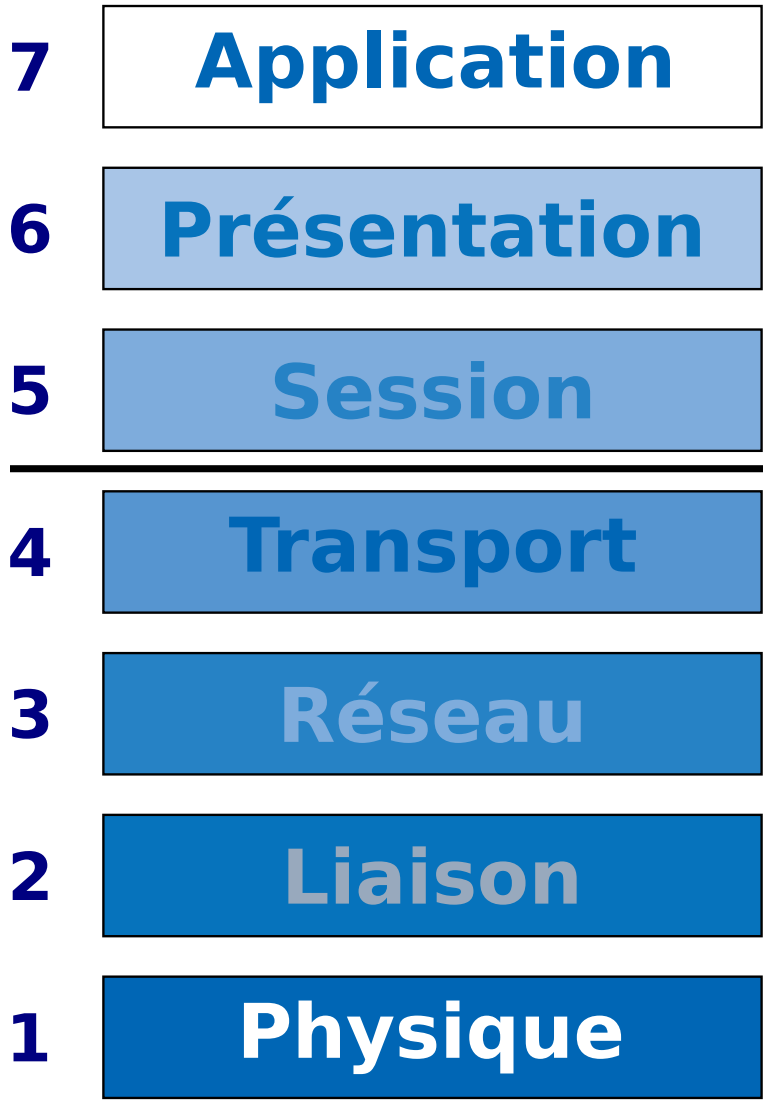
Les protocoles TCP / IP

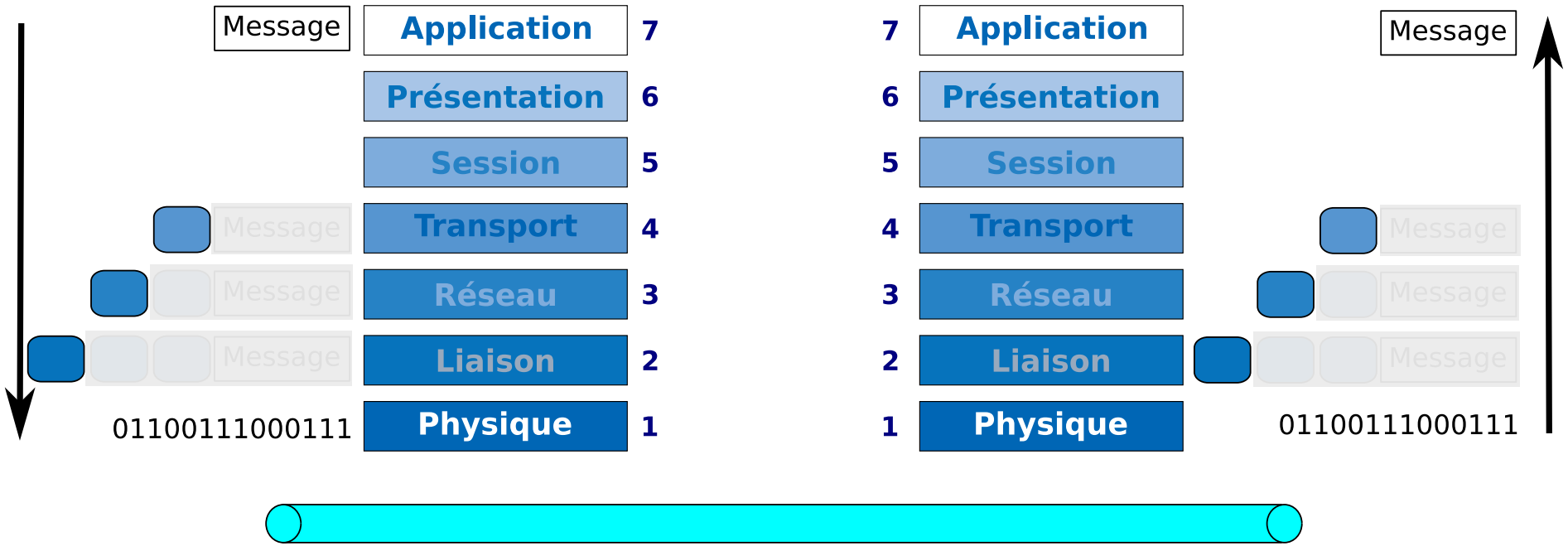
Yvan Peter

IUT A - Université de Lille

Norme Open Systems Interconnection (OSI) : ISO 7498-1

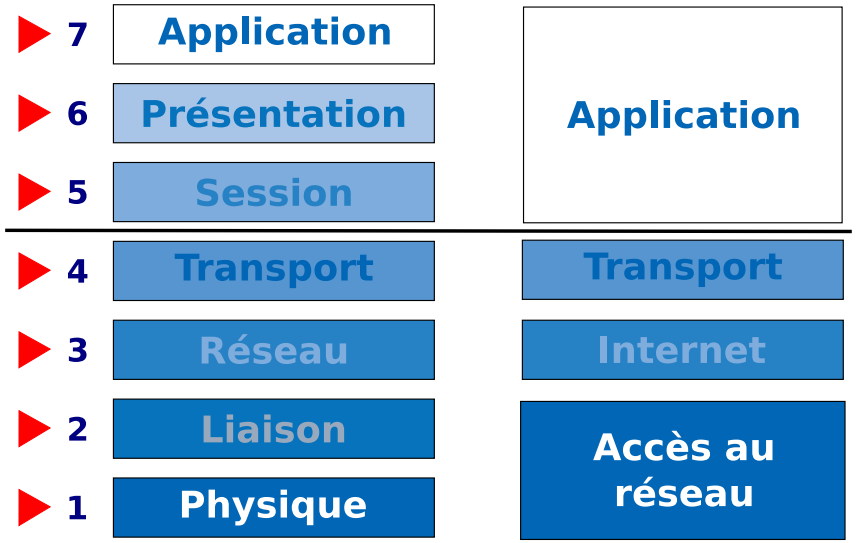


Encapsulation



- Chaque couche ajoute une en-tête propre à son protocole
- Les informations transmises par une couche sont opaques pour la couche inférieure

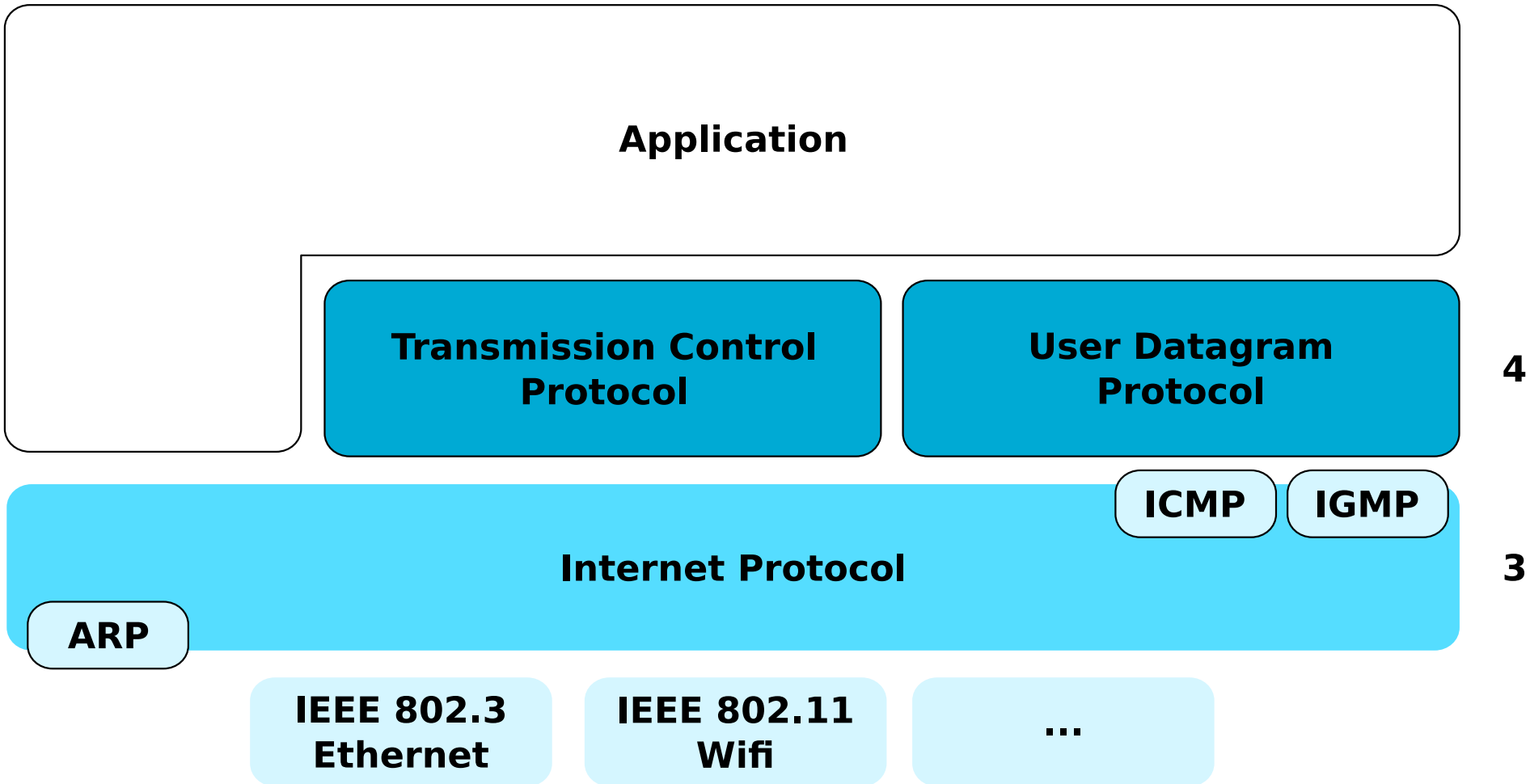
Modèle en couches : TCP/IP



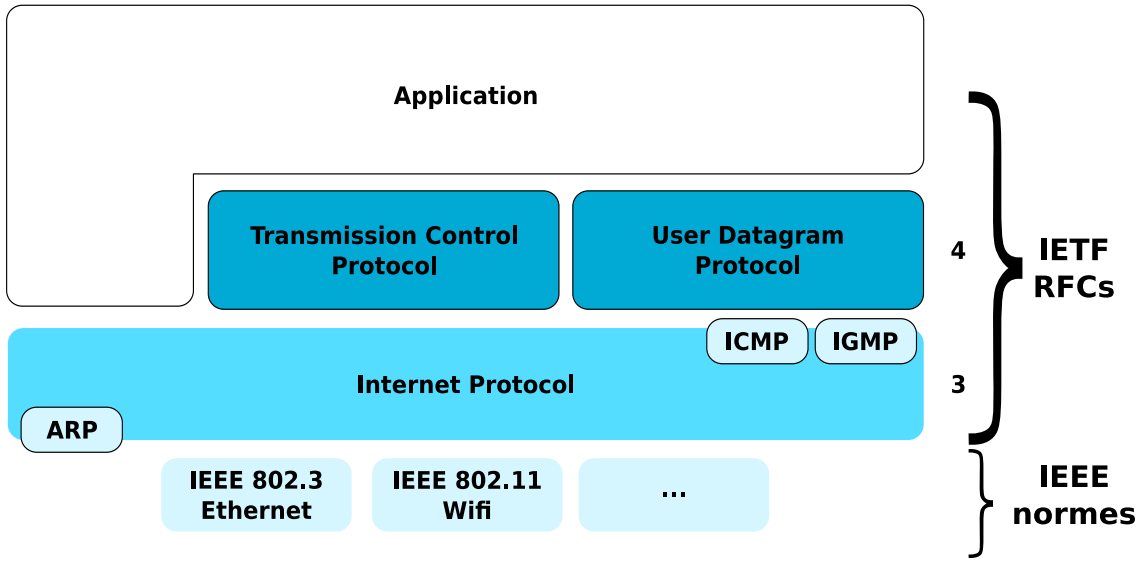
Le modèle TCP/IP

- Modèle pragmatique issu des travaux du *Departement of Defense* (DOD) et d'ARPANET
- L'implémentation de la couche 3 correspond au protocole **Internet Protocol**
- La couche 4 offre deux protocoles : **Transmission Control Protocol** (mode connecté, fiable) et **User Datagram Protocol** (mode déconnecté, non fiable)

Modèle en couches : TCP/IP



Modèle en couches : TCP/IP



La standardisation

- Les couches 1 et 2 sont normalisées par l'IEEE.
- L'*Internet Engineering Task Force* (IETF) produit des *Request for Comments* (RFC) relatifs aux protocoles et au fonctionnement d'Internet
- La plupart des RFC sont disponibles en français

Internet Protocol (RFC 791)

- IP est un protocole de niveau réseau (3) destiné à acheminer des données d'une machine à une autre.
- Service rendu "au mieux" (*best effort*)
- Une opération de routage à chaque sortie de réseau
  - chaque paquet est routé individuellement
  - des paquets pourront manquer ou arriver dans le désordre
- IP fournit une vision unifiée du réseau
  - mécanisme d'adressage et de routage de "haut niveau"
  - mécanisme de correspondance entre adresse IP et adresse physique

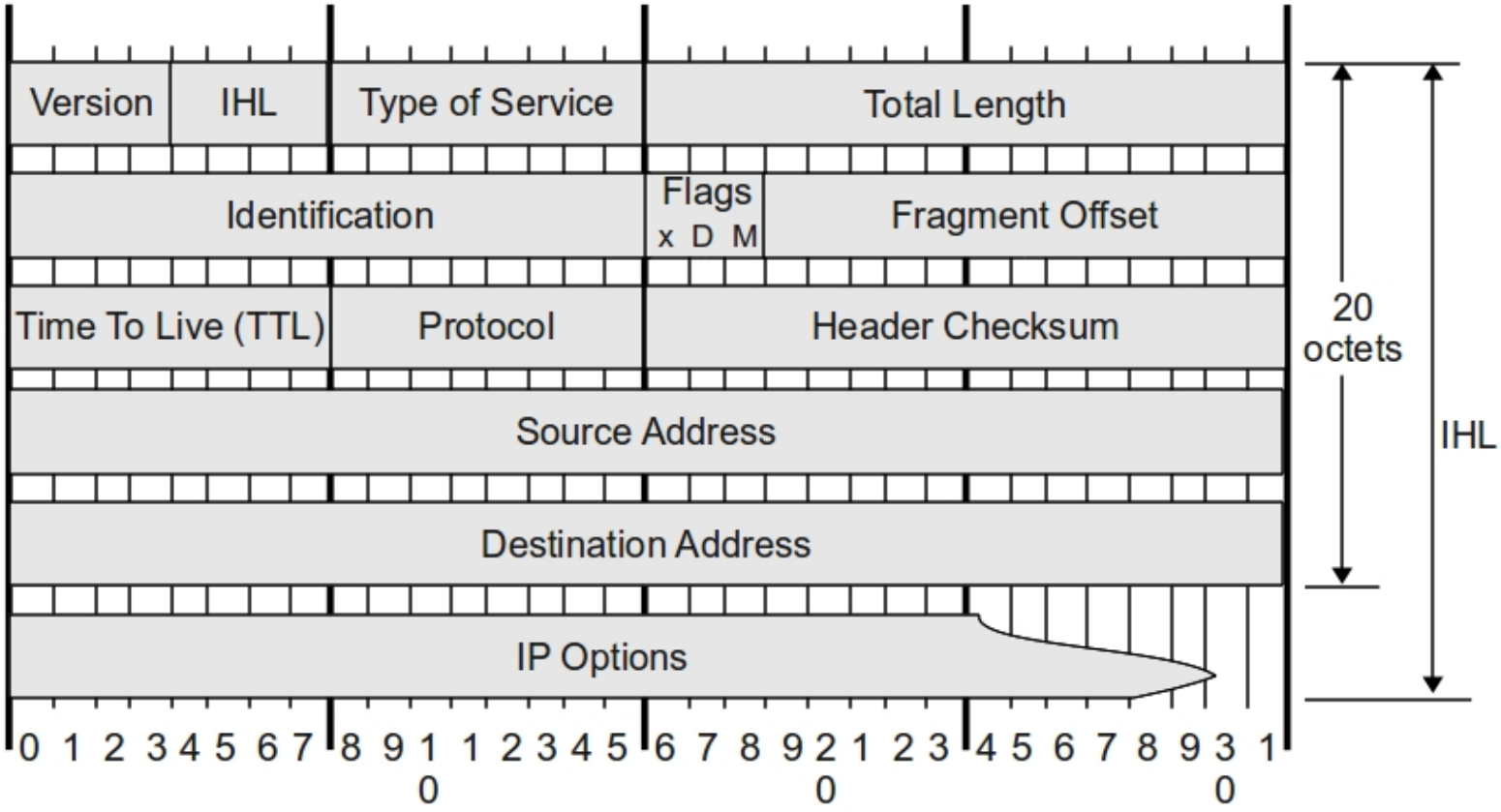
Internet Protocol

- Une mécanisme de **fragmentation** permet de transmettre les datagrammes sur des réseaux de capacités différentes
- Les fragments sont rassemblés par le destinataire pour reconstituer le datagramme initial

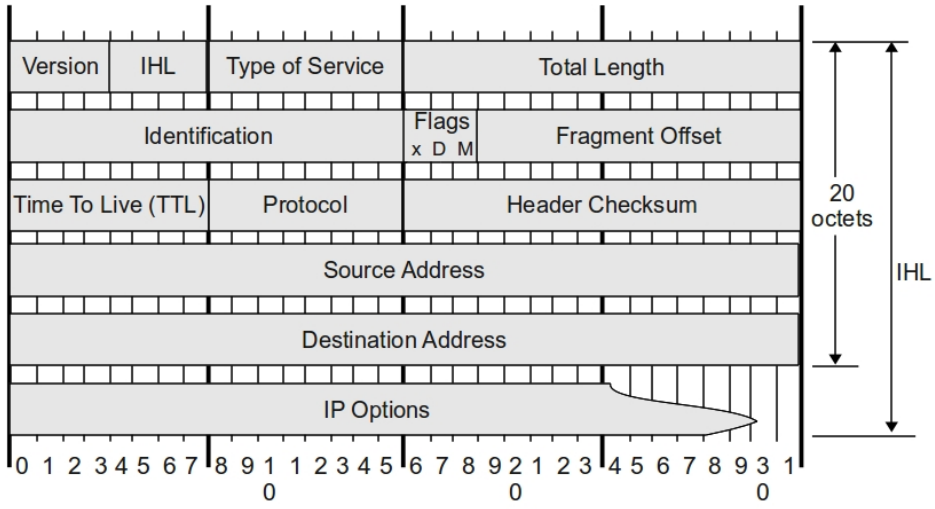
Maximum Transmission Unit

Le MTU est la taille maximale des données qui peuvent être transportées dans une trame pour un réseau donné.
Pour Ethernet = 1500 octets
PPPoE = 1492 octets
WLAN (802.11) = 2304 octets

En-tête IPv4


Version
IHL
Type of Service
Total Length
Identification
Flags
x D M
Fragment Offset
Time To Live (TTL)
Protocol
Header Checksum
Source Address
Destination Address
IP Options
20 octets
IHL
0 1 2 3 4 5 6 7
8 9 1 0
1 2 3 4 5
6 7 8 9 2 0
1 2 3
4 5 6 7 8 9 3 0
1

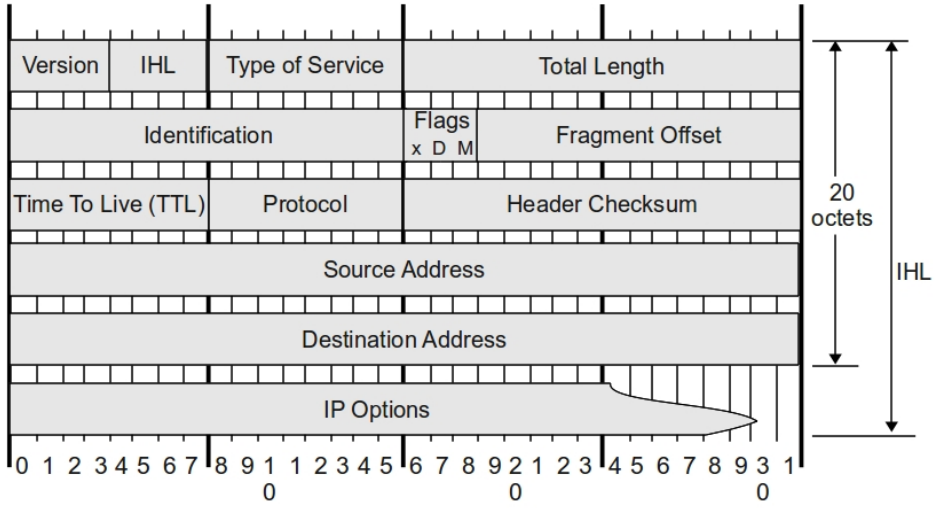
En-tête IPv4



Version (4 bits)

Le champ version indique la version d'IP utilisée. C'est le seul champ commun à IPv4 et IPv6

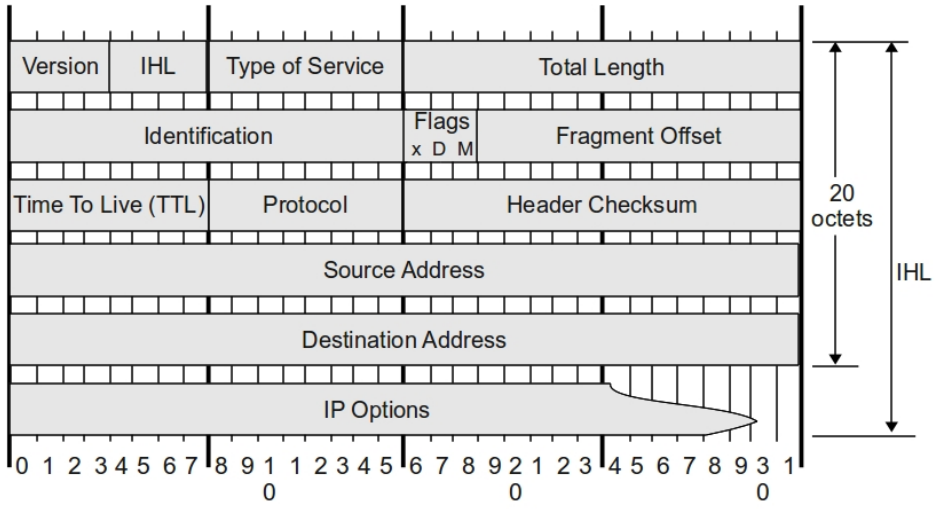
En-tête IPv4



Taille de l'en-tête (4 bits)

- Ce champ indique la taille de l'en-tête en mots de 32 bits
- La valeur doit être comprise entre 5 et 16.
- En général on a 20 octets (5 mots)

En-tête IPv4



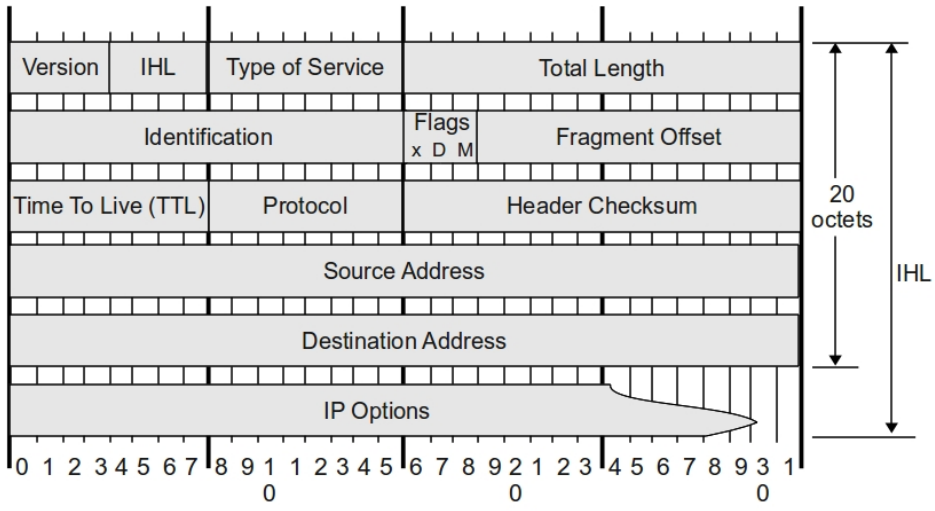
Type de Service (8 bits)

Ce champs permet de spécifier une qualité de service pour le traitement des paquets.

Dans sa version actuelle il est décomposé en

- *Differentiated Services Field* (6 bits) (RFC 2474)
- *Explicit Congestion Notification* (2 bits) (RFC 3168)

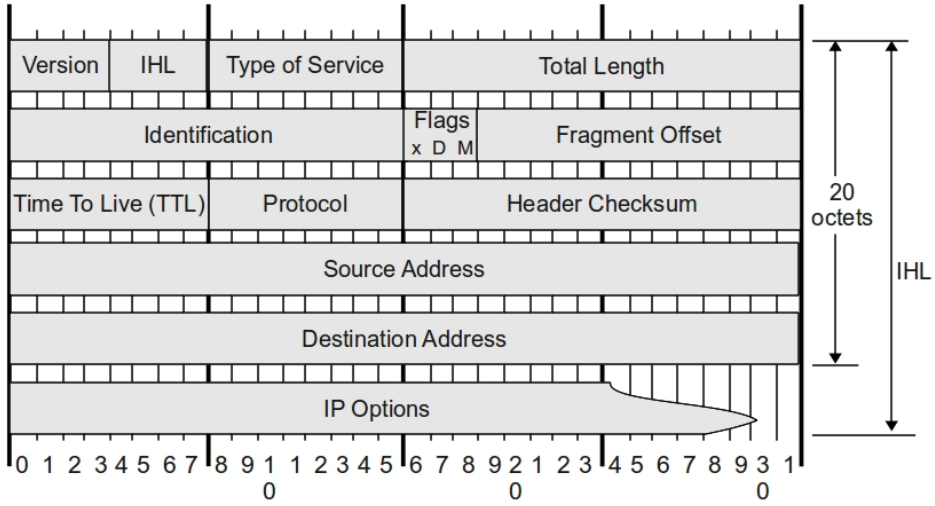
En-tête IPv4



Taille totale (16 bits)

Taille totale du paquet, en-tête comprise (max. 65535 octets)

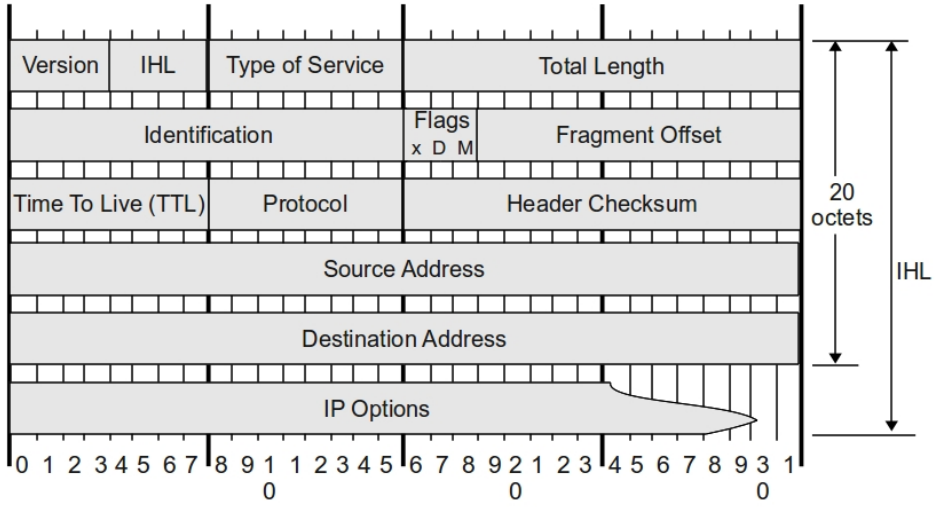
En-tête IPv4



Gestion de la fragmentation (16 + 16 bits)

- Identification : permet de rassembler les fragments d'un même paquet
- Drapeaux (3 bits) :
  - 1 bit **D**on't **F**ragment (DF) : fragmentation interdite ? (1 = oui)
  - 1 bit **M**ore **F**ragment (MF) : encore des fragments après ? (1 = oui)
- Déplacement (*Offset*) : emplacement du fragment dans le paquet initial (en unité de 8 octets)

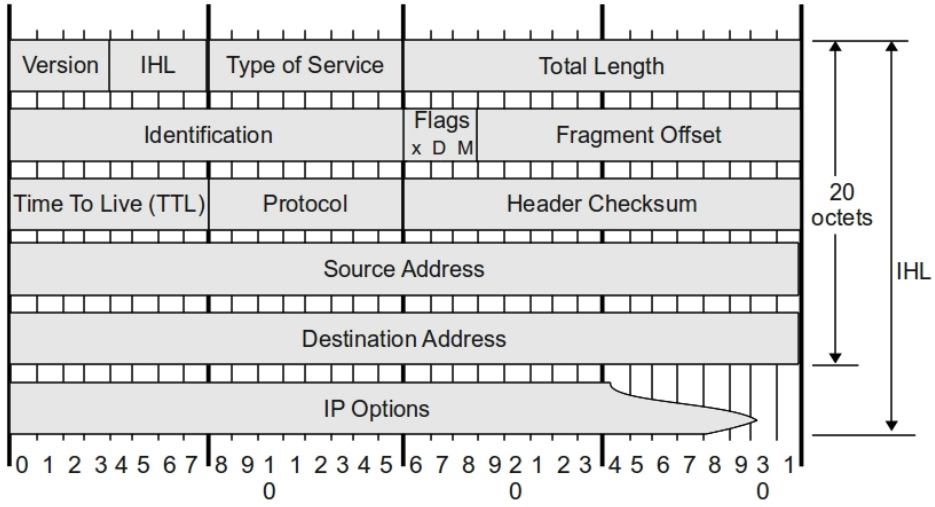
En-tête IPv4



Durée de vie (Time to Live ou TTL) (8 bits)

- Nombre maximum de routeurs traversés (≤ 255)
- Décrémenté de 1 à chaque routeur
- Si le TTL tombe à 0, le paquet est détruit

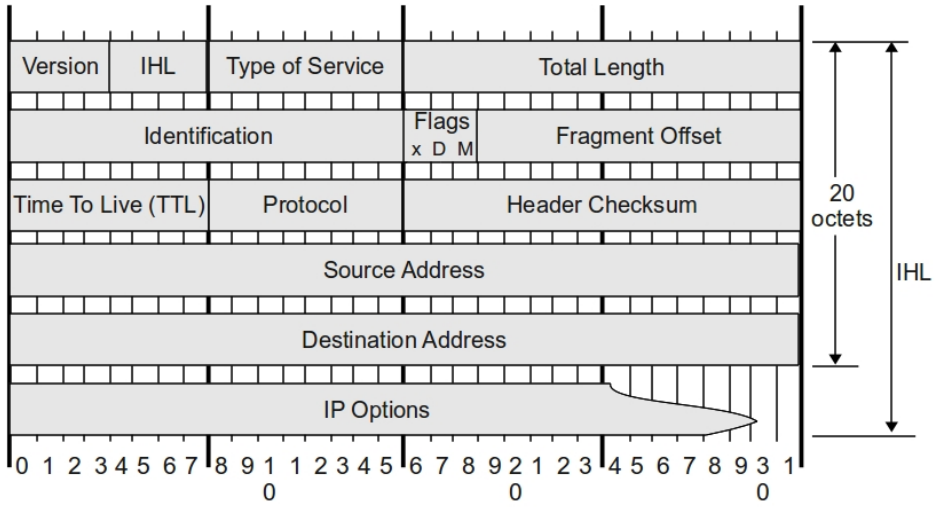
En-tête IPv4



Protocole (8 bits)

- Indique le protocole transporté (ce qu'on trouve après l'en-tête IP)
- Numéros répertoriés par l'IANA.
- Les plus courants : 1 – ICMP, 6 – TCP, 17 – UDP

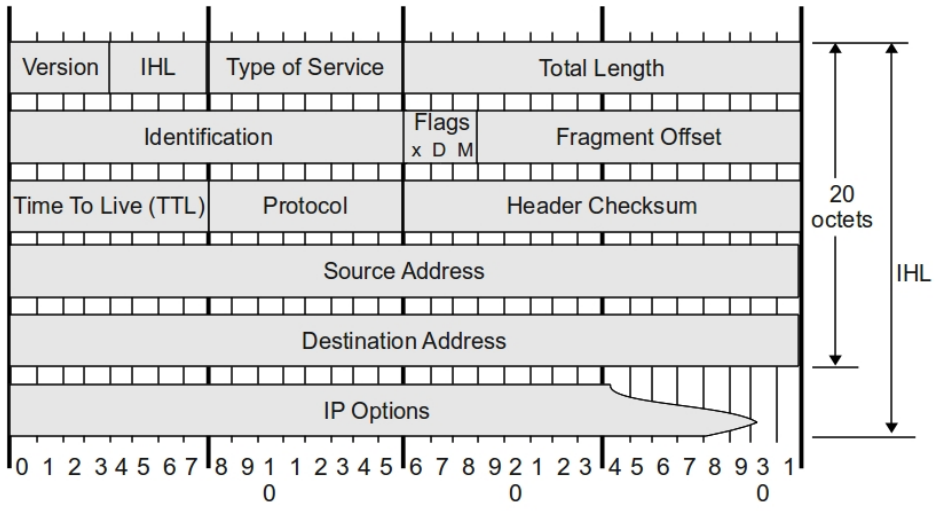
En-tête IPv4



Somme de contrôle (16 bits)

Calculée sur l'en-tête IP, elle permet de vérifier que celle-ci n'a pas été corrompue.

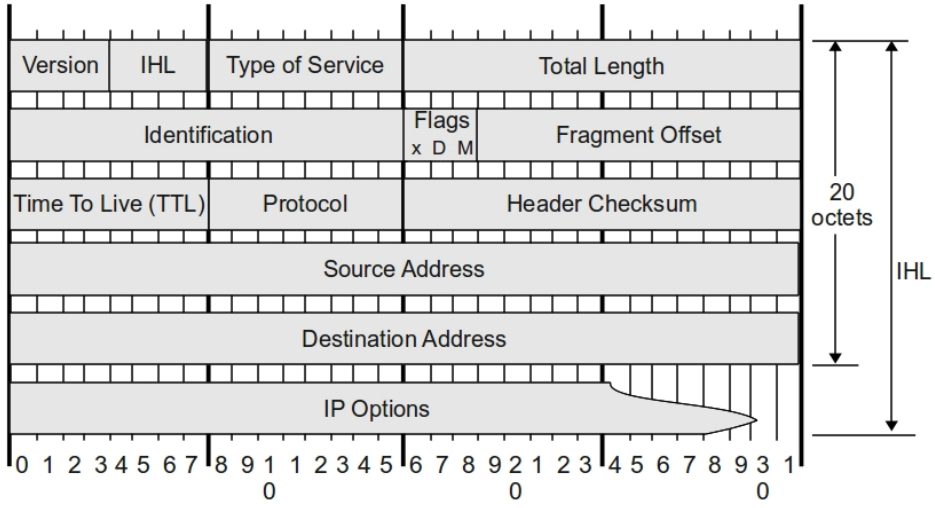
En-tête IPv4



Adresses source et destination (32 + 32 bits)

Indique l'adresse de l'émetteur et du destinataire

En-tête IPv4



Options

Option historiques pour la plupart. Rarement traitées par les routeurs.

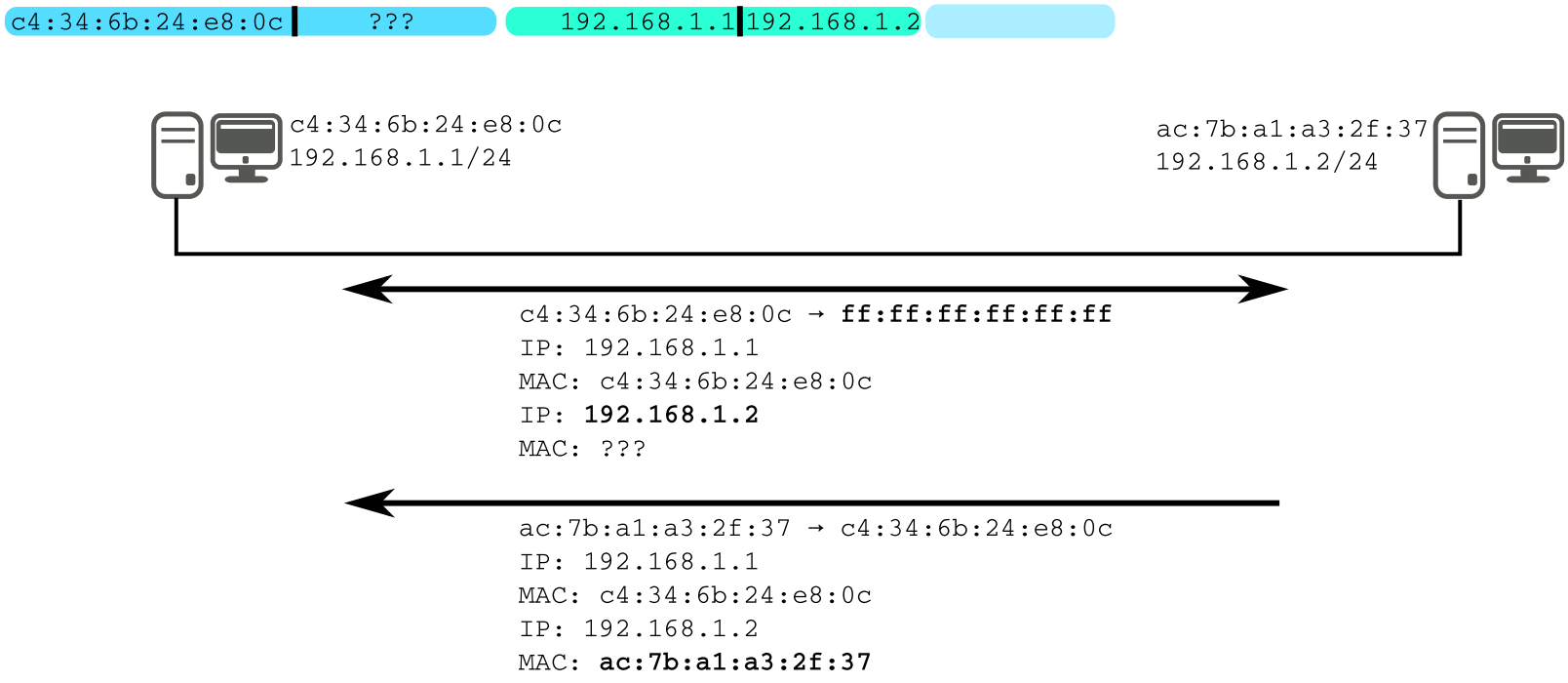
Address Resolution Protocol

Le lien entre adresse IP et adresse de niveau 2

Address Resolution Protocol (RFC 826)

- Lors de l'envoi d'un paquet, la machine compare son adresse de réseau avec celle de la machine destinataire
- Si c'est le même réseau
  - On fait une remise directe
  - La trame est émise vers cette machine
- Si le réseau est différent
  - On doit confier le paquet au routeur de sortie du réseau
  - La trame est émise vers le routeur
- Pour émettre une trame, il faut une adresse de niveau 2 (adresse MAC)
- ARP permet d'apprendre les adresses MAC

Address Resolution Protocol



fonctionnement

- Requête en diffusion et réponse directe de la machine concernée.
- Utilisation d'un cache pour limiter les requêtes
- Consultation/manipulation du cache : `ip neigh [options]`

Internet Control Message Protocol

La gestion des erreurs et les tests

ICMP (RFC 792)

- IP n'est pas un protocole fiable
- ICMP permet :
  - De signaler les erreurs et pertes de paquets
  - D'indiquer des optimisations possibles (choix du routeur de sortie)
  - De faire des tests de connectivité (`ping`)
- Les paquets ICMP sont transportés par IP
- La perte d'un paquet ICMP ne génère pas d'erreur ICMP…

ICMP

- Types de messages courants :

Type	Nom	Usage
0	Echo Reply	réponse à un ping
3	Destination Unreachable	impossible de joindre l'hôte ou l'application
5	Redirect	indique un autre routeur à utiliser
8	Echo	paquet de requête à un ping
11	Time Exceeded	TTL tombé à 0
12	Parameter Problem	paquet ou en-tête malformé(e)

- Certain types sont raffinés en codes d'erreur / d'information plus précis
- Un paquet d'erreur ICMP transporte le début du paquet détruit (jusqu'à 576 octets) afin de pouvoir retrouver les adresses et numéros de ports concernés.

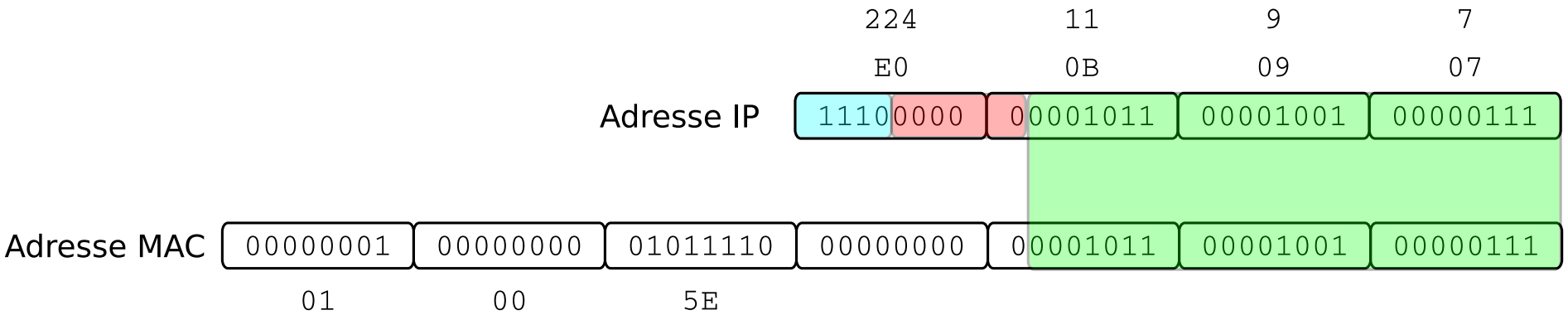
Communication Multicast

- Les communications multicast sont transportées par IP sur la base des adresses de classe D.
- Trois mécanismes sont nécessaires :
  - Un mécanisme de correspondance @IP → @MAC
  - Un mécanisme de contrôle de l'étendue de la diffusion
  - Un mécanisme de diffusion aux membres du groupe
    - Gestion des membres présents sur un réseau
    - Routage multicast

Communication Multicast

Mécanisme de correspondance

- Utilise une correspondance **statique**
- Adresse MAC de multicast prédéfinie (**0x01005E**000000)
- On ajoute les 23 derniers bits de l'adresse IP
  - 32 adresses sont projetées sur une seule adresse MAC
  - L'hôte doit filtrer pour vérifier que c'est un groupe qui l'intéresse



Communication Multicast

Contrôle de l'étendue de la diffusion

- Le contrôle de la diffusion repose sur le TTL
  - TTL = 0 : diffusion limitée à la source
  - TTL = 1 : diffusion à tous les membres du réseau local
    - Le TTL passe à zéro sans générer d'erreur ICMP
  - TTL ≥ 2 : la diffusion aux autres réseaux dépend de l'adresse destination
    - 224.0.0.0/24 : diffusion limitée au réseau local
    - Adresse de diffusion globale : routage normal avec décrémentation du TTL à chaque routeur

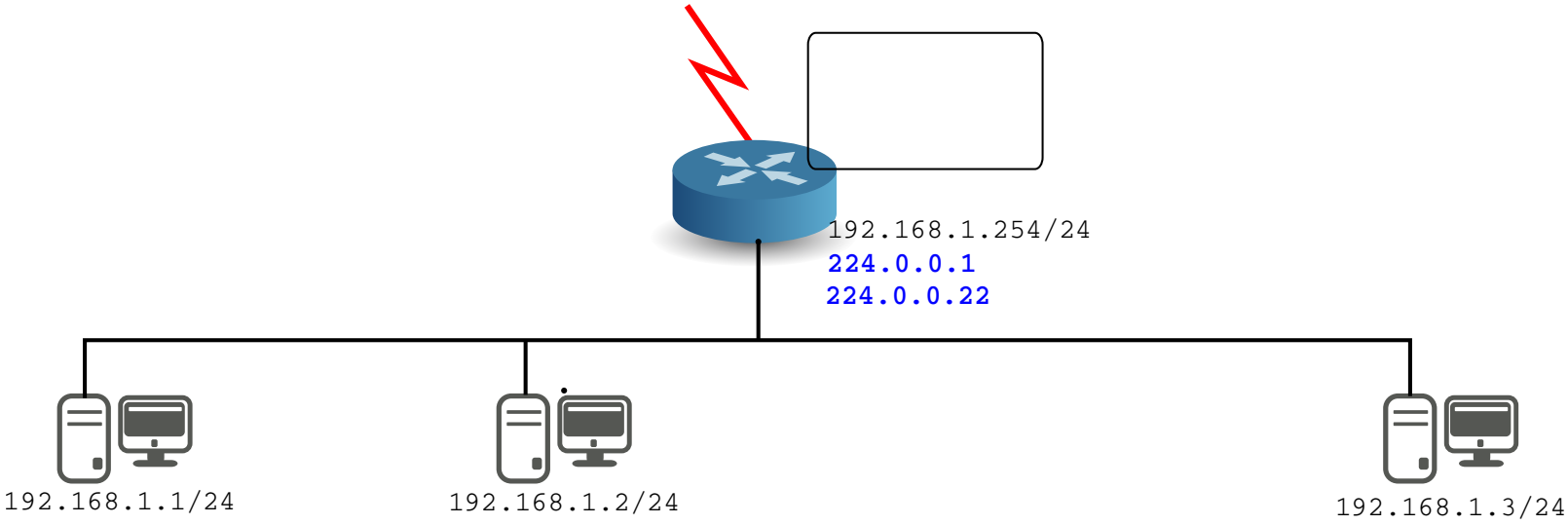
Internet Group Management Protocol

La gestion du multicast sur le LAN

IGMPv3 (RFC 3376)

- Pour écouter un groupe multicast, la pile IP :
  - Fournit un mécanisme d'adhésion à un groupe
  - Permet de définir un filtrage sur les sources (IGMPv3)
- IGMP permet à un routeur multicast de connaître les groupes souscris sur un réseau local
- IGMPv3 utilise deux types de messages
  - Demande d'appartenance (*Membership Query*). Envoyé par le routeur à l'adresse `224.0.0.1`
  - Rapport d'appartenance (*Membership Report*). Envoyé par les hôtes multicast à l'adresse `224.0.0.22`
- Les paquets IGMP sont transportés par IP (valeur de protocole : 2)
- La destruction d'un paquet IGMP ne génère pas d'erreur ICMP...

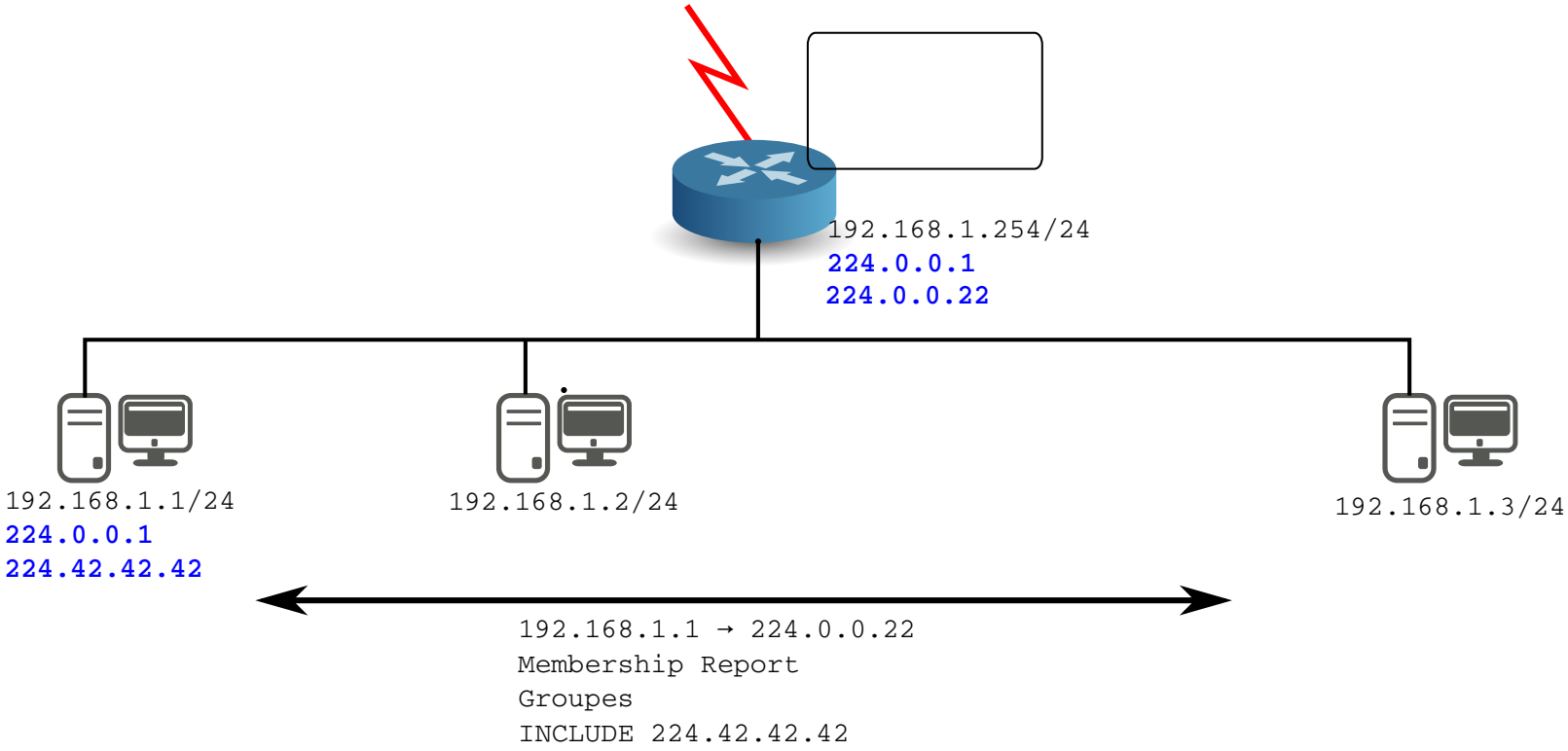
IGMPv3 : Protocole



Adhésion

- Une machine peut faire une demande d'adhésion pour un groupe et une ou plusieurs sources associées
- Le routeur multicast doit connaître les groupes et sources desservies sur le réseau local

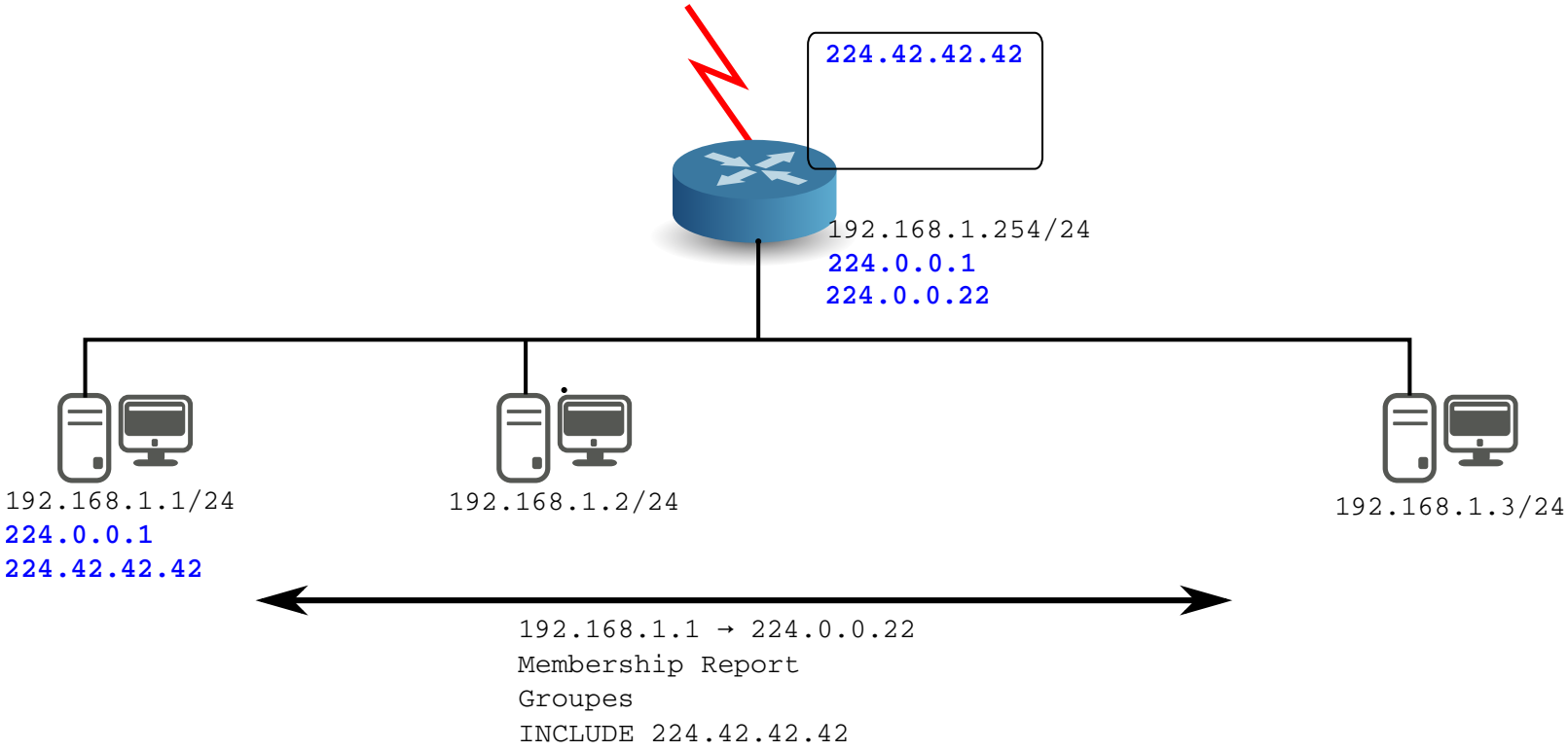
IGMPv3 : Protocole



Adhésion

- Une machine peut faire une demande d'adhésion pour un groupe et une ou plusieurs sources associées
- Le routeur multicast doit connaître les groupes et sources desservies sur le réseau local

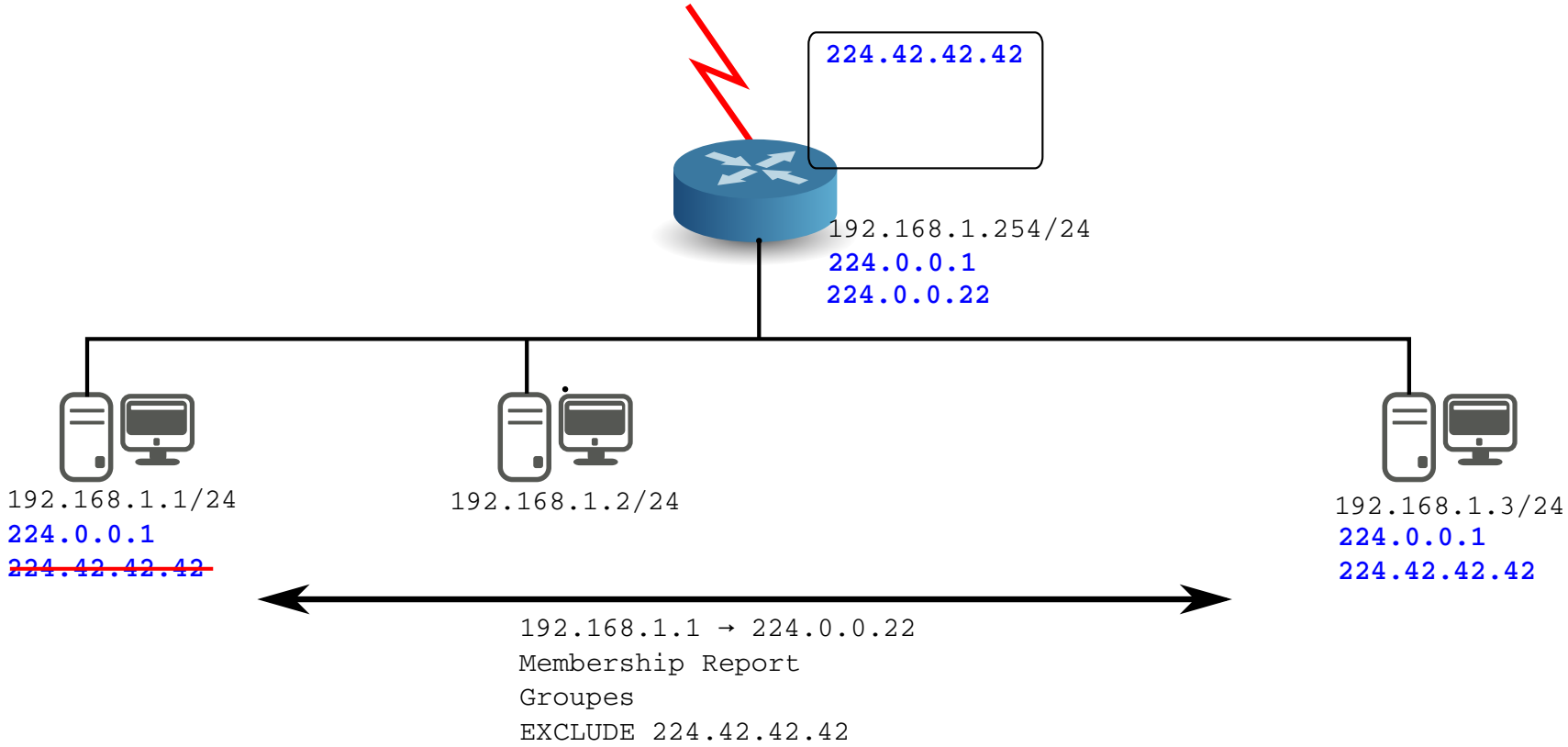
IGMPv3 : Protocole



Adhésion

- Une machine peut faire une demande d'adhésion pour un groupe et une ou plusieurs sources associées
- Le routeur multicast doit connaître les groupes et sources desservies sur le réseau local

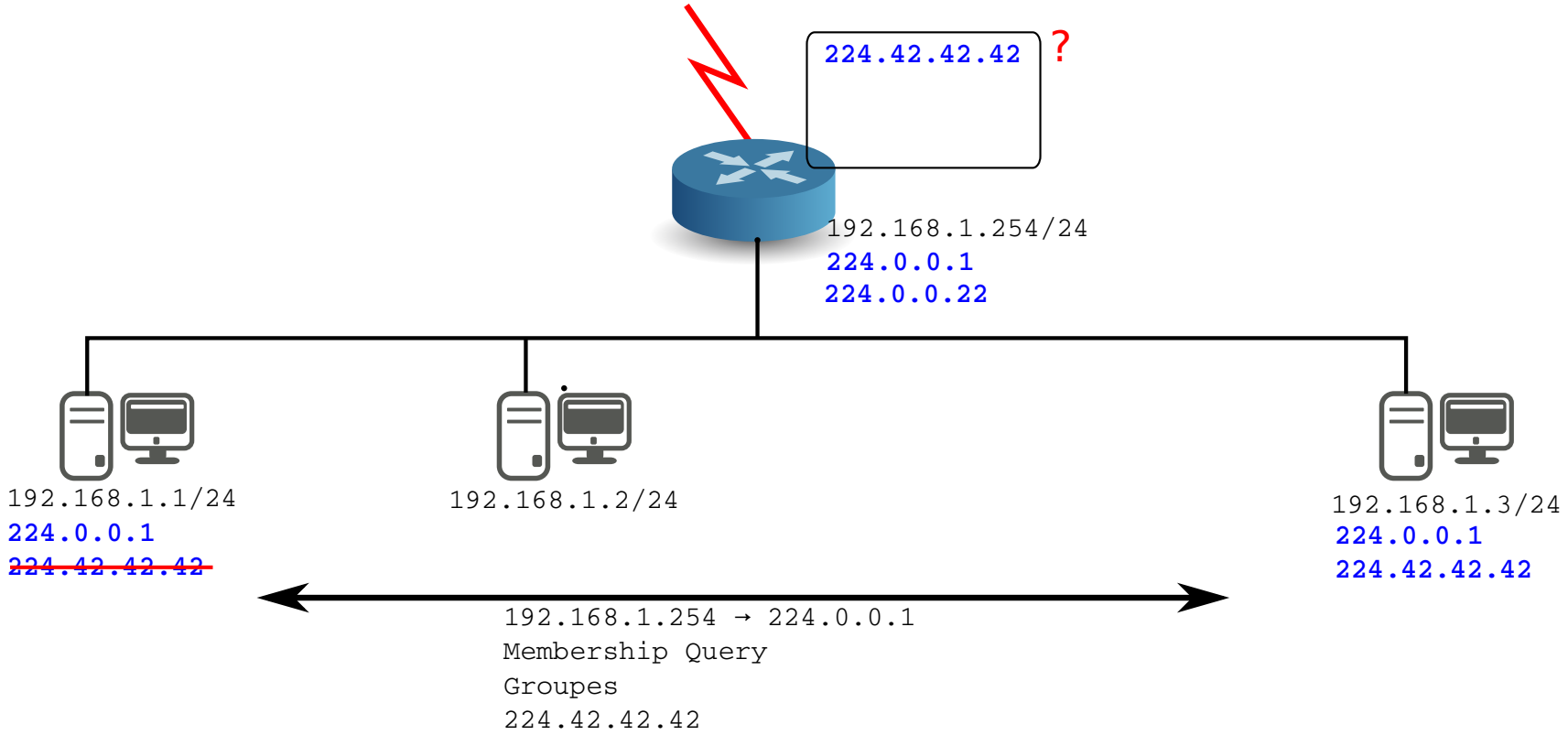
IGMPv3 : Protocole



Retrait

- Un hôte signale les modifications dans les groupes et sources souscrits.
- En cas de retrait, le routeur multicast doit savoir s'il reste des machines intéressées par le(s) groupe(s)/source(s)

IGMPv3 : Protocole



Retrait

- Un hôte signale les modifications dans les groupes et sources souscrits.
- En cas de retrait, le routeur multicast doit savoir s'il reste des machines intéressées par le(s) groupe(s)/source(s)

IGMPv3 : Protocole

- Le routeur envoi une demande d'appartenance générale à 224.0.0.1
  - A quel groupe/source voulez-vous vous abonner ?
  - Toutes les 100 secondes par défaut
- Un hôte renvoie un rapport d'appartenance
  - Qui indique les adresses des groupes/sources souscrits
  - Après un délai aléatoire
  - Une seule réponse suffit pour chaque groupe/source
- S'il n'y a pas de réponse pour un groupe/source donné, le routeur ne réémettra pas les paquets pour ce groupe/source

La couche transport

UDP et TCP

Couche transport

Notion de port

- La couche réseau permet d'identifier les machines
- La couche transport fournit la notion de **port** qui permet d'identifier les applications

Notion de *socket*

- Une *socket* est l'association d'une adresse IP et d'un numéro de port
- Elle identifie une application sur une machine
- La combinaison de deux *sockets* identifie une connexion TCP ou une communication UDP

Les ports de communication

Côté serveur

- Attend les connexions / datagrammes sur un numéro de port connu
- Les numéros de ports standards sont recensé par l'IANA
- On les retrouve également dans le fichier `/etc/services`

Côté client

- Le système fournit à l'application un numéro de port éphémère (> 1024)

Les protocoles

User Datagram Protocol (UDP)

- Un service sans connexion
- Non fiable mais simple

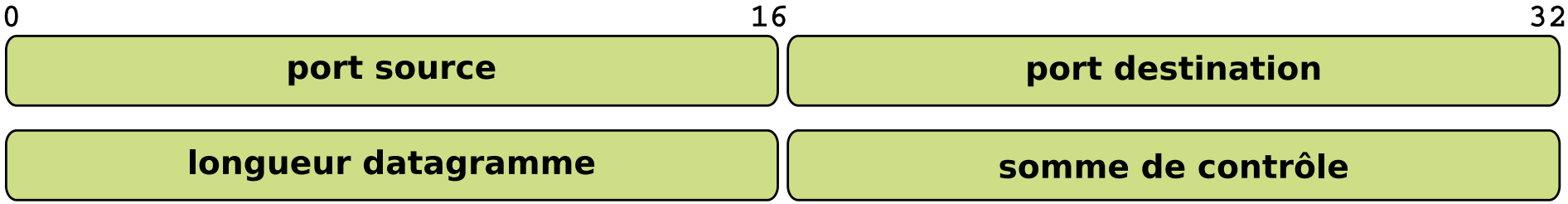
Transmission Control Protocol (TCP)

- Un service orienté connexion
- Fourni une communication fiable et régulée

User Datagram Protocol

- Un service minimum...
  - Pas de gestion des pertes de datagrammes
  - Les datagrammes peuvent arriver dans le désordre
- Mais pas inutile
  - Quand le message tient dans un datagramme (par ex. requête DNS)
  - Quand on ne veut pas subir un contrôle de flux (multimédia)
  - Pour les diffusions et le multicast

User Datagram Protocol



- La longueur du datagramme comprend l'en-tête et les données
- La somme de contrôle porte sur l'en-tête uniquement

Transmission Control Protocol

Un protocole fiable

- Les segments arrivent tous et dans l'ordre

Un protocole optimisé

- Attentif aux capacités du médium physique
- Attentif aux variations de vitesse de transmission
- Attentif à la capacité de réception du destinataire

Un protocole équitable

- Partage de la bande passante entre les différentes connexions

TCP : un protocole fiable

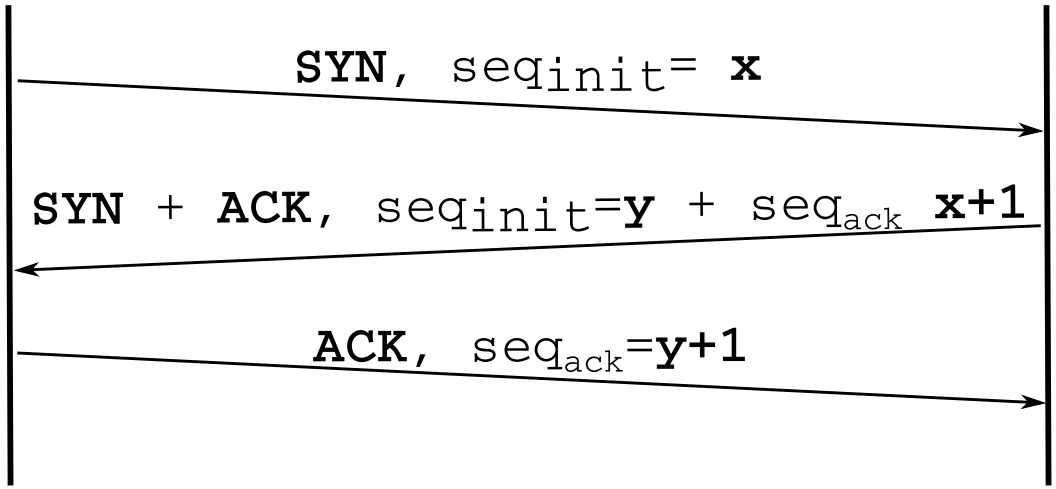
Les paquets arrivent tous

- Repose sur un mécanisme d'*acquittement*
- Chaque segment émis doit être acquitté. Dans le cas contraire, il est réémis.

Les paquets arrivent dans l'ordre

- Chaque segment possède un numéro de séquence
- Permet au destinataire de les ordonner (et de supprimer les doublons)

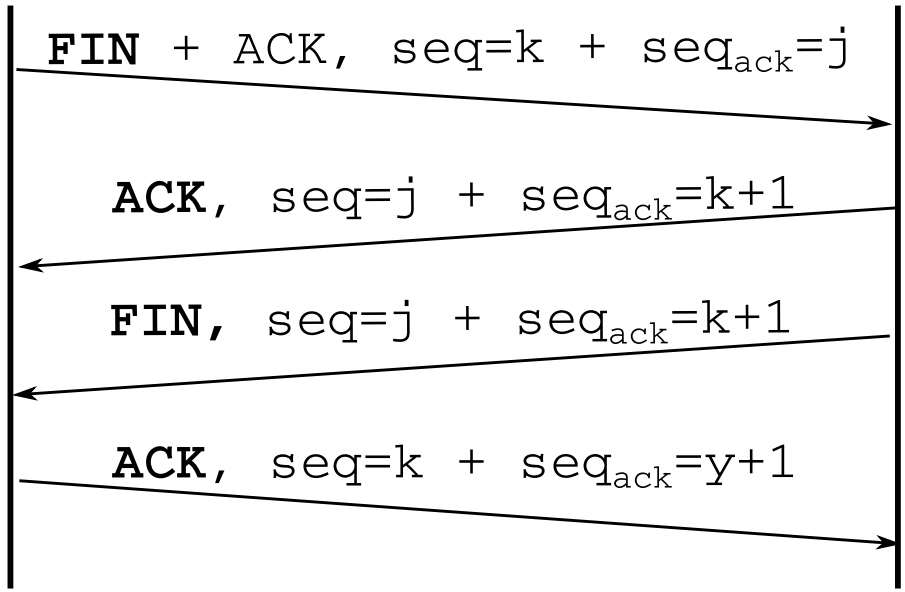
TCP : mise en connexion



Phase de mise en connexion

- Partager un état cohérent pour gérer la connexion
- Échange des numéros de segment initiaux
- Basé sur trois messages (*three-way handshake*)

TCP : fin de connexion



Phase de fin de connexion

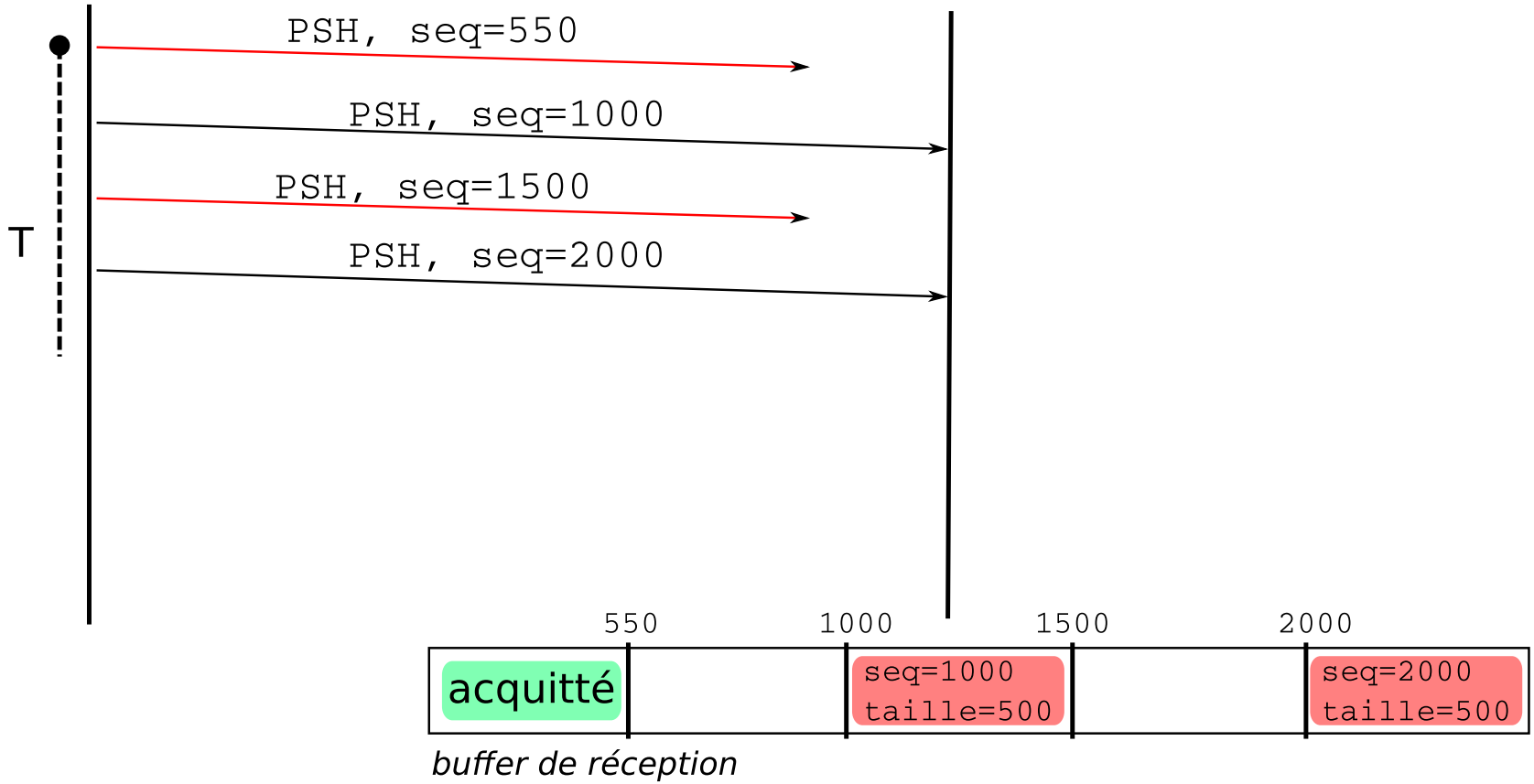
- Chaque partie doit clore sa connexion

TCP : échange de données

Fiabilité

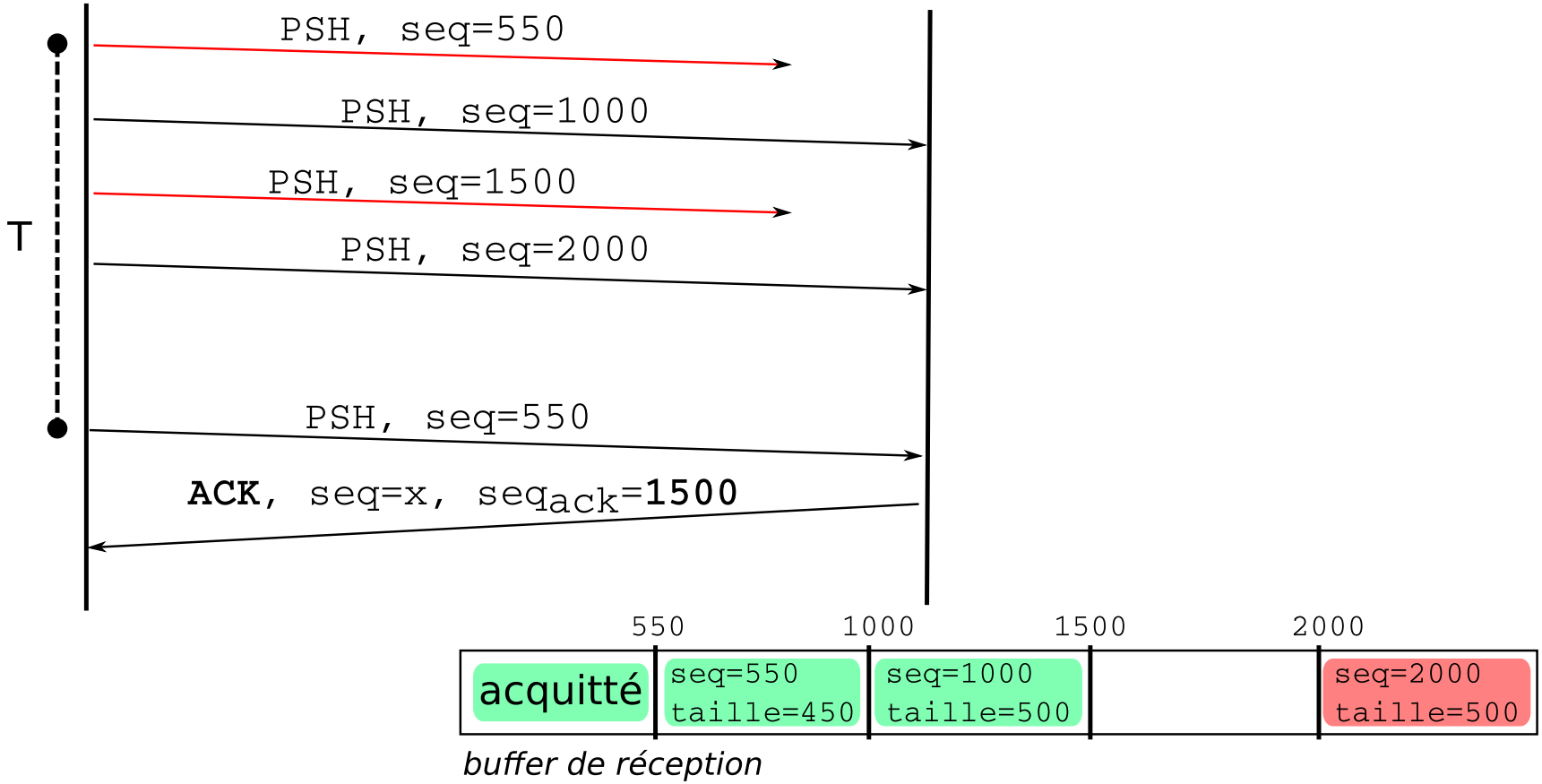
- La fiabilité est apportée par les acquittements
  - Un temporisateur est associé à chaque envoi
  - Si l'acquittement n'est pas reçu dans ce délai, on réémet
  - Doit prendre en compte le temps de trajet dans le réseau pour optimiser
- L'ordonnancement des données repose sur le numéros de séquence

TCP : échange de données



- Un temporisateur **T** est armé à chaque envoi
- Le numéro de séquence est un repère dans le flux (ordre et position)
- On n'acquitte que ce qui est bien reçu dans l'ordre

TCP : échange de données



- Quand le temporisateur tombe à 0, on réémet
- On acquitte l'ensemble des données reçues dans l'ordre

TCP : échange de données

Comment fixer la valeur du temporisateur ?

- L'émetteur calcule le temps d'aller-retour pour chaque segment (émission + réception acquittement)
- Ce temps d'aller-retour sert de base pour fixer le temporisateur

Méthode historique

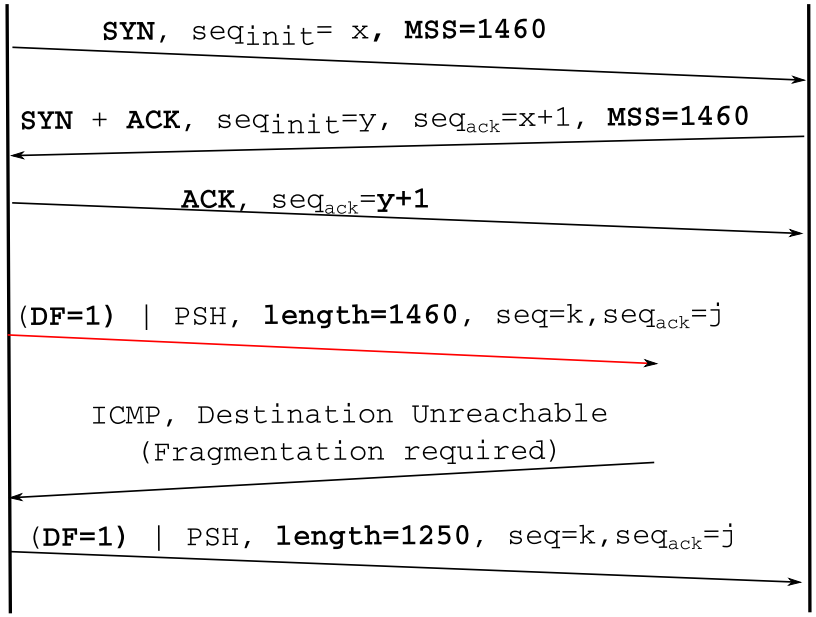
$$tb_t \leftarrow \alpha(tb_{t-1}) + (1-\alpha)tb_{mesure}$$

$$timer = min(ubound, max(lbound, \beta tb_t))$$

tb = temps de boucle, $0.8 \leq \alpha \leq 0.9$, $1.3 \leq \beta \leq 2.0$

$ubound$ = 1 minute, $lbound$ = 1 seconde

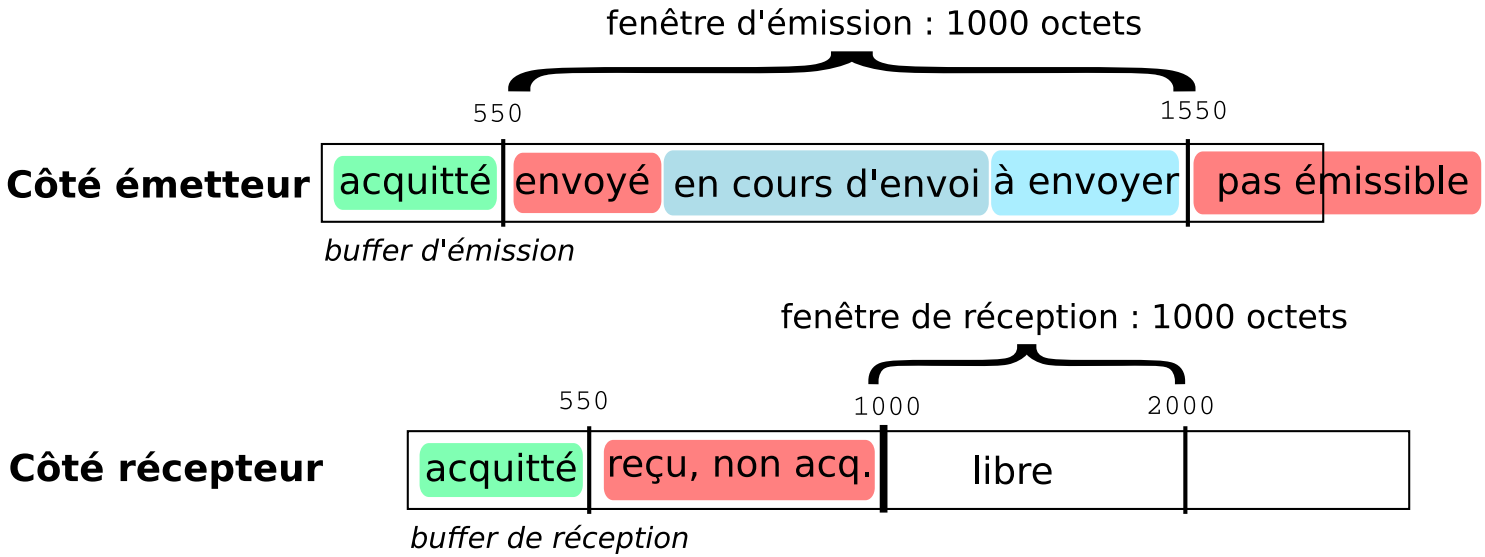
TCP : un protocole optimisé



Éviter la fragmentation

- Échange du MTU lors de l'établissement de connexion
- Mise en œuvre d'un protocole de découverte de MTU de chemin
  - Mettre le bit DF à 1 dans l'en-tête IP
  - Ajuster la taille des segments en cas d'erreur ICMP *Destination Unreachable - Fragmentation Required*

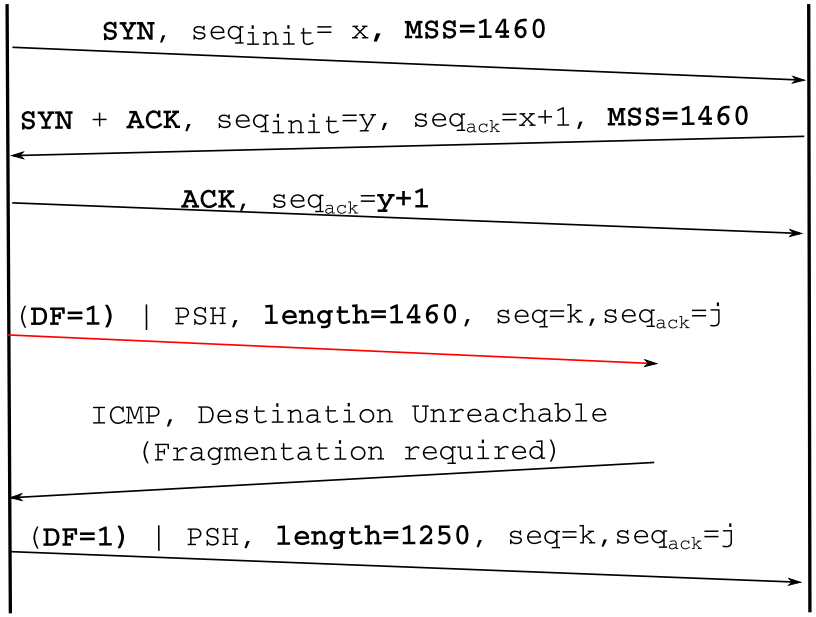
TCP : un protocole optimisé



Régulation du flux entre émetteur et récepteur

- Le récepteur indique dans les segments, la taille disponible dans sa fenêtre de réception
- L'émetteur ne peux pas envoyer plus de données
- La fenêtre se déplace au fur et à mesure des acquittements et traitements des données

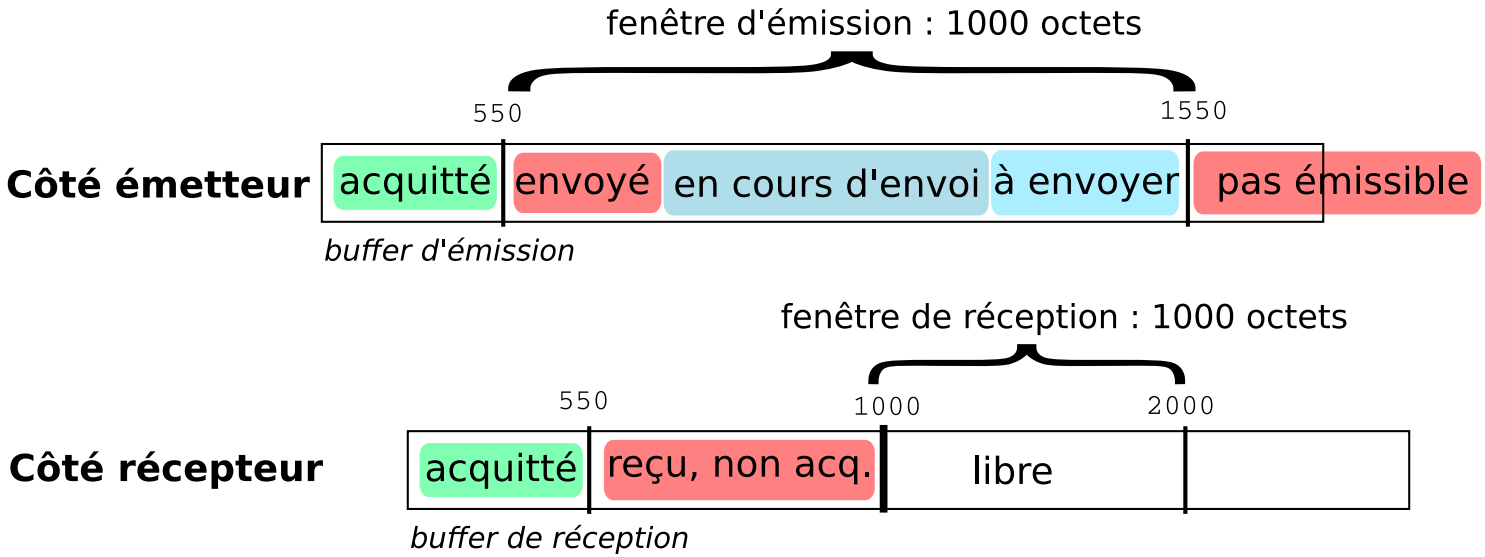
TCP : un protocole optimisé



Éviter la fragmentation

- Échange du MTU lors de l'établissement de connexion
- Mise en œuvre d'un protocole de découverte de MTU de chemin
  - Mettre le bit DF à 1 dans l'en-tête IP
  - Ajuster la taille des segments en cas d'erreur ICMP *Destination Unreachable - Fragmentation Required*

TCP : un protocole optimisé



Régulation du flux entre émetteur et récepteur

- Le récepteur indique dans les segments, la taille disponible dans sa fenêtre de réception
- L'émetteur ne peux pas envoyer plus de données
- La fenêtre se déplace au fur et à mesure des acquittements et traitements des données

TCP : un protocole équitable

Contrôle de congestion

- Éviter la surcharge des liens / routeurs
- Adapte le débit d'émission quand il détecte une congestion
- TCP conclut à une congestion en cas de perte de segment

Fenêtre d'émission

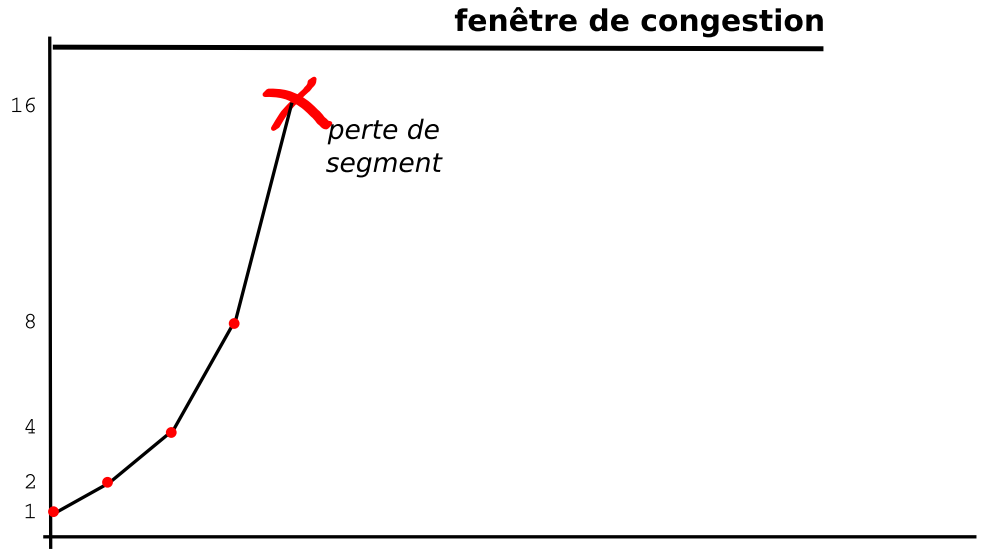
La fenêtre d'émission **W** correspond à la quantité de données qui peut être transmise sur le réseau

$$W = min(cwnd, awnd)$$

cwnd est la fenêtre de congestion évaluée par TCP

awnd est la fenêtre de réception du destinataire

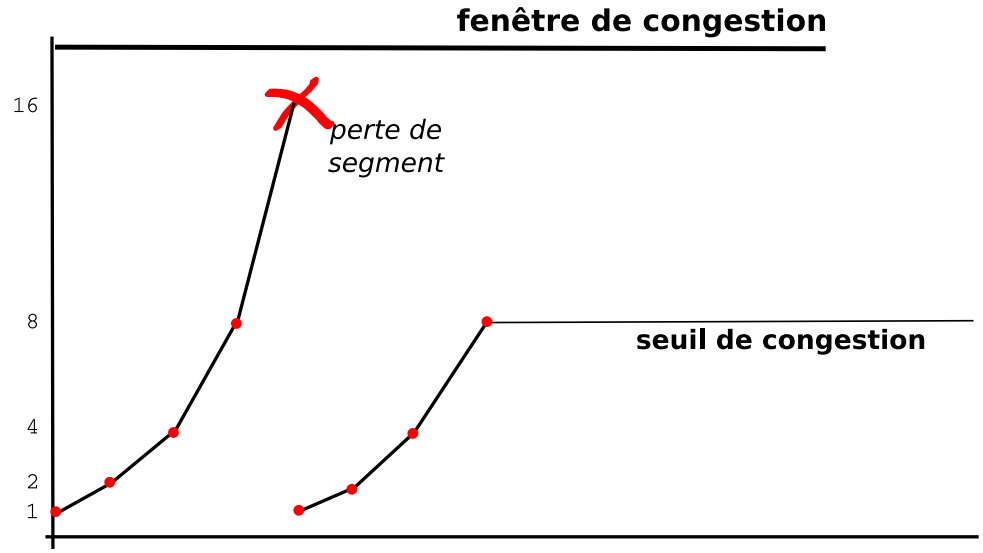
TCP : un protocole équitable



Démarrage lent

- Augmentation progressive du nombre de segments émis jusqu'à
  - Atteinte du seuil de congestion (cwnd) par une perte de segment
  - Ou atteinte de la fenêtre de réception
  - Cela détermine le seuil de congestion
- Le seuil de congestion est divisé par 2 en cas de perte de paquet

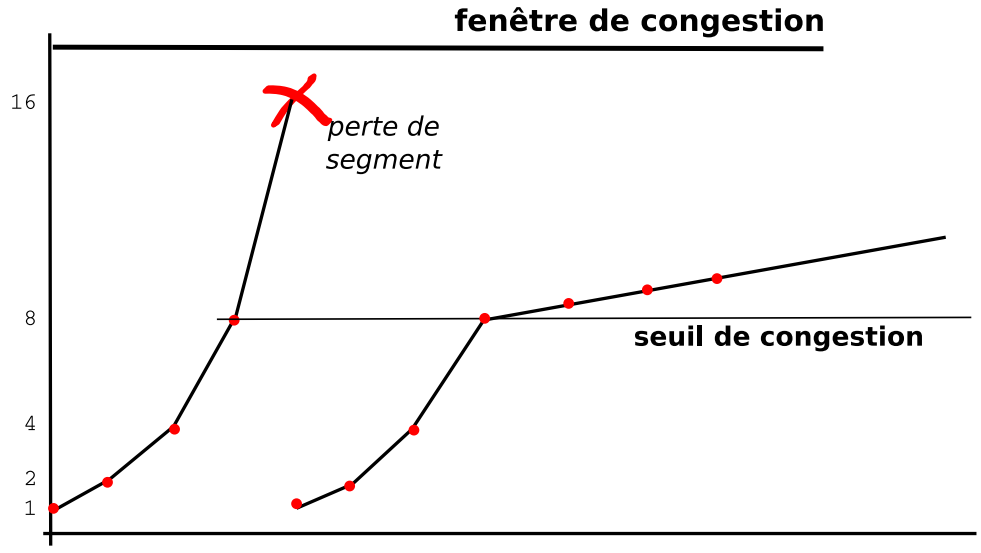
TCP : un protocole équitable



Démarrage lent

- Augmentation progressive du nombre de segments émis jusqu'à
  - Atteinte du seuil de congestion (cwnd) par une perte de segment
  - Ou atteinte de la fenêtre de réception
  - Cela détermine le seuil de congestion
- Le seuil de congestion est divisé par 2 en cas de perte de paquet

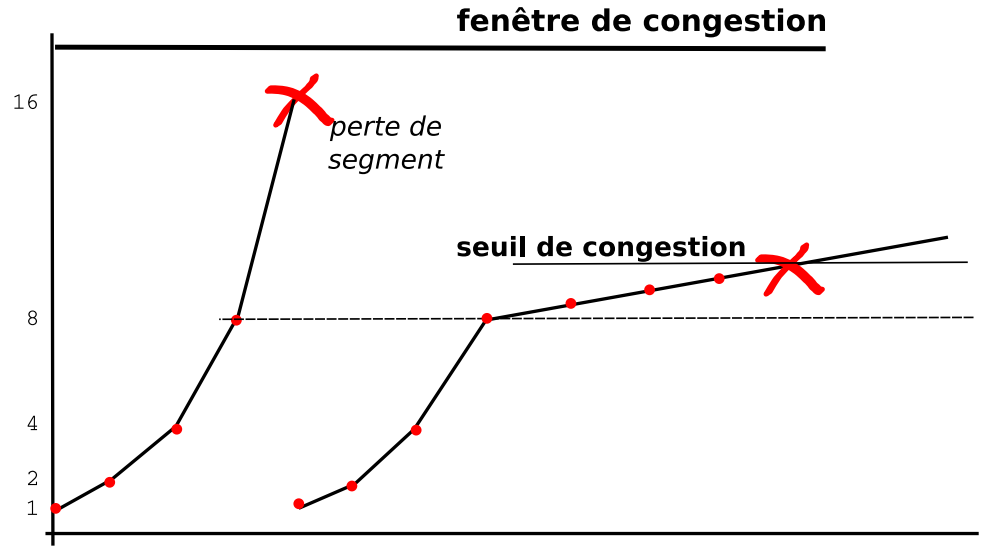
TCP : un protocole équitable



Évitement de congestion

- TCP recherche la fenêtre de congestion en augmentant le nombre de paquets de manière unitaire

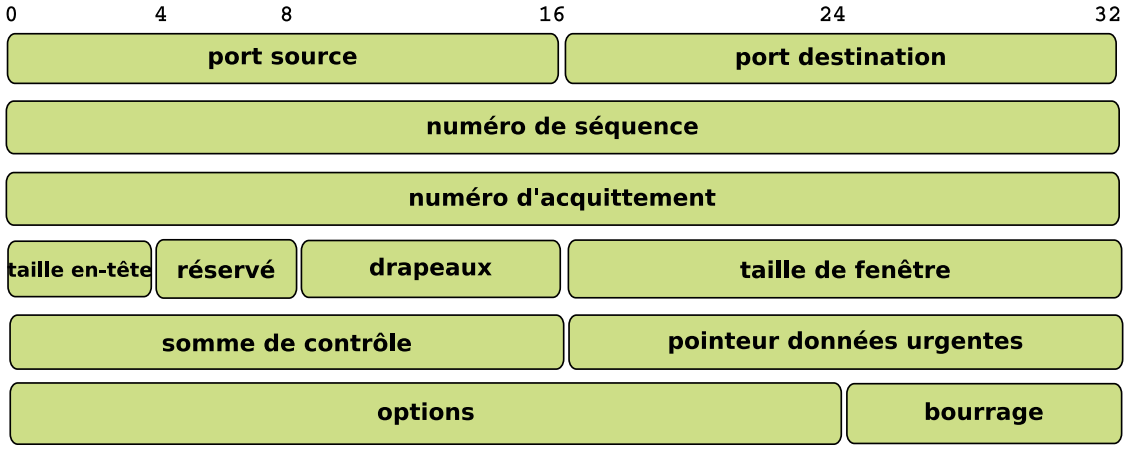
TCP : un protocole équitable



Évitement de congestion

- TCP recherche la fenêtre de congestion en augmentant le nombre de paquets de manière unitaire

TCP : en-tête



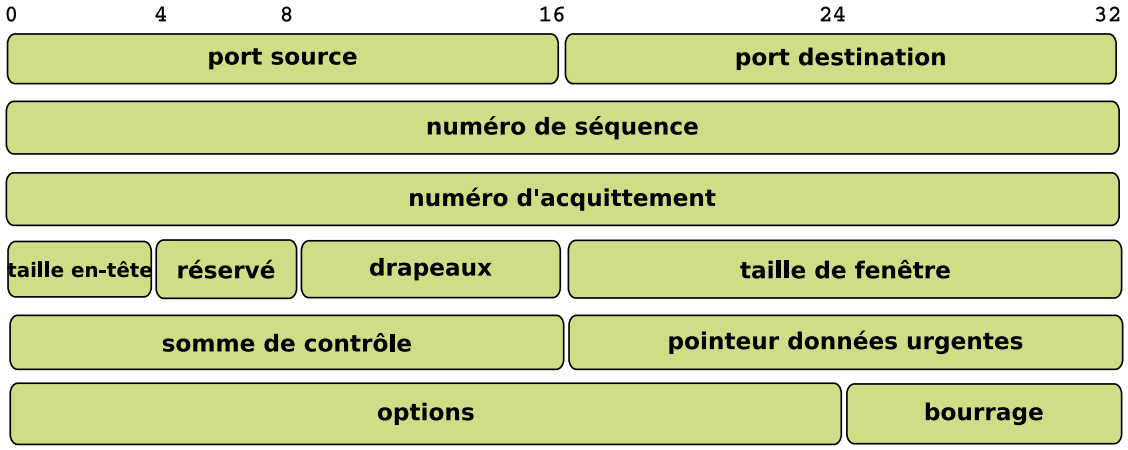
Numéros de port source et destination (16 + 16 bits)

Identifie les applications qui communiquent

Numéros de séquence et d'acquittement (32 + 32 bits)

Repères dans le flux et dans les données bien reçues

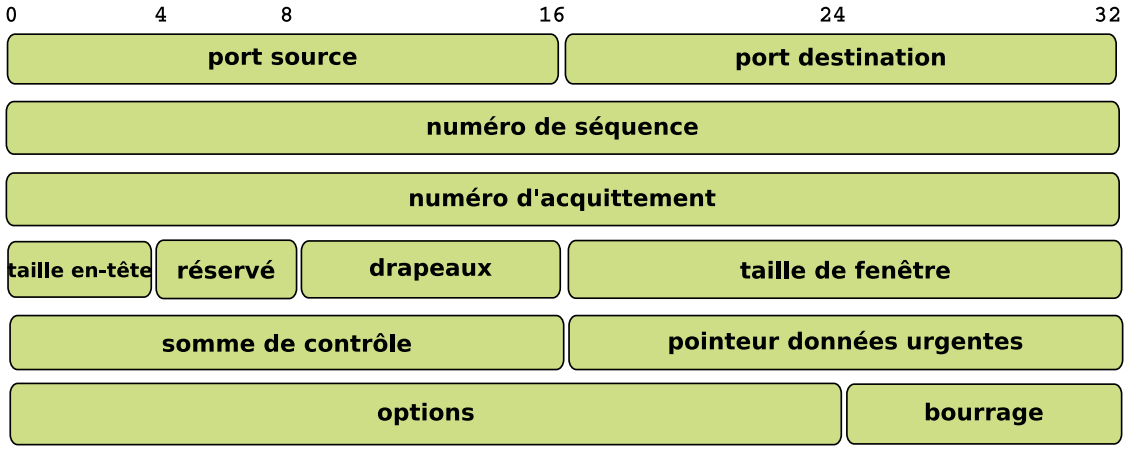
TCP : en-tête



Taille de l'en-tête (4 bits)

- longueur de l'en-tête en mots de 32 bits
- permet de savoir où commencent les données transportées par TCP
- sans options, on a une taille de 20 octets (5 mots)

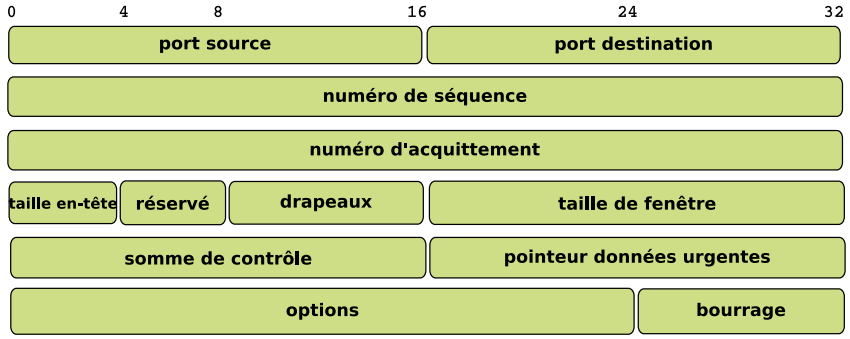
TCP : en-tête



Drapeaux (8 bits)

- donnent des informations sur le paquet ou la connexion
- les drapeaux “historiques” sont :
  - `URG` : indique des données urgentes à traiter en priorité
  - `ACK` : on acquitte des données (numéro d’acquittement)
  - `PSH` : contient des données à remonter à l’application
  - `RST` : abandon de connexion (*reset*)
  - `SYN` : mise en connexion. Annonce du numéro de séquence initial
  - `FIN` : fin de connexion

TCP : en-tête



Taille de fenêtre (16 bits)

Quantité de données acceptable par le destinataire. Permet de faire le contrôle de flux.

Somme de contrôle (16 bits)

Calculée sur l'en-tête TCP, les données et certains champs de l'en-tête IP

Pointeur de données urgente (16 bits)

Combiné avec le drapeau `URG`. Indique le déplacement par rapport au numéro de séquence courant pour aller lire les données urgentes.