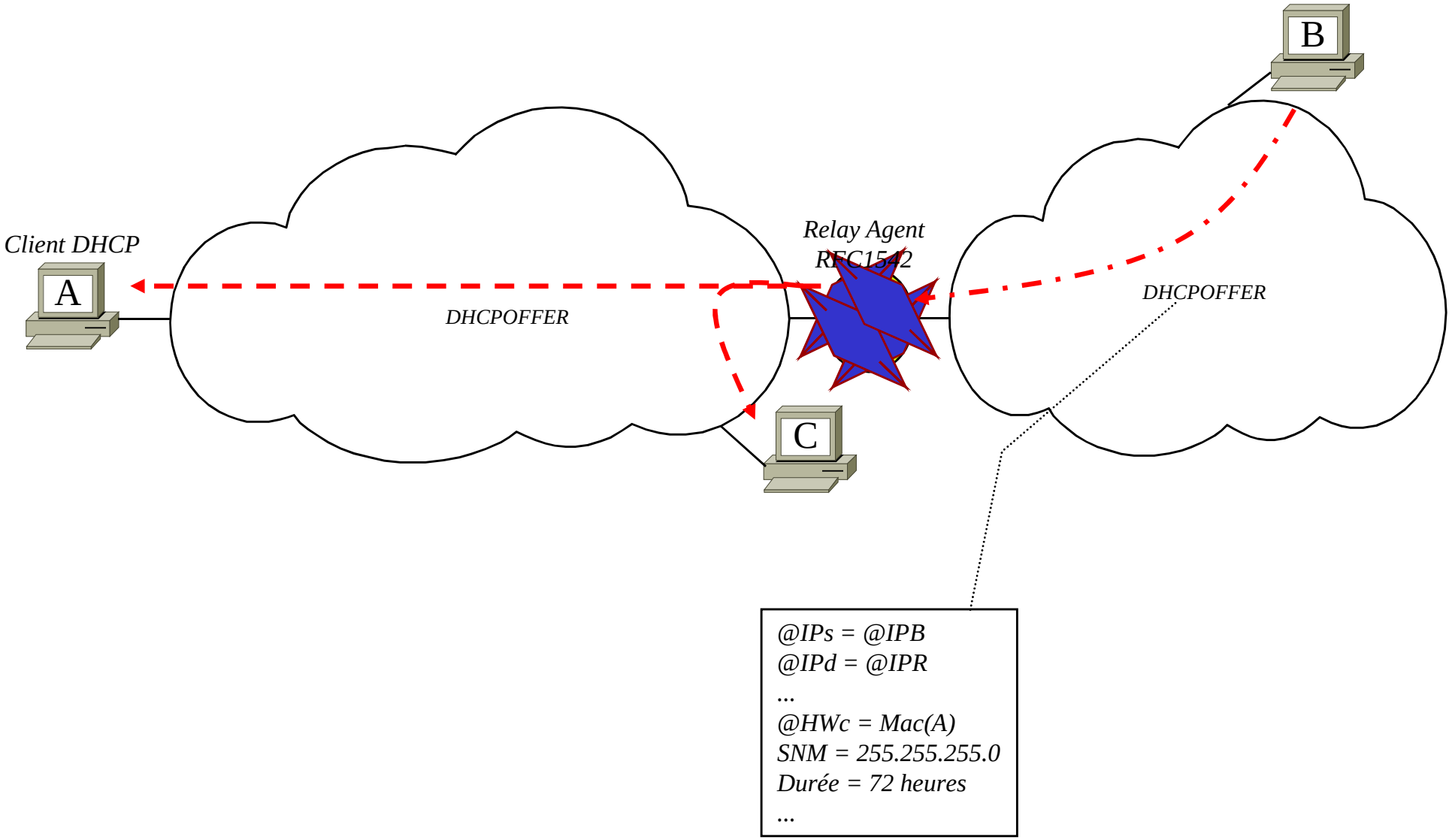
DHCP : diagramme d ’état



DHCP : configuration dynamique via routeur



DHCP : configuration dynamique via routeur



DHCP : documents de references

- RFC 2131: Dynamic Host Configuration Protocol
- RFC 2132: DHCP Options and BOOTP Vendor Extensions
- RFC 1534: Interoperation Between DHCP and BOOTP
- RFC 2136: Dynamic Updates in the Domain Name System (DNS UPDATE)
- RFC 2137: Secure Domain Name System Dynamic Update