Support de cours

Concepts Généraux des réseaux et architecture TCP/IP

Enseignant: Professeur Samuel OUYA

Présentation cours : Sommaire

- Cours :
  - Chapitre 1 : Topologie et classification des réseaux
  - Chapitre 2 : Le modèle OSI de l'ISO
  - Chapitre 3 : Les équipements d'interconnexion
  - Chapitre 4 : Le protocole Ethernet
  - Chapitre 5 : Couche 3, Les protocoles IP, ARP et ICMP
  - Chapitre 6 : Le routage IP
  - Chapitre 7 : Couche 4, Les protocoles TCP et UDP
  - Chapitre 8 : Les protocoles d'applications

C’est quoi étudier dans le domaine des réseaux?

Les compétences à acquérir par un apprenant dans le domaine de réseaux sont de 2 ordres :

1. Les compétences en connaissance et maîtrise des équipements réseaux :
   - Carte réseau
   - Hub
   - Switch
   - Routeur
   - Firewall
   - Proxy

2. Les compétences en connaissance et maîtrise des logiciels :
   - Les systèmes d’exploitation des ordinateurs(Linux, Windows, Mac OS
   - Les systèmes d’exploitation des équipements réseaux (IOS) :
     - Cisco
     - Juniper
     - Huawei
     - etc
   - Les couches protocolaires
   - Les applications réseaux (DNS, DHCP, WEB, Messagerie, Téléphonie sur IP, services d’annuaires LDAP, services de base de données, services d’authentification etc)

Présentation TD / TP

- Voir fiches de TD et TP

Que faut-il pour construire un réseau ?

- **Des équipements informatiques** : Ordinateurs, imprimantes, serveurs…

- **Des supports de transmissions** : Leur rôle est d'acheminer l'information d'un matériel à un autre en les reliant.

- **Des dispositifs d'interconnexion de réseau** : Pour réunir plusieurs réseaux ou de subdiviser le réseau en plusieurs réseaux.

- **Des systèmes d'exploitation réseau** : Dans certains cas on parlera de logiciel CLIENT et de logiciel SERVEUR.

- **Des protocoles** : l'hétérogénéité des matériels utilisés impose d'utiliser un certain nombre de règles.

A quoi sert fondamentalement un réseau informatique ?

Un réseau informatique qui ne fournit pas de services aux utilisateurs finaux n’a pas d’intérêt

Un service aux utilisateurs dans un réseau est fourni à travers un ensemble de programmes en informatique appelé **application** qui échange des informations à travers le réseau.

Un programme qui est en exécution est appelé en informatique **processus**.

Nous pouvons préciser qu’une application réseau n’est qu’un ensemble de processus qui échangent les informations à travers un réseau.

Chapitre 1 : Topologie et classification des réseaux

Objectifs spécifiques du chapitre 1 : Topologie et classification des réseaux

1. Comprendre deux principales compétences à développer quand on étudie dans le domaine des réseaux informatiques: les équipements et les logiciels
2. Comprendre fondamentalement ce à quoi un réseau informatique est destiné : le transport des messages des applications
3. Savoir identifier les besoins des différentes classes d'applications
4. Comprendre l'importance des procédures et règles dans un réseau
5. Être capable de classifier les réseaux selon leur taille
6. Comprendre les caractéristiques des supports de transmission
7. Être capable de donner les principales caractéristiques d'un système de transmission en terme de sens
8. Comprendre les deux types de commutation et les réseaux dans lesquels ils sont mis en œuvre.
9. Comprendre les caractéristiques des signaux et savoir faire la distinction entre la transmission en bande de base et la transmission par modulation.

Sommaire

1.1 Classification des réseaux

On peut classifier les réseaux selon leur taille en terme de zone de couverture de LAN, MAN et WAN comme le montre la figure 1.1 :

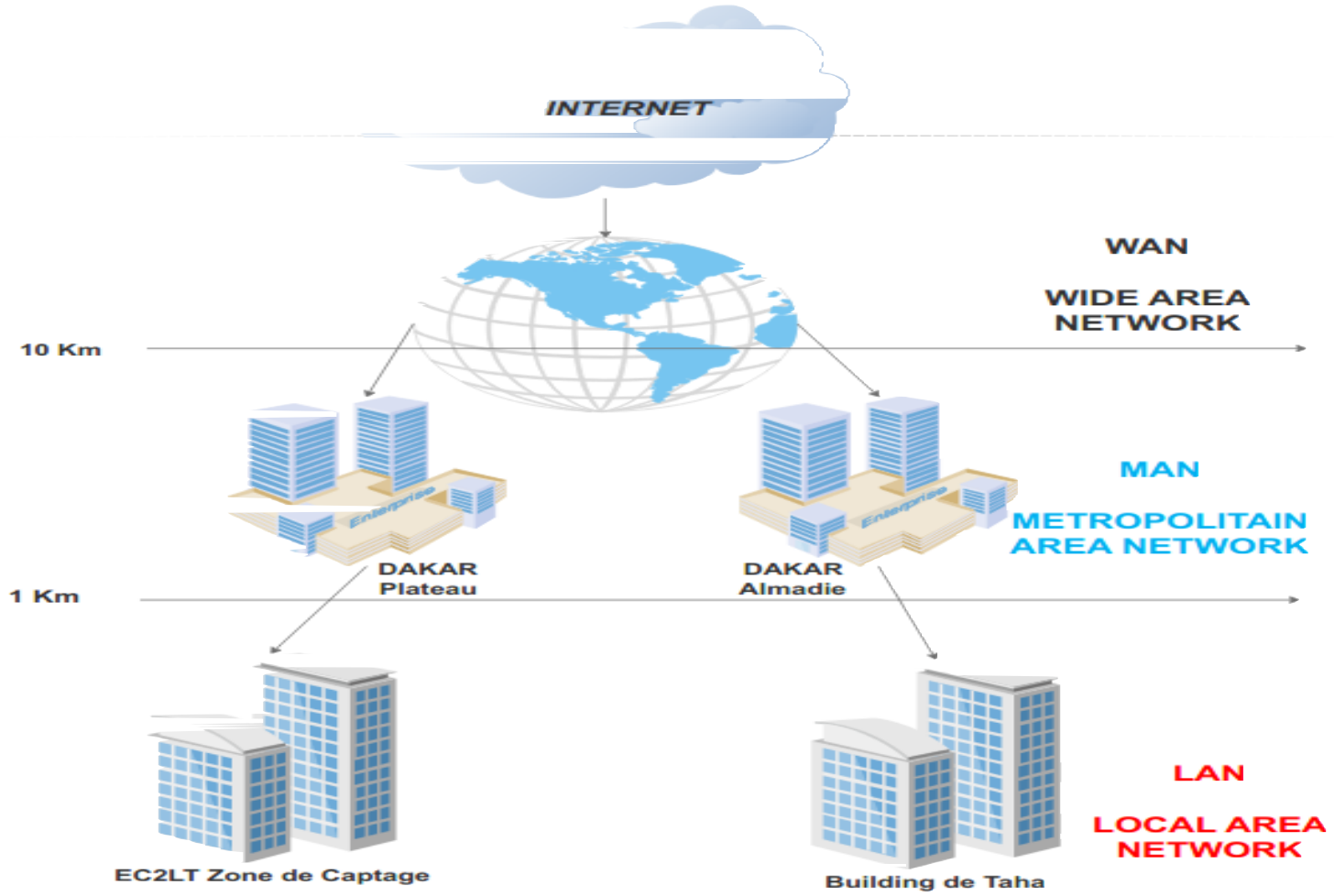


figure 1.1 classification des réseaux selon la taille

- Le réseau local ou **LAN** ou Local Area Network limité à une entreprise, une administration ou à un utilisateur privé

- Le réseau métropolitain ou **MAN** pour Metropolitain Area Network qui peut s'étendre à l'échelle d'un campus ou d'une ville

- Le réseau **WAN** ou Wide Area Network qui peut s'étendre à l'échelle d'un pays ou du monde

1.2 Caractéristiques d'un support de transmission

Les cinq principales caractéristiques d'un support de transmission peut être énumérées comme suit :

- **Débit maximal :** Nombre de bits/seconde pouvant être transporté sur le support. Dépend des caractéristiques physiques du matériau.

  Exemple :

  Une capture sur une machine Windows donne la figure 1.2 :

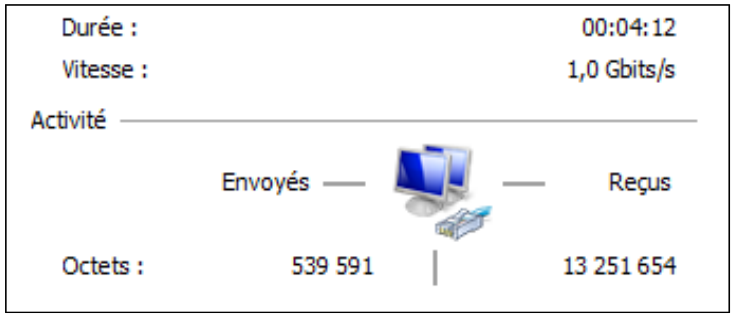


Figure 1.2 Débit sur une carte réseau sous Windows

Quel est le débit maximal de cette connexion au réseau local ?

Quel est le débit moyen montant et descendant sur la période d'utilisation ?

Quel a été le pourcentage d'utilisation de la ligne ?

- **Type du signal véhiculé :** Électrique, lumineux ou ondes électromagnétiques.
- **Atténuation :** En dB/m. Affaiblissement du signal le long de la ligne.
- **La sensibilité** aux perturbations électromagnétiques
- **Les coûts** : Fabrication et installation

1.2.1 Quelques exemples de supports de transmission

a) Câble à paires torsadées

- Provient du monde de la téléphonie. Les fils de cuivre des différentes paires sont isolés les uns des autres par du plastique et enfermés dans une gaine.

- Support de transmission constitué de 4 paires de fils. Une pour l'émission, une pour la réception, les deux dernières sont réservées aux commandes.

- Chaque paire est torsadée sur elle même, afin d'éviter les phénomènes de diaphonie (interférence entre conducteurs).

- **Caractéristiques:**
  - **Type de signal** véhiculé courant électrique
  - **Sensible** aux ondes électromagnétiques (si le câble n'est pas blaindé)
  - **Atténuation** : De l'ordre de 20dB/km
  - **Débit** : 100 / 1000 Mbps sur de courtes distances.
  - Pose très facile.
  - **Coût** : le moins cher du marché.

b) Fibre optique

Ce support de transmission est utilisé pour des liaisons longues distances. Il est **insensible aux perturbations électromagnétiques**.

- Type de signal : ondes lumineuses
- Atténuation : de l’ordre de 0.15 dB/km
- Vitesses de transmission très élevées
- Pose délicate (matériau rigide, angles de courbures importants)
- Coût élevé : Surtout pour les interconnexions optique-numérique.
- Poids au mètre faible (facteur important, aussi bien pour réduire le poids qu'exercent les installations complexes dans les bâtiments, que pour réduire la traction des longs câbles à leurs extrémités).

C) Les réseaux sans fil

On distingue plusieurs types de réseaux sans fil qu'on peut classifier selon la taille de zone de couverture en :

- **WPAN** : (Wireless Personal Area Network) :

  exemple :
  - **Bluetooth :** Faible portée / fort débit
- **WLAN** (Wireless Local Area Network)

  exemple :
  - **WIFI**
  - **Zigbee** : Faible débit / Low Power
- **WMAN** (Wireless Metropolitan Area Network), principalement destiné aux opérateurs de télécommunication

  exemple :
  - **WiMax**
- **WWAN** (Wireless Wide Area Network) R**éseau cellulaire mobile**.

  Exemple:
  - **GSM** (*Global System for Mobile Communication*).
  - **GPRS** (*General Packet Radio Service*).
  - **EDGE, 3G, 4G**

1.3 Topologie des réseaux

La topologie physique désigne la façon dont les ordinateurs sont interconnectés entres eux alors la topologie logique désigne l'interconnexion interne.

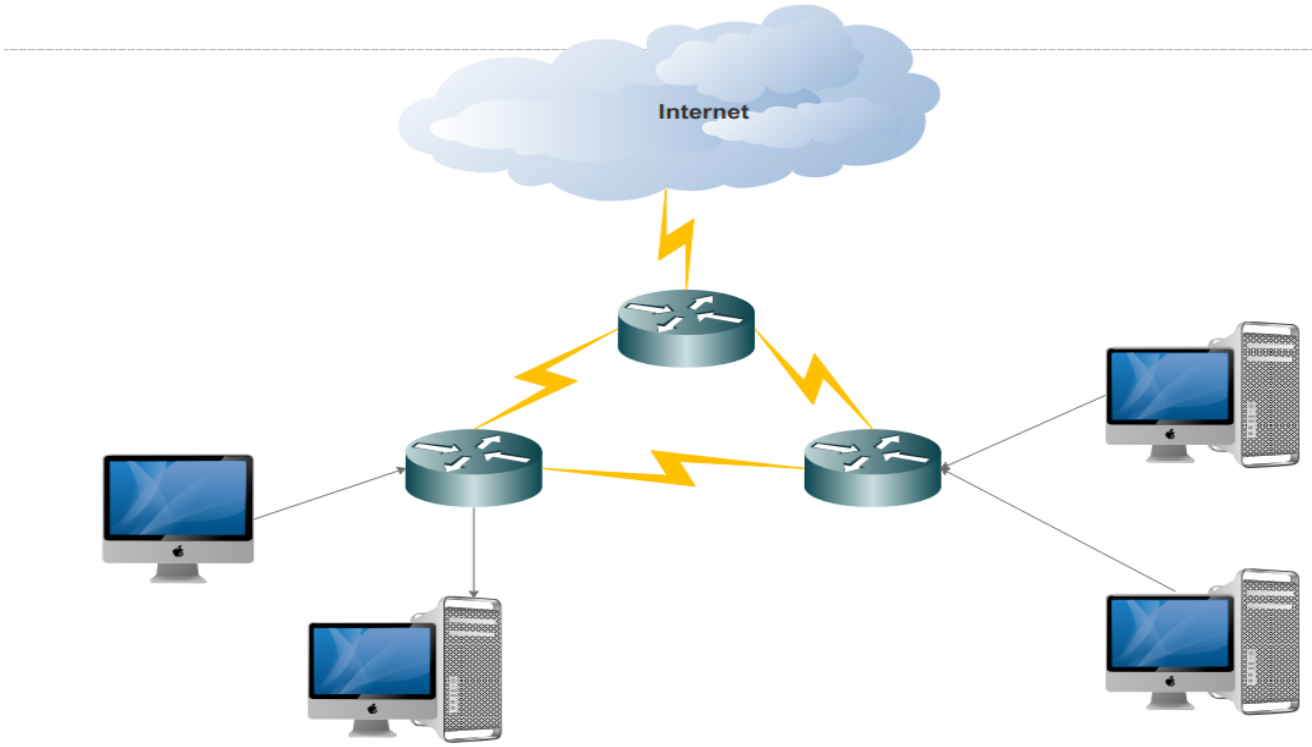


Figure 1.3 Topologie physique d'un réseau

1.4 Caractéristique d’une transmission

Dans le domaine des réseaux on note principalement 3 modes de transmissions en terme de sens de communication.

On dit qu’une transmission est en mode :

- **Full duplex** : lorsque l’échange bidirectionnel en même temps
- **Half duplex** : lorsque l’échange bidirectionnel mais alternativement
- **Simplex** : lorsque l’échange unidirectionnel

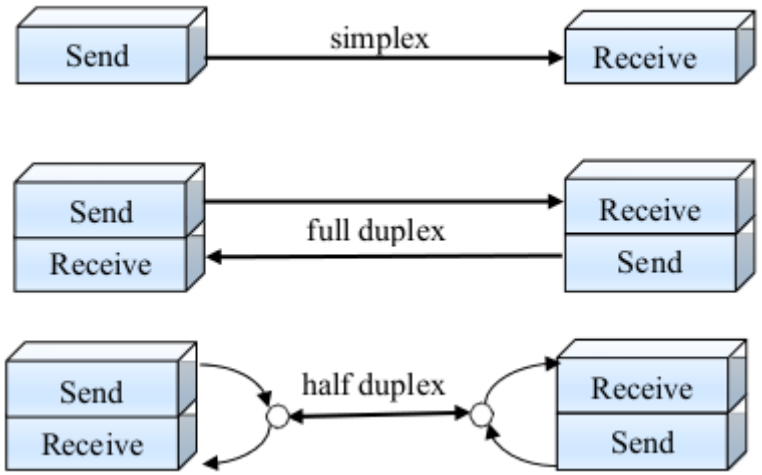


figure 1.4 Modes de transmission

Les types de commutations

Un réseau étant constitué de plusieurs nœuds interconnectés par des lignes de communication,il existe plusieurs méthodes permettant de transférer une donnée d'un nœud émetteur à un nœud dit récepteur c’est qu’on appelle la **commutation.**

On distingue globalement deux méthodes de transfert de données qui sont :

- **La commutation de circuits :**

  - La commutation de circuit est une méthode de transfert de données consistant à établir **un circuit dédié au sein d'un réseau**. La ligne est libéré seulement à la fin de la transmission.

  - Il s'agit notamment de la méthode utilisée dans le réseau téléphonique commuté (RTC). **Le lien physique est maintenu** jusqu’à la fin de l’appel et ne peut être perturbé (tonalité « occupé

- **La commutation de paquets**

  - Lors d'une transmission de données par **commutation de paquets,** les données à transmettre sont découpées en paquets de données(on parle de segmentation) et **émis indépendamment** sur le réseau.
  - Les nœuds du réseau sont libres de déterminer la route de chaque paquet individuellement. Les paquets ainsi émis peuvent **emprunter des routes différentes** et sont réassemblés à l'arrivée par le nœud destinataire. Il s'agit du mode de transfert utilisé sur internet, car il comporte les avantages d'être très tolérant aux pannes des nœuds intermédiaires (plusieurs chemins possibles).

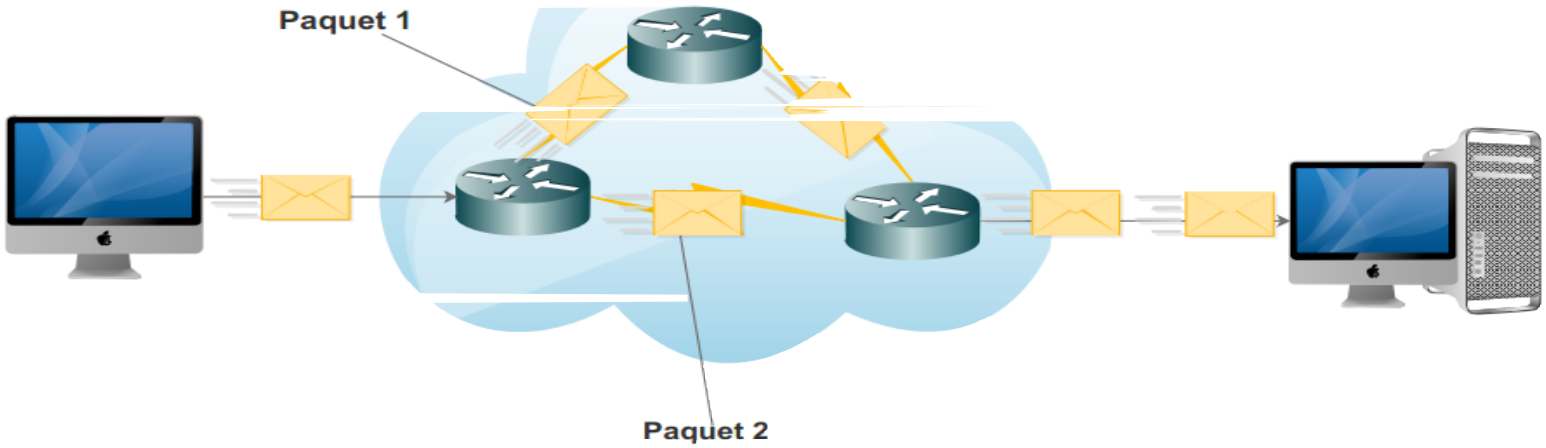


figure 1.5 La commutation de paquets

1.5 Caractéristiques des signaux

Devant un support de transmission il y a deux cas de figures :

Premier cas : Transmission en bande de base

Les caractéristiques du signal lui permet d'être transporté en tant que tel sur le support auquel cas le signal est codé et véhiculé sur le support.
On dit, dans ce cas, qu'on a fait une transmission en bande de base :
c'est le cas dans les réseaux IP câblés ou les bits sont codés en courant avant d'être transporté sur un câble à paires torsadées et lorsque le courant arrive au niveaux des cartes de réseaux du récepteur il est converti en bit pour que l'ordinateur puisse le comprendre.

On peut noter les i**nconvénients suivants en bande de base :**

- Monopolisation du support
- Dispersion du spectre (étalement du signal)
- Sensibilité aux perturbations

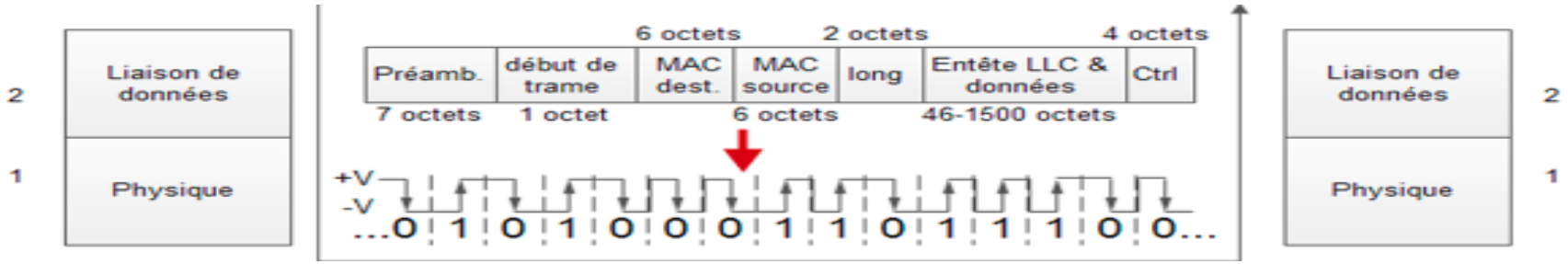


figure 1.7 Exemple de transmission en bande de base dans un réseau local

Deuxième cas : Transmission en bande transposé ou par modulation

Le canal de transmission utilisé en bande de base étant trop sensible aux perturbations sur de longues distances, on modifie les caractéristiques du signal pour que le signal modifié puisse être véhiculer sur le support quitte à ce que à la réception qu'on puisse trouver une technique de restitution du signal original.

On dit dans ce cas on dit qu'on a fait une transmission par modulation.

On utilise pour cela des équipements appelés modem (pour **mo**dulateur et **dém**odulateur)

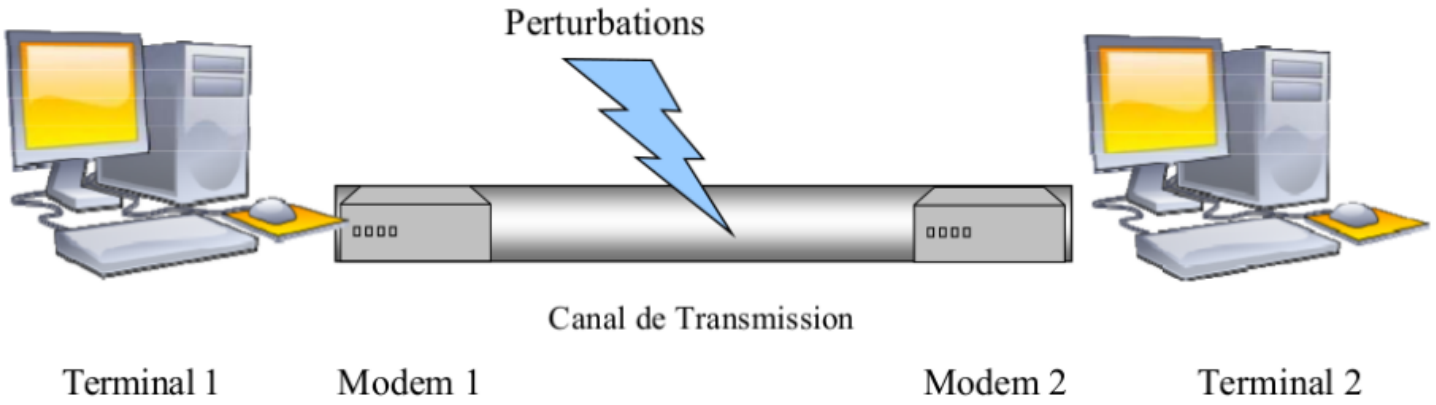


Figure 1.7 Utilisation de modems aux extrémités d'un canal de transmission

La transmission par modulation est utilisée sur des liaisons de type ADSL pour permettre aux abonnés fixes d'un opérateur de télécoms de se connecter à internet via les équipements distants de l'opérateur.

Sure la figure 1.8 on distingue la nature d’un signal avant sa modulation, il est transformé par le modem 1 à un signal analogique puis arrivé au niveau du modem 2 il reconstitue comme au départ .

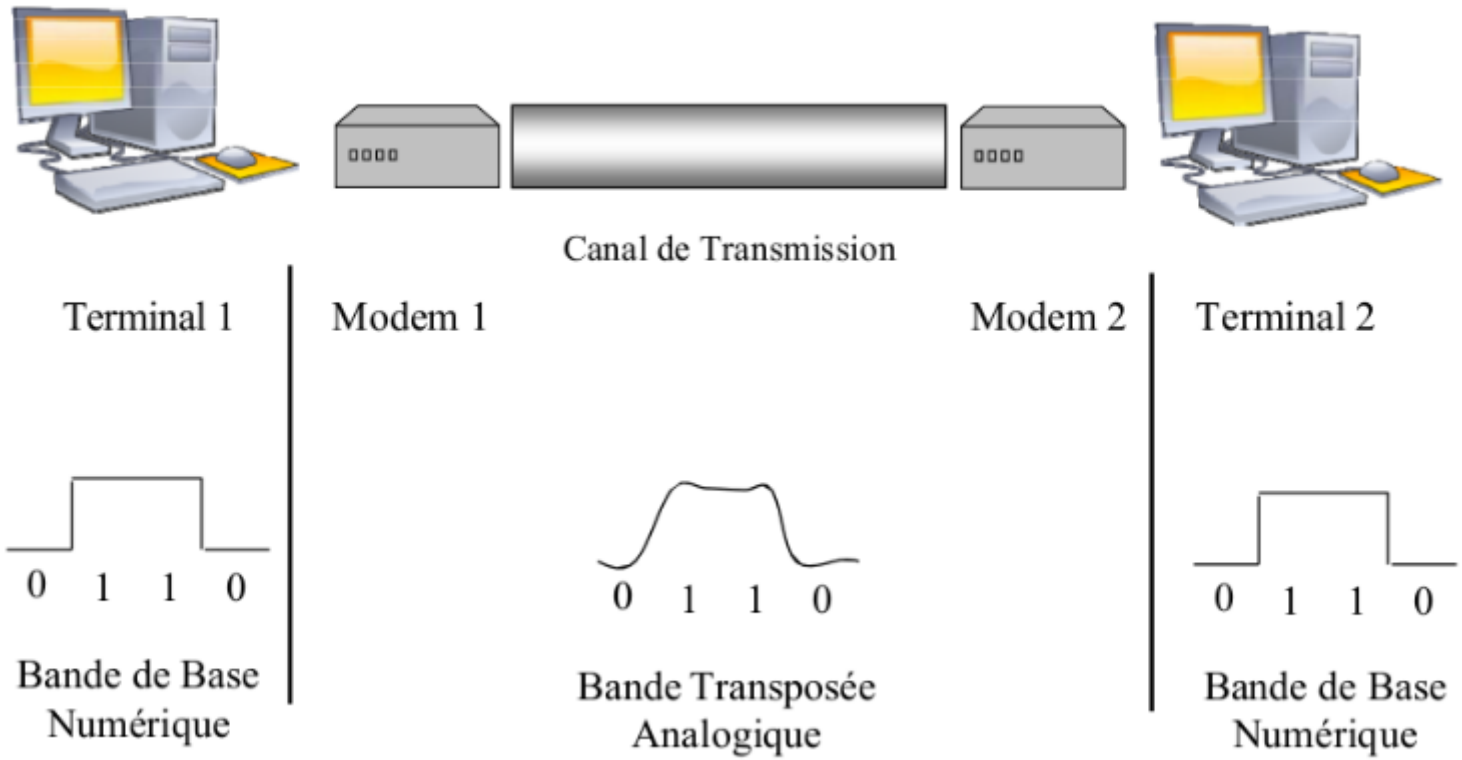


Figure 1.8 Modulation et démodulation d’un signal numérique

Chapitre 2 : Le modèle OSI de l’ISO

Objectifs spécifiques du chapitre 2 : Le modèle OSI de l’ISO

1. Comprendre le fonctionnement en couches du modèle OSI et de l'architecture TCP/IP

2. Pouvoir comparer les couches du modèle OSI et celles de TCP/ IP

3. Expliquer et comprendre le processus de génération de données à partir d'une machine recherchant à transmettre des données dans un réseau IP

4. Comprendre et expliquer le processus de désencapsulation des données à la réception

Sommaire

Dans un réseau informatique ou dans une société des hommes il est important de définir les règles à respecter et les règles de communication pour une bonne entente ou un bon fonctionnement c'est l'objet des protocoles ou des modèles d'organisations.

En réseau informatique on a défini deux modèles appelés OSI et TCP/IP.

Protocole

Les protocoles dans le domaine des réseaux sont, pour les ordinateurs, comparables aux langues que les hommes utilisent pour communiquer entre eux.

De manière précise un protocole : est **la façon dont sera organisée l'information** pour quelle soit compréhensible par deux entités distantes.

Exemple : Il existe des protocoles pour envoyer des e-mails, pour télécharger des fichiers, pour spécifier l'adresse du destinataire d'un paquet…etc.

Pour comprendre l'organisation en couche du modèle OSI on peut s'inspirer du cas d'un directeur qui voudrait envoyer un courrier à un collaborateur.

1. Le directeur écrit son courrier en utilisant un langage (français, anglais…)
2. La secrétaire met le courrier dans une enveloppe et inscrit l'adresse.
3. Le service postal va mettre la lettre dans un sac et inscrire l'adresse du centre de tri de destination.
4. **ACHEMINEMENT DU COURRIER**…
5. Le centre de tri de réception va ouvrir le sac et distribuer le courrier.
6. La secrétaire va récupérer le courrier et ouvrir la lettre
7. Le collaborateur va lire le courrier qui lui a été écrit par le directeur.

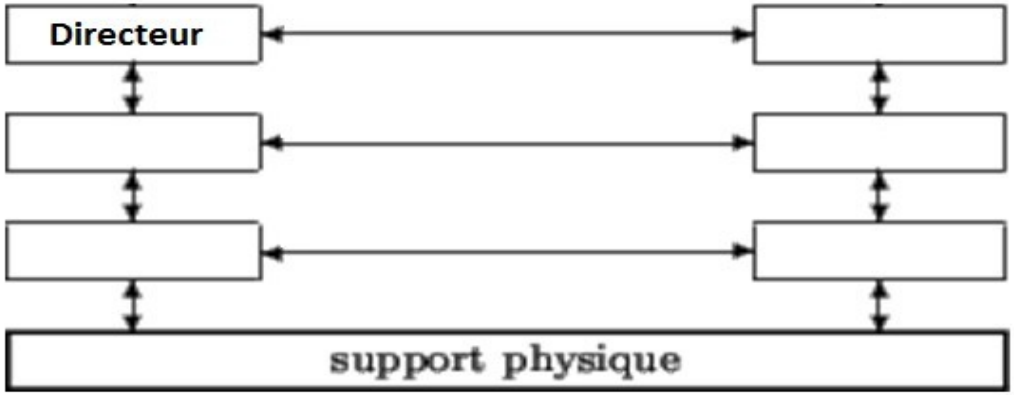


Figure 2.1 Type d'organisation pour la bonne remise d'un courrier

L'International Organization for Standardization **(ISO)** a défini un modèle de base appelé modèle **OSI**. Ce modèle définit 7 niveaux (couches) différents pour le transfert de données.

Chaque couche rend service aux couches adjacentes. Un protocole appartient à **une seule et unique** couche.

Ce modèle sur 7 couches n'est que très rarement utilisé.

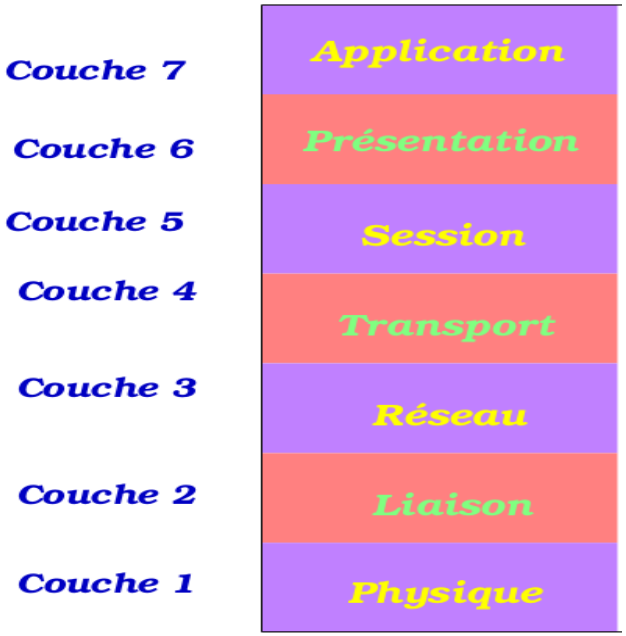


Figure 2.2 Les 7 couches du modèle OSI

Chaque couche du modèle OSI a un rôle bien déterminé dans le processus de communication entre une source et une destination :

- La couche **application** est la plus proche de l'utilisateur final. C'est elle qui sert d'interface entre les applications que nous utilisons pour communiquer et le réseau via lequel nos messages sont transmis. Les protocoles de couche application les plus connus sont notamment HTTP, FTP, IMAP et DNS.

- La couche **présentation** met en forme ou présente les données provenant du périphérique source dans un format compatible pour la réception par le périphérique de destination. Elle permet aussi de compresser les données de sorte que celles-ci puissent être décompressées par le périphérique de destination.

- **Couche session** comme leur nom l'indique, crée et gère les dialogues entre les applications source et de destination. La couche session traite l'échange des informations pour commencer et maintenir un dialogue et pour redémarrer les sessions interrompues ou inactives pendant une longue période.

- **La couche transport** prépare les données à transmettre sur le réseau. Un ordinateur source communique avec un ordinateur destinataire pour décider de la méthode de division des données en **segments**, de la méthode permettant de s'assurer qu'aucun des segments n'est perdu et de la méthode de vérification permettant de savoir si tous les segments sont arrivés. L’unité d’information de la couche transport est le **segment**.

- **La couche réseau** fournit des services permettant aux périphériques finaux(ordinateurs, serveurs, etc) d'échanger des données sur le réseau. Pour effectuer ce transport de bout en bout, la couche réseau utilise quatre processus de base :
  Adressage des périphériques finaux, Encapsulation, Routage et Désencapsulation.
  L'unité d'information de la couche réseau est le **paquet IP** ou **datagramme IP.**

- **La couche liaison de données** est responsable de l'échange des trames entre les nœuds via un support réseau physique. Elle permet aux couches supérieures d'accéder aux supports et contrôle la manière dont les données sont placées et reçues sur les supports. Plus précisément, la couche liaison de données assure ces deux services de base :

  - Elle accepte les paquets de couche 3 et les encapsule dans des unités de données appelées des trames.

  - Elle contrôle l'accès au support et détecte les erreurs.

- **La couche physique** fournit un moyen de transporter sur le support réseau les bits constituant une trame de couche liaison de données. Cette couche accepte une trame complète et la code sous la forme d'une série de signaux transmis sur les supports. Les bits codés composant une trame sont reçus par un périphérique final ou intermédiaire.

Le processus subi par les données, du nœud source au nœud de destination, est le suivant :

- Les données utilisateur sont segmentées par la couche transport, placées dans des paquets par la couche réseau, puis encapsulées sous forme de trames par la couche liaison de données.

- La couche physique code les trames et crée les signaux électriques, optiques, onde radio qui représentent les bits dans chaque trame.

- Ces signaux sont alors envoyés sur le support individuellement.

- La couche physique du nœud de destination récupère ces signaux individuels sur les supports, les convertit en représentations binaires et transmet les bits à la couche liaison de données sous forme de trame complète.

Il faut remarquer dans le modèle OSI les couches de l'émetteur communiquent avec leur homologue du récepteur comme le montre la figure 2.3 :

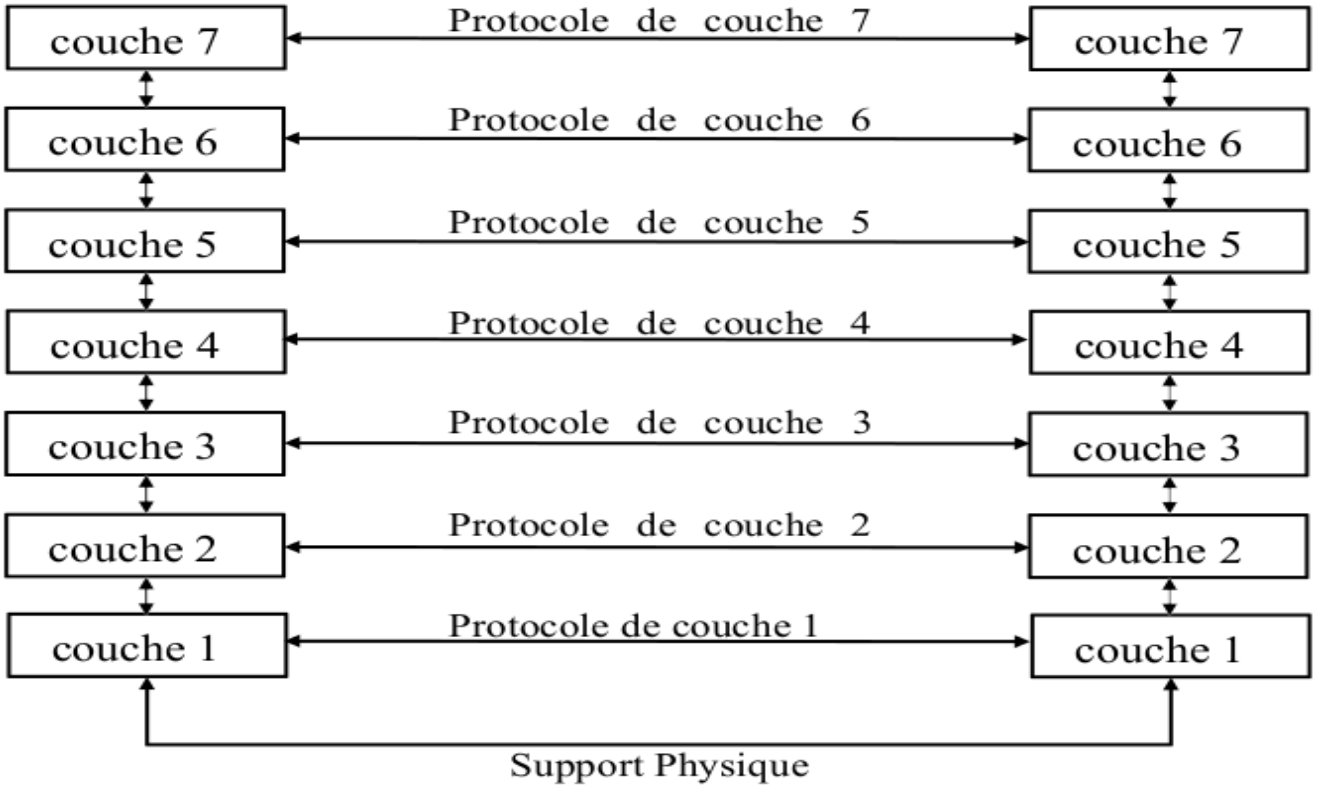


figure 2.3 Communication homologue à homologue

2.2 Le Standard TCP/IP

Le standard TCP/IP est le mode d'organisation qui est véritablement mise en œuvre dans un réseau IP.

Ce standard est structuré en quatre niveaux :

- **L'interface réseau physique** (couches 1 et 2 du modèle OSI) : dispositifs d'interconnexion et protocole Ethernet.

- **La couche réseau** (couche 3 du modèle OSI) : acheminer les paquets (routage) d'un ordinateur à un autre.

- L**e couche transport** (couche 4 du modèle OSI) : Assurer le transport et éventuellement le bon acheminement des paquets.
- **La couche application** (couches 5, 6 et 7 du modèle OSI) : Protocoles d'applications.

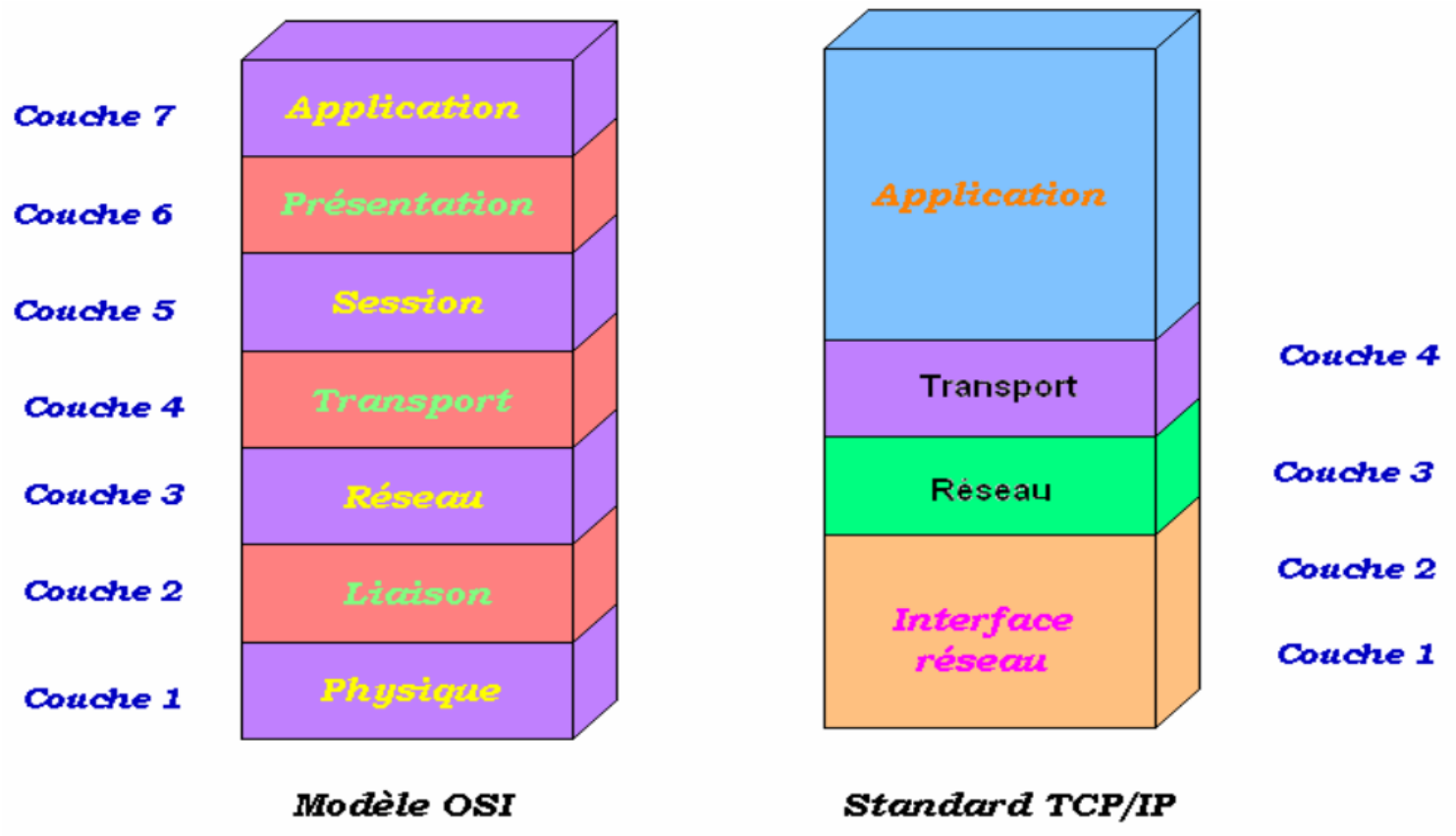


figure 2.4 Comparaison modèle OSI et standard TCP/IP

Il est important quand on étudie les réseaux

- de comprendre le processus de génération des données à transporter au niveau d'un émetteur appelé **encapsulation**
- de décrire le fonctionnement des protocoles à chaque couche
- de décrire le format des données manipulées par chaque couche
- de comprendre le processus de réstitution des données utiles au niveau du destinataire appelé **desencapsulation**

La figure 2.5 donne un exemple d'encapsulation des données

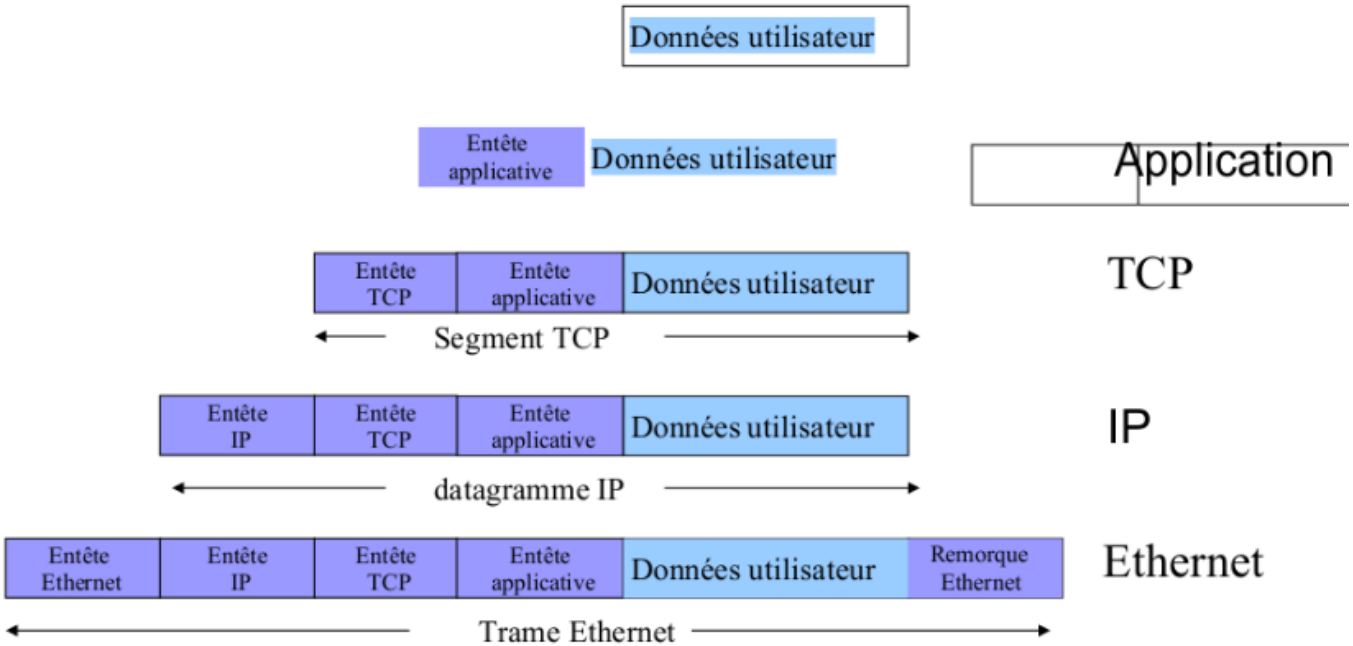


figure 2.5 Encapsulation des données et nom des données à chaque couche

Le sens de communication au niveau de l'émetteur se fait de la couche application vers la couche physique et on parle **d'encapsulation.**
Le sens de communication au niveau du récepteur se fait de la couche physique vers la couche application et on parle de **désencapsulation** comme le montre la figure ci-dessous :

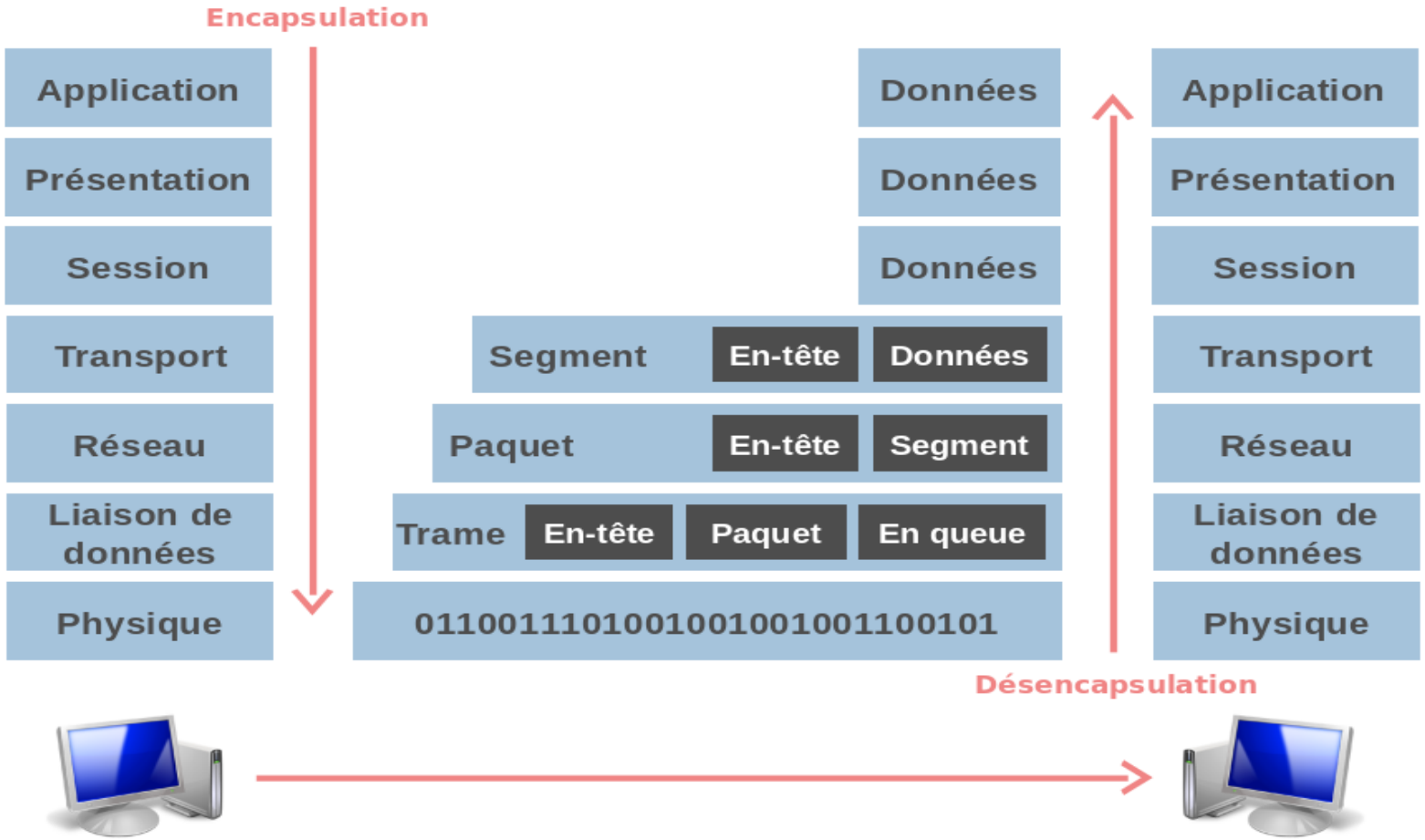


figure 2.6 Encapsulation des désencapsulation

Chapitre 3 : Les éléments d’interconnexion

Objectifs spécifiques du chapitre 3 : Les équipements d’interconnexion

1. Comprendre le fonctionnement des équipements d'interconnexion

2. Situer le niveau d'opération des équipements par rapport au modèle de référence OSI.

Sommaire

Dans une formation en réseau une importance est accordée à la maîtrise des équipements.
C’est l’objet de ce chapitre qui introduit les principaux équipements réseaux d’interconnexion

On ne peut parler de réseau sans une carte réseau qui permet de convertir les signaux venant de support de transmission et allant vers un ordinateur ou qui convertit les signaux sortant de l’ordinateur et allant vers le support de transmission.

Nos machines d’aujourd’hui sont le plus souvent dotées de types de cartes réseaux :

- Carte Ethernet pour les réseaux câblés
- Carte wifi pour l’accès aux réseaux sans fil

Le processus de conversion de bit en courant électrique dans les réseaux à câble est appelé **codage en ligne.**

De même le processus de conversion des bits en ondes radio dans des réseaux sans fil tel que le wifi est aussi appelé **codage en ligne.**

3.1 Le répéteur

Le répéteur permet de régénérer le signal d'un même réseau.

Il fonctionne au niveau de la couche 1 du modèle OSI.

Fonctions d'un répéteur :

- **Répéter** les blocs d'informations d'un segment à l'autre.
- **Régénérer** le signal pour compenser l'affaiblissement.

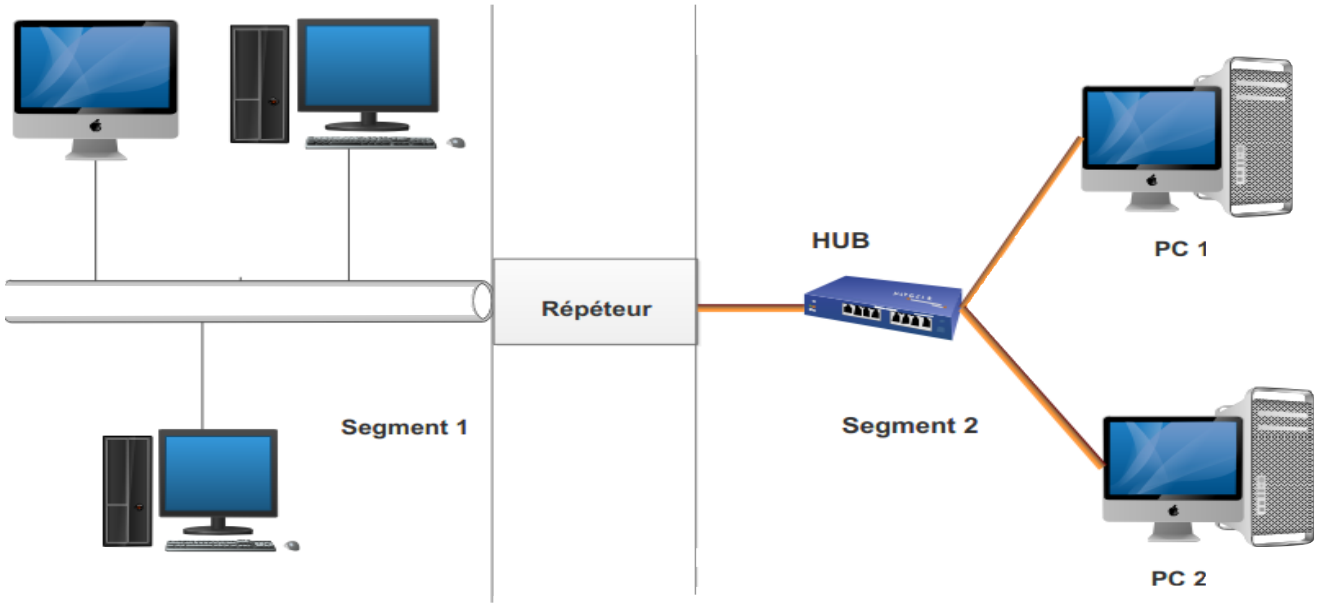


Figure 3.1 L'interconnexion de 2 partie d'un réseau par un répéteur

Un HUB est un répéteur à plusieurs ports qui permet de connecter plusieurs machines entre elles.

Fonctions d'un HUB :

- **Répéter** de bloc d'informations d'un segment à l'autre.
- **Régénérer** le signal pour compenser l'affaiblissement.
- **Concentrer** plusieurs lignes en une seule.

L'inconvénient est de partager le débit du réseau concerné.

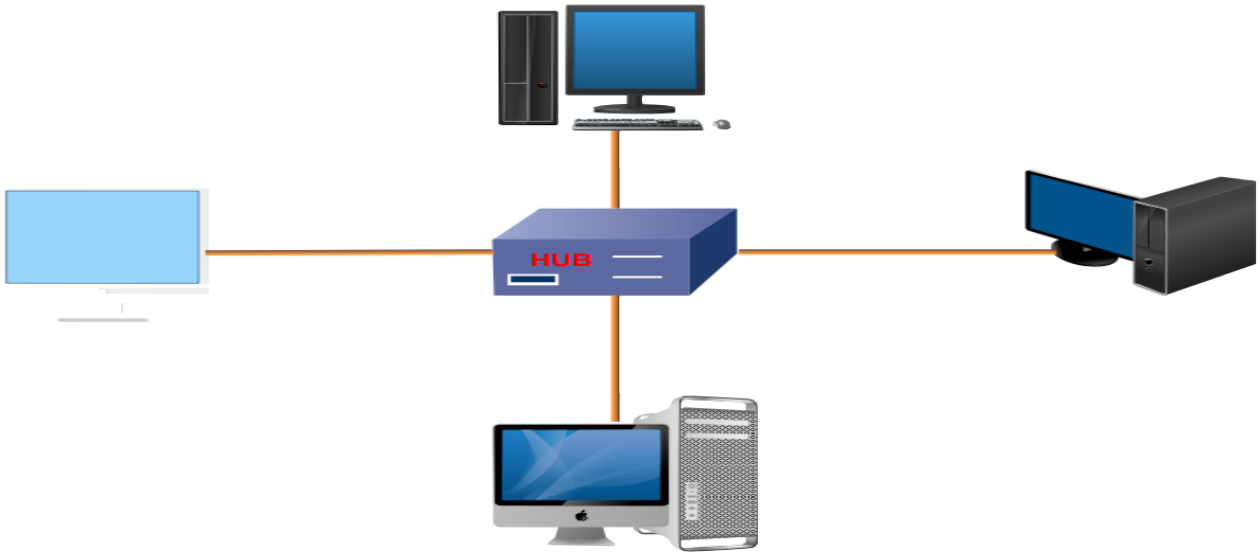


Figure 3.2 L'interconnexion des machines par un hub

Le répéteur et hub fonctionnent sont des équipements de la couche physique du modèle OSI comme représenté sur la figure 3.3 :

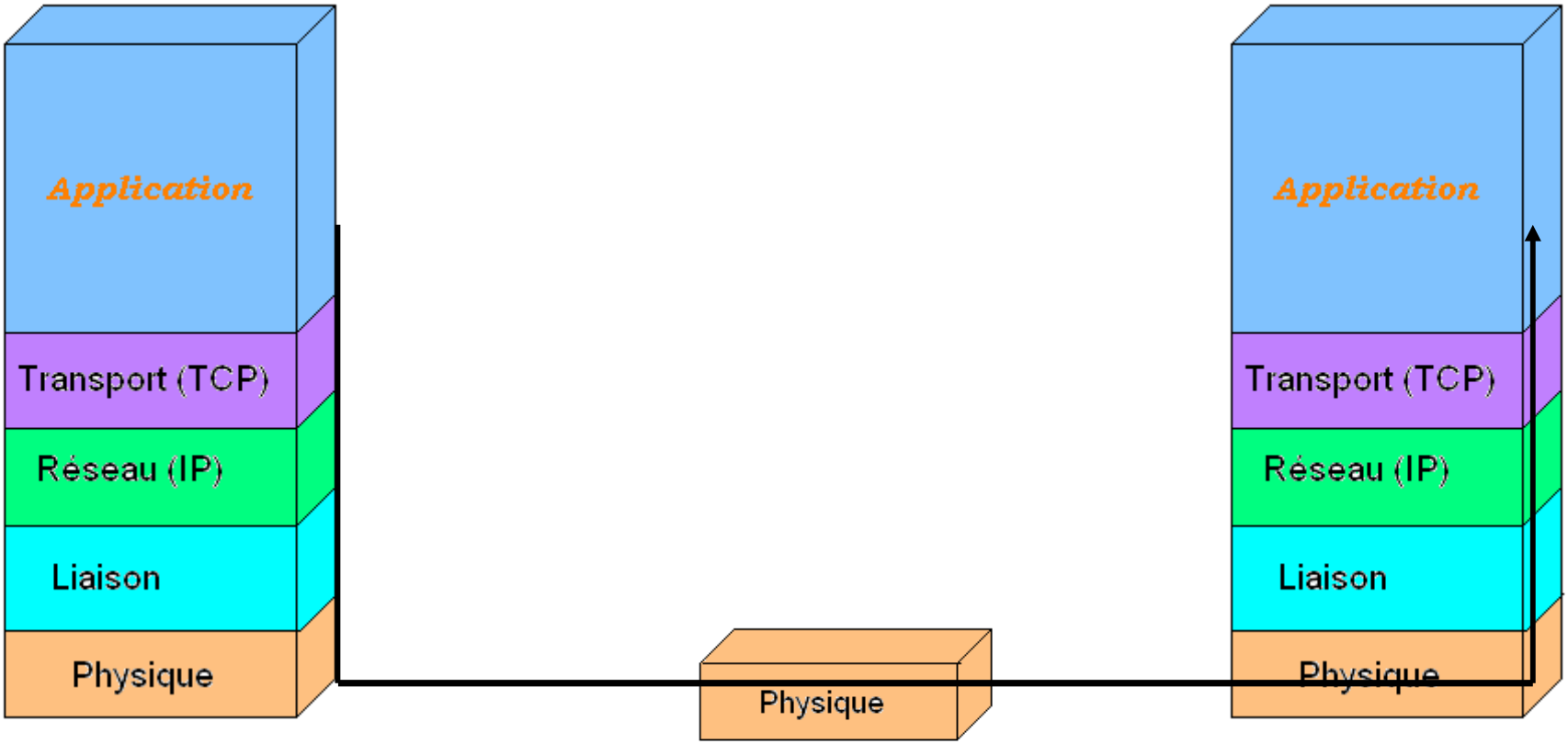


figure 3.3 Niveau de fonctionnement d'un répéteur et d'un Hub

3.2 Pont et Switch

Un PONT a pour

- **Fonctions :**
  - **Reconnaître** les adresses physiques des informations qui transitent.
  - **Filtrer** et laisser passer seulement l'information destiné au réseau raccordé.
  - **Assurer** l'interconnexion de stations ou de segments d'un LAN en leur attribuant **l'intégralité** de la bande passante. Le débit disponible **n'est plus partagé** entre tous les utilisateurs.

Il analyse l'entête de niveau 2 (adresse physique)

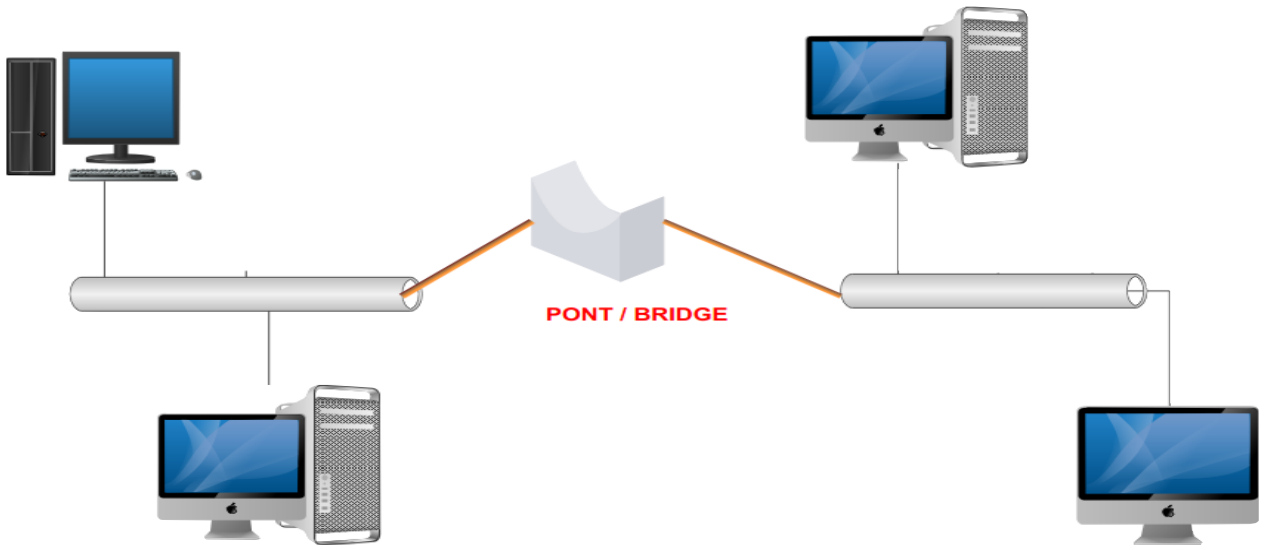


figure 3.4 Interconnexion de 2 segments d'un réseau par un pont

Un switch est un pont à plusieurs ports et fonction au niveau de la couche liaison de données du modèle OSI comme le montre la figure 3.5.

Ainis les ponts et switchs manipulent les **trames** contrairement au répéteur et aux hubs qui manipulent des **bits**.

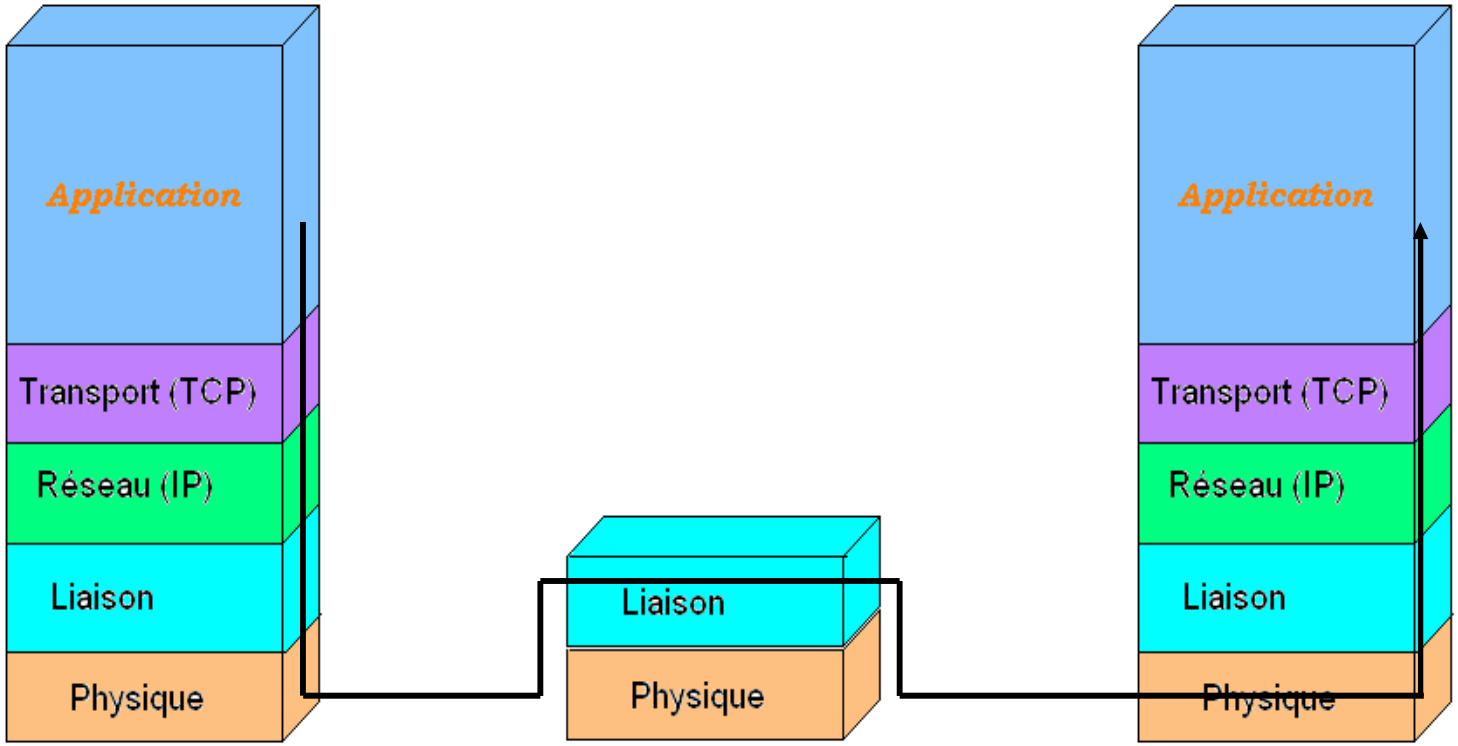


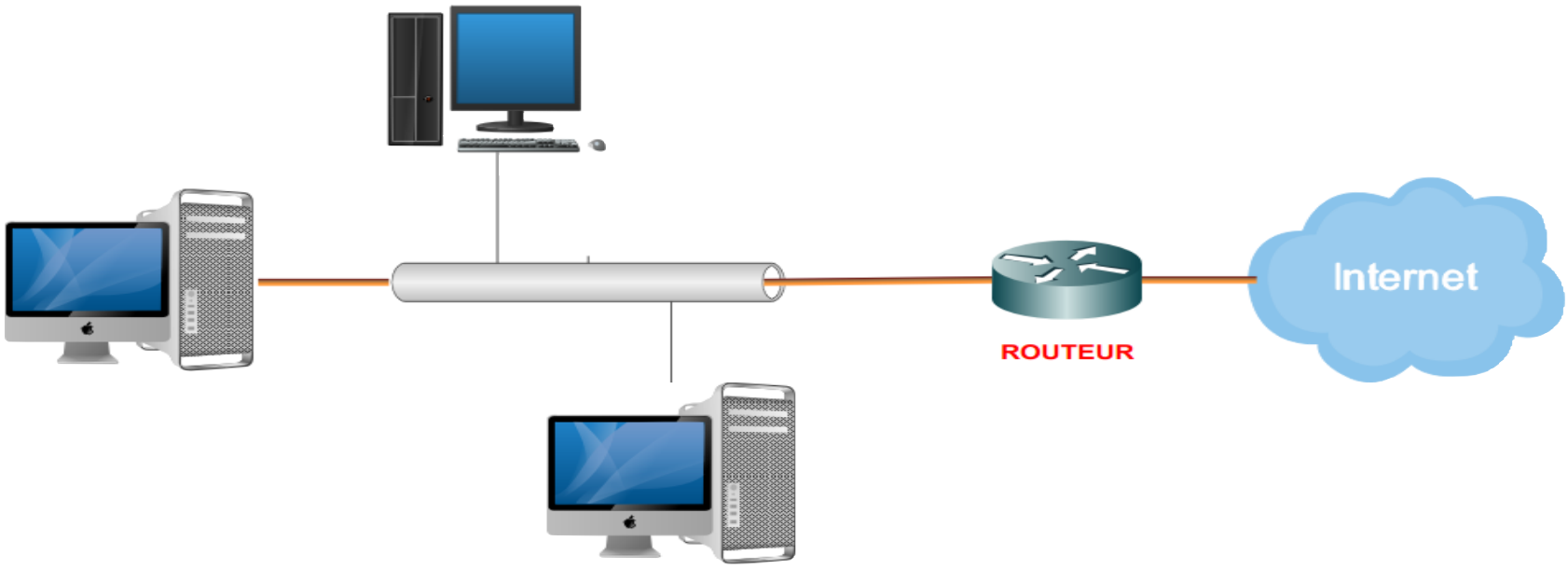
figure 3.5 Niveau de fonctionnement d'un pont et d'un switch

3.3 Le routeur

Le routeur permet de relier de nombreux réseaux locaux de telle façon à permettre la circulation des datagramme IP d'un réseau à un autre.

Fonctions d’un routeur:

- **Analyser** et de **choisir** un chemin à travers le réseau pour véhiculer les paquets sur le réseau
- **fragmenter** si nécessaire un datagramme IP pour respecter la taille maximale des données supportée par le réseau sur lequel le datagramme est transféré.



figurer 3.6 L’interconnexion de 2 réseaux par un routeur

Le routeur fonctionne au niveau 3 du modèle OSI et par conséquent manipule les datagramme IP comme le montre la figure 3.7.

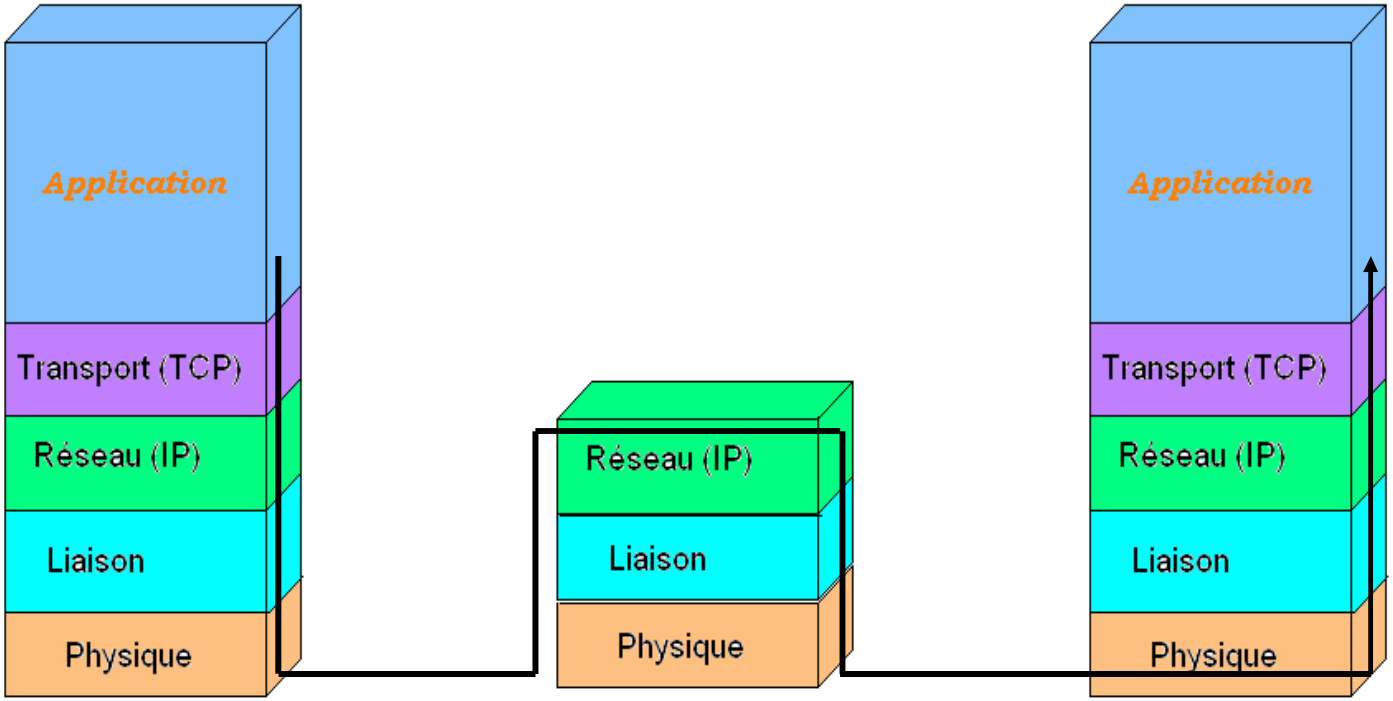


figure 3.7 Niveau de fonctionnement d’un routeur

Chapitre 4 : Couche 2, le protocole Ethernet

Objectifs spécifiques du chapitre 4 : Le protocole Ethernet

1. Savoir faire la distinction entre Ethernet partagé et Ethernet commuté
2. Savoir décrire le format d'une trame Ethernet 1
3. Savoir décrire le format d'une trame Ethernet 2
4. Comprendre l'algorithme de backoff utilisé dans Ethernet partagé
5. Comprendre le fonctionnement des commutateurs en terme génération et de gestion des tables de commutation.
6. Identifier les numéros des deux fils utilisés pour la transmission et les deux fils utilisés pour la réception dans un câble à paire torsadée
7. Faire la distinction entre la méthode d'accès Ethernet 1 et 2
8. Comprendre les différents types de supports de transmission utilisés en Ethernet et leur nomenclature

L'objectif de ce chapitre est d'étudier l'une des deux normes les plus importantes des réseaux locaux.

C'est deux principales normes sont:

- La norme dite **802.3 ou Ethernet** déployé dans les réseaux câbles
- La norme **802.11** dont le nom commercial est le **WIFI** pour Wireless Fidelity.

NB: En général le but de l'étude d'une norme de réseau est de:

- d'étudier les équipements à utiliser pour déployer un réseau respectant cette norme
- d'étudier les supports de transmission utilisés par un réseau respectant cette norme
- d'étudier aussi les formats de données manipulées dans cette norme.
- d'étudier les méthodes d'accès ou les règles d'accès aux supports de transmission

La couche de liaison du modèle OSI a été divisé en 2 parties:

- Une sous-couche LLC (norme 802.2) qui est chargée d’effectuer directement des contrôles sans le concours des couches supérieures

- Une sous-couche MAC (norme 802.3) qui définit la méthode d’accès au support

Ethernet partagé :

- Tous les ordinateurs d'un réseau Ethernet sont reliés à une même ligne de transmission
- toute machine est autorisée à émettre sur la ligne à n'importe quel moment et sans notion de priorité entre les machines

La trame Ethernet

La couche de liaison du modèle OSI a été divisé en 2 parties :

- Une sous-couche LLC (norme 802.2) qui est chargée d'effectuer directement des contrôles sans le concours des couches supérieures
- Une sous-couche MAC (norme 802.3) qui définit la méthode d'accès au support

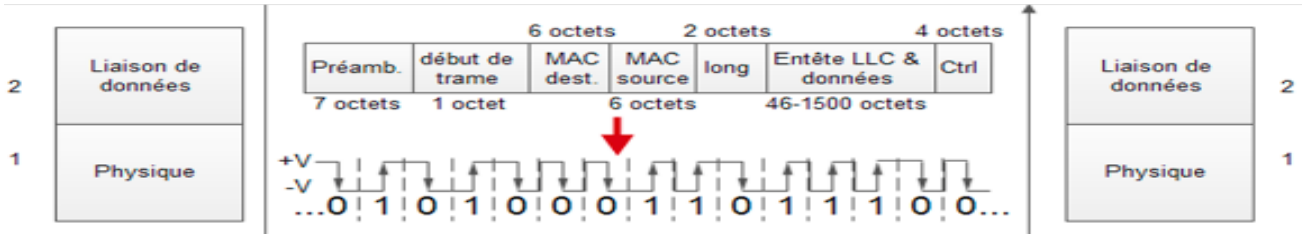


figure 4.1 Codage de l'information en Ethernet

Ethernet partagé :

La première version d'Ethernet qui n'est pratiquement plus utilisée a les caractéristiques suivantes :

- Tous les ordinateurs d'un réseau Ethernet sont reliés à une même ligne de transmission
- Toute machine est autorisée à émettre sur la ligne à n'importe quel moment et sans notion de priorité entre les machines ;
- Les différents nœuds réseau sont reliés entre eux par un concentrateur (hub);
- Un hub transmet sur tous les autres ports ce qu'il reçoit sur un port;
- A l'extérieur, rien ne le distingue d'un switch

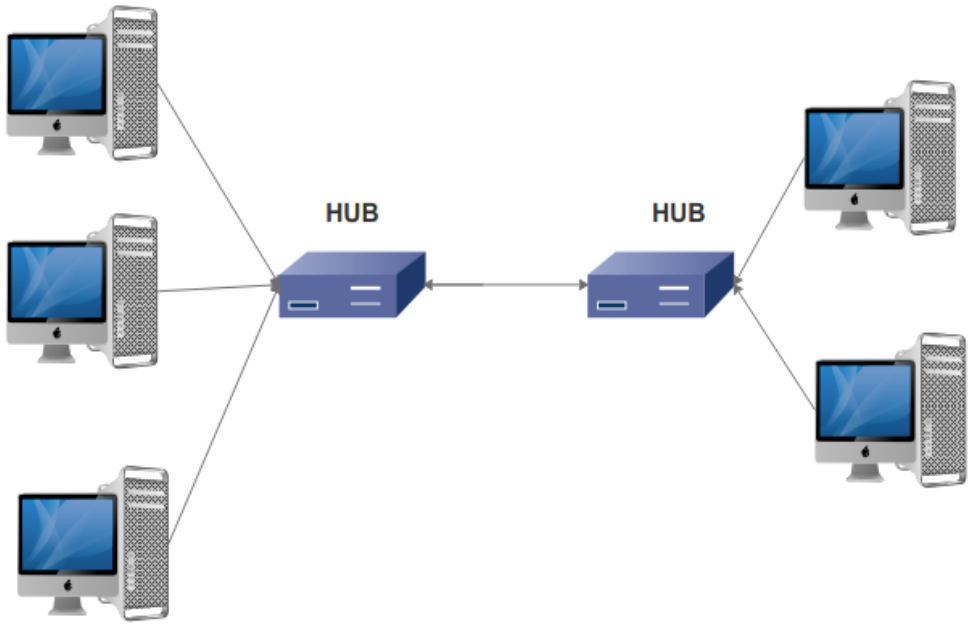


figure 4.2 Mise en cascade de deux Hubs

Ethernet partagé utilise :

- Une topologie physique en étoile, puisque les PC sont connectés à un point central ;

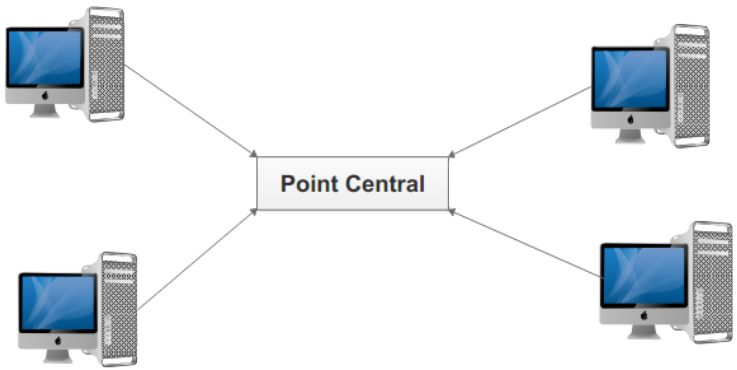


figure 4.3 Topologie physique de Ethernet partagé

- Une topologie logique en bus, puisque les données circulent comme si tous les PC étaient connectés sur la même ligne ;

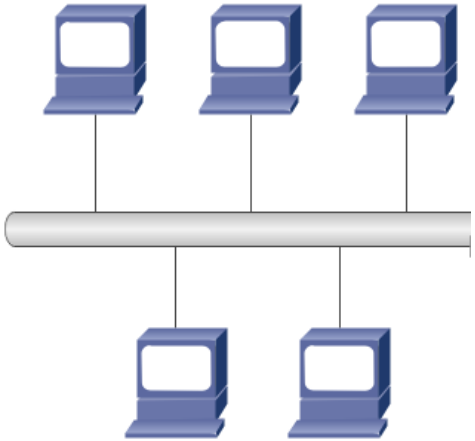


figure 4.4 Topologie logique de Ethernet partagé

Ethernet partagé : Accès aux supports de transmission

Dans Ethernet partagé on note :

- Chaque hôte émet des trames sur le câble
- Ces trames sont reçues par TOUS les autres hôtes (diffusion)
- Pas de confidentialité car il existe des sniffers tel que Wireshark qui peuvent écouter le support de transmission
- La bande passante disponible est partagée par l'ensemble des machines
  => plus il y a de machines, moins c'est fluide

figure 4.5 Propagation des données en Ethernet partagé

Méthode d’accès au support de transmissions

Ethernet partagé utilise la méthode d’accès CSMA/CD dont le principe est expliqué ci-dessous

Principe du CSMA/CD

Dans Ethernet partagé, le bus est partagé par toutes les stations.

Donc 2 stations émettant en même temps peuvent voir leurs signaux se brouiller : on dit alors qu’il y a **collision**.

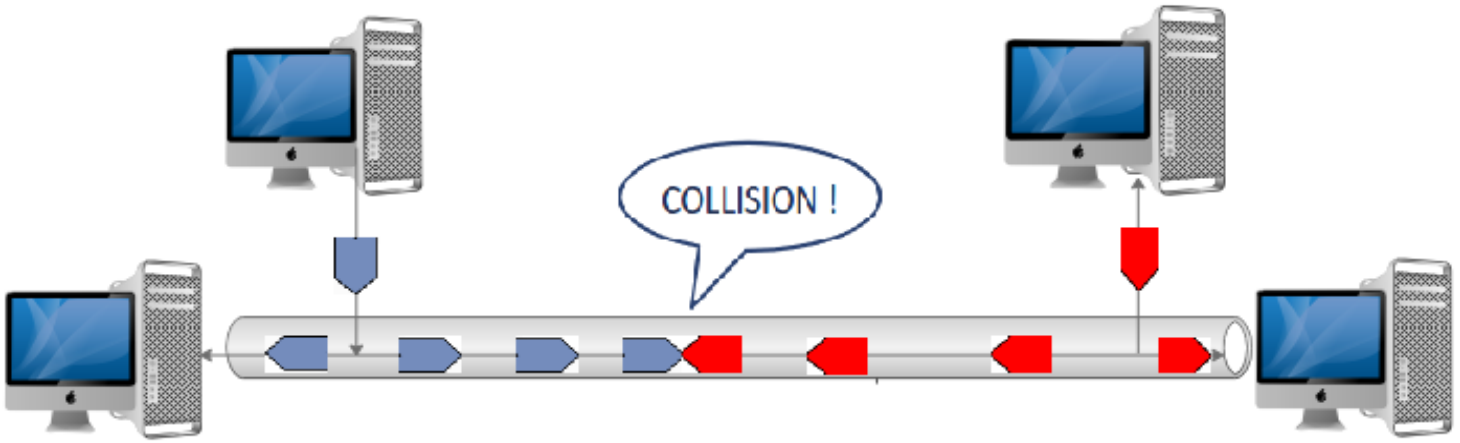


figure 4.6 Collision des données sur un support dans Ethernet partagé

Deux stations connectées à un même Hub sont dites être dans le même domaine de collision.

Pour résoudre le problème l'approche CSMA consiste à **écouter** tout d'abord la porteuse du signal pour savoir si les données y circulent sinon **transmettre** mais mettre en place un mécanisme pour **détecter la collision.**

Notons que :

- **CS = Carrier Send =** écoute de la porteuse (le signal)

  Avant de transmettre, on écoute le bus pour savoir si des données arrivent**.**
- **MA = Multiple Access**

  liaison partagée par toutes les machines : le bus
- **CD = Collision Detection**

  On ne cherche pas à éviter les collisions, on les détecte puis les corrige.

  Détection détectée en cas d'émission et de réception simultanées

Notons que la version d'Ethernet sans fil (**wifi**) utilise la méthode CSMA/CA qui consiste à écouter le support avant d'émettre tout en mettant un mécanisme pour éviter les collisions puisqu'il est difficile de les détecter en sans fil.

Dans la méthode CSMA/CA, CA signifie Collision Avoidance (évitement des collisions)

Principe du CSMA/CD

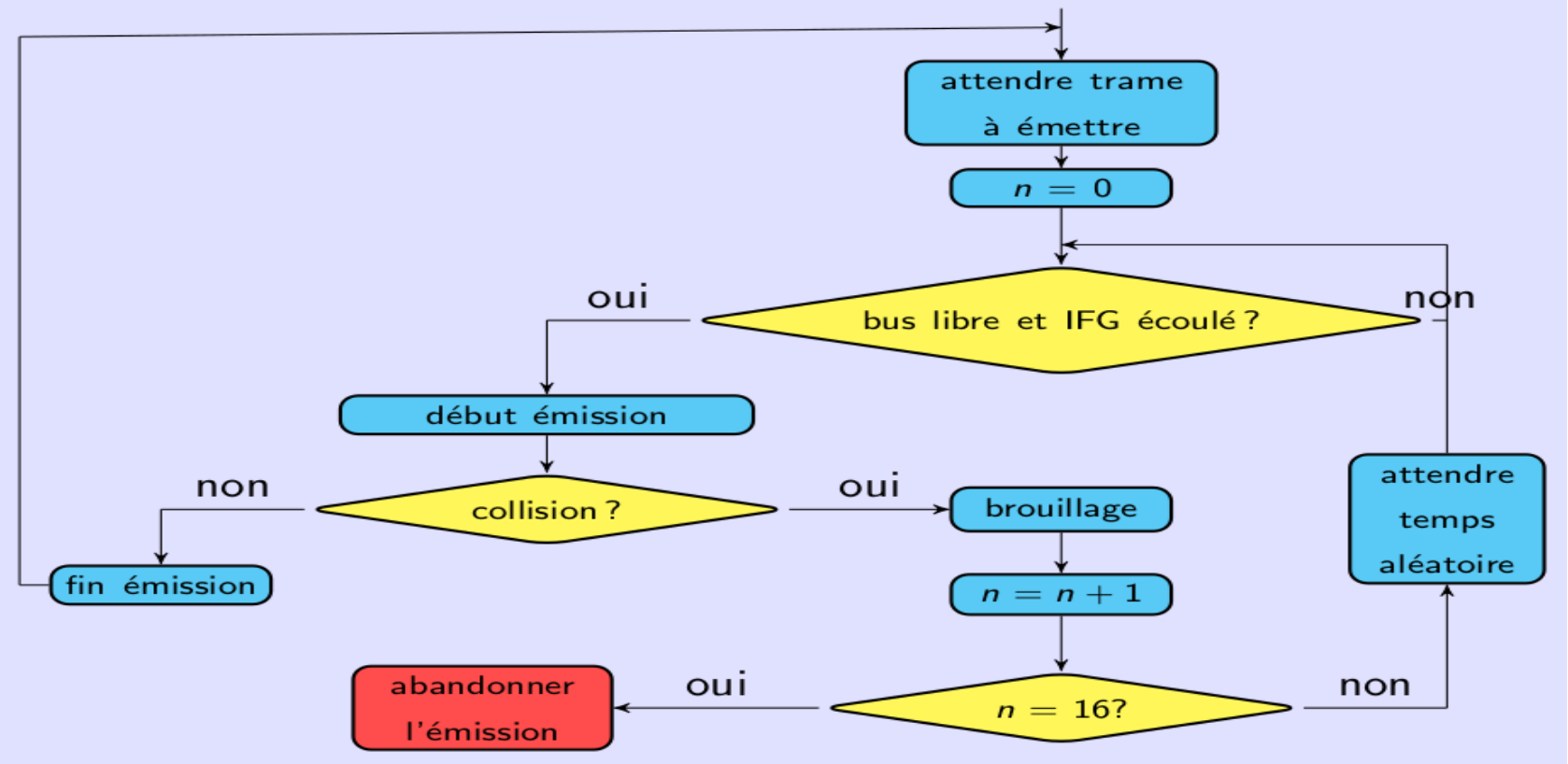


figure 4. Principe du CSMA/CD

Dans CSMA/CD, on utilise un délai de 9.6µs entre deux émissions ou réceptions appelé IFG pour Inter Fram Gap (Délai inter-trame) pour marquer une séparation entre les trames et permettre aux interfaces de se préparer à une nouvelle trame.

Lors d'une émission de données si une collision est détectée alors on envoie une séquence de 4 octets incohérents pour permettre à toutes les machines du réseau de s'assurer de la collision.
Cet envoi de 4 octets incohérents est appelé **brouillage.**

En cas de détection de collision, pour éviter que plusieurs machines éssayent de retransmettre au même moment elles attendent un temps tiré aléatoirement appelé délai de **BackOff ou délai aléatoire après collision.**

Méthode de réception

La méthode de réception mise en œuvre par une interface Ethernet peut être décrit comme suit :

1. écouter sur le bus et attendre qu'une trame arrive
2. quand une trame est arrivée, on vérifie :

   2.1 qu'elle a une longueur ≥ à 72 octets

   si une trame a une longueur de moins de 72 octets c'est qu'elle a subi une collision

   2.2 et qu'elle est correcte (reste de la division des champs de la trame par le polynôme générateur = FCS)

   (en cas de collision le brouillage garantit que la trame sera incorrecte)
3. si la trame est correcte on regarde ensuite son champ DA (Destination Adresse) :

   3.1 si DA = l'adresse de l'interface Ethernet ou FF:FF:FF:FF:FF:FF alors on délivre le champ de Données à la couche supérieure (au système d'exploitation dans le cas d'un paquet IP)

   3.2 sinon, la trame n'est pas destinée à l'interface Ethernet et on l'ignore

❖ **TRAME 802. 3**

Une trame Ethernet 1 est constituée des champs suivants comme le montre la figure 4.7:

- Préambule :
  - 56 bits = 7 X (10101010), dure 5.6 µs et permet aux autres stations d'acquérir la synchronisation bit
- Délimiteur de début de trame
  - (Start Frame Delimiter) : 8 bits = 10101011 ; permet aux autres stations d'acquérir la synchronisation trame.
- Adresse destination :
  - Adresse individuelle,
  - Adresse multicast,
  - Adresse broadcast
- Adresse source :
  - Adresse physique de la station émettrice, c'est une adresse individuelle
- Longueur du champ de données :

- Valeur comprise entre 1 et 1500, indique le nombre d'octets contenus dans le champ suivant; si la valeur est supérieure à 1500, la trame peut être utilisée à d'autres fins (autre protocole que IEEE 802.3, permet la compatibilité avec Ethernet)

- Padding :

  - Contenu sans signification complétant à 64 octets la taille totale d'une trame dont la longueur des données est inférieure à 46 octets; en effet, une trame est considérée valide (non percutée par une collision) si sa longueur est d'au moins 64 octets; 46 <= (données + padding) <= 1500 (6+6+2+1+45+2=64 octets)

- Contrôle :
  - Séquence de contrôle basée sur un CRC polynomial de degré 32

  - Sens de circulation des octets : selon la structure logique de la trame : préambule = premier octet émis, FCS = dernier octet émis

  - Le sens de circulation des bits par octets se fait selon le schéma suivant : LSB first

Trame IEEE 802.3 :

figure 4.7 Champs de la trame Ethernet 1 sans padding

Trame Ethernet II

- Il existe une autre trame Ethernet légèrement différente de la trame 802.3, la trame Ethernet II
- Elles peuvent coexister grâce au champ long/type différent

Trame Ethernet II :

Préambule	Délimiteur du début de trame	Adr.MAC Destination	Adr. MAC source	Type	Entéte LLC & Données	Ctrl
7 octets	**1 octet**	**6 octets**	**6 octets**	**2 octets**	**46-1500 octets**	4 octets

Trame IEEE 802.3 :

Préambule	Délimiteur du début de trame	Adr.MAC Destination	Adr. MAC source	Long	Entéte LLC & Données	Ctrl
7 octets	**1 octet**	**6 octets**	**6 octets**	**2 octets**	**46-1500 octets**	4 octets

figure 4.8 Comparaison entre les trames Ethernet I et Ethernet II

- Le champ type indique le protocole de couche supérieure recevant les données
  Exemple :
  0x0800 : Si le protocole de la couche supérieure est **IPv4**
  0x86DD : Si le protocole de la couche supérieure est **IPv6**
  0x0806 : S'il faut remettre les données utiles au protocole **ARP**

- La trame Ethernet II n’ayant de sous-couche LLC, le padding éventuel sera enlevé par la couche supérieure

Trame Ethernet II :

figure 4.9 La trame Ethernet II

Adresses MAC

- Une adresse MAC est un identifiant unique attribué à chaque périphérique réseau (comme une carte Wifi ou Ethernet) par le fabricant. MAC signifie Media Access Control et chaque identifiant est unique à chaque périphérique.
- Une adresse MAC est codée sur 48 bits (6 octets) et se compose de six groupes de deux caractères chacun, séparés par deux points. Voici un exemple d'adresse MAC : 3c:07:54:3c:75:f9

Exemple d'affichage de l'adresse MAC de la carte Ethernet enp2s0 sous Linux:

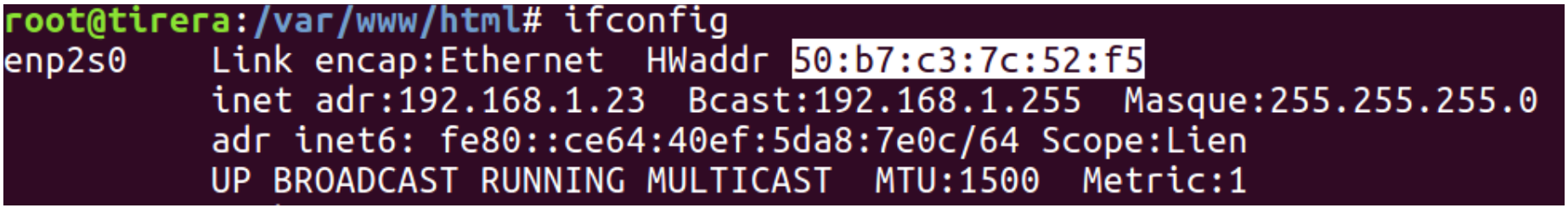


figure 4.10 Affichage de l'adresse MAC d'une interface sous Linux

- L'adresse MAC de diffusion comporte 48 uns (1), représentés au format hexadécimal **FF:FF:FF:FF:FF:FF.**

Ethernet commuté

Ethernet commuté a été mis en place pour résoudre les problèmes que pose Ethernet I à savoir :

- La bande passante est partagée et plus il y a d'hôtes, plus c'est lent

- Comme il ne peut y avoir qu'une trame à la fois sur le support, on travaille en half-duplex, ce qui est plus lent que le full-duplex (transmission simultanée dans les deux sens)

- Plus il y a d'hôtes, plus les collisions sont fréquentes et ralentissent encore le système

LE PRINCIPE d'Ethernet commuté

- La topologie physique reste en étoile, organisée autour d'un commutateur (switch). Mais le commutateur utilise un mécanisme de filtrage et de commutation.
- La topologie logique est une étoile
- Le switch inspecte les adresses de source et de destination des trames, dresse une table appelée **table de commutation** lui permettant de savoir quelle machine (adresse MAC) est connectée sur quel port du switch. La table de commutation se fait par auto-apprentissage.

- Connaissant le port du destinataire, le commutateur ne transmettra la trame que sur le port adéquat, les autres ports restants dès lors libres pour d'autres transmissions pouvant se produire simultanément.

- Il en résulte que chaque échange peut s'effectuer à débit nominal (plus de partage de la bande passante), sans collisions, avec pour conséquence une augmentation très sensible de la bande passante du réseau (à vitesse nominale égale)

- Puisque la commutation permet d'éviter les collisions et que les techniques 10/100/1000 base T(X) disposent de **circuits séparés** pour la **transmission** et la **réception** (une paire torsadée par sens de transmission), la plupart des commutateurs modernes permet de **désactiver la détection de collision et de passer en mode full-duplex** sur les ports.

- De la sorte, les machines peuvent **émettre et recevoir en même temps** (ce qui contribue à nouveau à la performance du réseau)

- Le protocole CSMA/CD est donc devenu obsolète
- Les commutateurs Ethernet modernes détectent également la vitesse de transmission utilisée par chaque machine (autosensing) et si cette dernière supporte plusieurs vitesses (10 ou 100 ou 1000

megabits/sec), ils entament avec elle une négociation pour choisir une vitesse ainsi que le mode semi-duplex ou full-duplex de la transmission.

- Cela permet d'avoir un parc de machines ayant des performances différentes (par exemple un parc d'ordinateurs avec diverses configurations matérielles)

- Comme le trafic émis et reçu n'est plus transmis sur tous les ports, il devient beaucoup plus difficile d'espionner (sniffer) ce qui se passe

- Voilà qui contribue à la sécurité générale du réseau, qui est un thème fort sensible aujourd'hui

- Pour terminer, l'usage de commutateurs permet de construire des réseaux plus étendus géographiquement. En effet en Ethernet partagé, un message doit pouvoir atteindre toute autre machine dans le réseau dans un intervalle de temps précis (slot time) sans quoi le mécanisme de détection des collisions (CSMA/CD) ne fonctionne pas correctement. Ceci n'est plus d'application avec les commutateurs Ethernet. La distance n'est plus limitée que par les limites techniques du support utilisé (fibre optique ou paire torsadée, puissance du signal émis et sensibilité du récepteur, atténuation du signal avec la distance, …)

Auto-construction d’une table de commutation par un switch

La table de commutation contient une liste d’enregistrements (adresse MAC, port)

1.Le switch reçoit une trame Ethernet

2.Il vérifie la validité de celle-ci grâce au FCS contenu en fin de trame, si elle est valide on passe au n°3, sinon il la détruit.

3.Le switch analyse l’adresse MAC source de la trame.

3.1 Si elle n’est pas présente dans sa table de commutation, il rajoute une nouvelle entrée en l’associant à l’interface par laquelle elle est entrée et lui attribue une durée de vie (300 secondes par défaut sur un switch Cisco)

3.2 Si elle est présente et associée à la même interface, le switch rafraîchit la durée de vie.

3.3 Si elle est présente mais associée à une autre interface, le switch crée une nouvelle entrée comme s’il s’agissait d’une nouvelle adresse MAC et ensuite supprime l’ancienne entrée.

4.Le switch analyse l’adresse MAC destination.

4.1Si l’adresse MAC existe dans la table d’adresse MAC et associée à une interface dans le même vlan que celle d’entrée, le switch propage la trame uniquement par cette interface.

4.2Si l’adresse MAC n’est pas présente dans la table d’adresse MAC, le switch propage la trame par toutes les interfaces du même vlan sauf celle d’où elle provient.

4.3 Si l’adresse MAC est soit l’adresse broadcast soit une adresse multicast, le switch propage la trame par toutes les interfaces du même vlan sauf celle d’où elle provient.

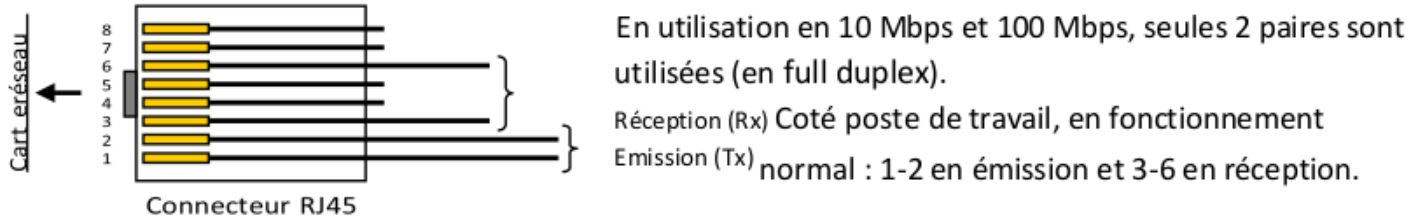
```
SWI#show mac-address-table
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
c202.1db9.0000          Self          1     Vlan1
c201.1daa.0000          Dynamic       1     FastEthernet1/0

SWI#
```

Figure 4.11 Exemple d'une table de commutation d'un switch CISCO

Remarque

Dans Ethernet avec un débit de moins de 100Mbits/s sur les 8 fils de câble à paires torsadées, on utilise deux fils pour la transmission et deux fils pour la réception des données comme le montre la figure ci-après:



Remarquer la numérotation des fils dans le connecteur RJ45 de gauche à droit

Remarque :

Ethernet est le type de réseau le plus répandu en informatique.

Ethernet utilise les 3 catégories de supports de transmission :

- câbles à paires torsadées
- fibre optique
- les ondes radios (Wifi)

Le tableau 4.1 ci-après donne quelques variantes d'Ethernet :

tableau 4.1 quelques variantes d'Ethernet

Sigle	Dénomination	Câble	Débit	Portée
10Base2	Ethernet ince (thin Ethernet)	Câble coaxial (50 Ohms) de faible diamètre	10 Mb/s	185m
10Base5	Ethernet épais (thick Ethernet)	Câble coaxial de gros diamètre	10Mb/s	500m
10Base-T	Ethernet standard	Paire torsadée (catégorie 3)	10 Mb/s	100m
100Base-TX	Ethernet rapide (Fast Ethernet)	Double paire torsadée (catégorie 5)	100 Mb/s	100m
100Base-FX	Ethernet rapide (Fast Ethernet)	Fibre optique multimode	100 Mb/s	412m
		Fibre optique monomode	100 Mb/s	2km

Gigabits Ethernet				
1000BaseLX		Fibre optique en multi-mode	1000 Mb/s	550m
		Fibre optique en Single mode	1000 Mb/s	3km
1000Base T	IEEE 802.3ab	UTP catégorie 5e ou 6^{e} avec 4 paires de fil utilisées	1000 Mb/s	1000m
1000BaseLH		Fibre optique en monomode	1000 Mb/s	10km
10gigabits Ethernet (IEEE 802.3ae)				
10GBase-SR		Fibre optique multimode	10Gb/s	400m
10GBase-LR		Fibre optique monomode	10Gb/s	10km
10GBase-ER		Fibre optique monomode	10Gb/s	40km
10GBase-T		UTP ou STP avec 4 paires torsadées	10Gb/s	100m

Chapitre 5 : Les protocoles de couche3(IP, ARP,ICMP…)

Objectifs spécifiques du chapitre 5 et 6 : Couche 3 Les protocoles IP, ARP et ICMP et routage IP

1. Comprendre le fonctionnement de IP
2. Être capable de décrire les différents champs d'un datagramme IP
3. Comprendre l'identifiant par adresse IP d'une machine dans un réseau IP
4. Comprendre le rôle de masque de réseau en terme d'identification de la partie NET_ID d'une adresse IP
5. Comprendre et mettre en œuvre le Subneting et le Superneting
6. Comprendre le rôle du protocole ARP et savoir afficher la table ARP d'un équipement réseau.
7. Comprendre le rôle du protocole ICMP en terme de test de connectivité et d'information de non remise de datagramme IP suite à un problème réseau.
8. Comprendre et mettre en œuvre la translation d'adresse et de port

Sommaire

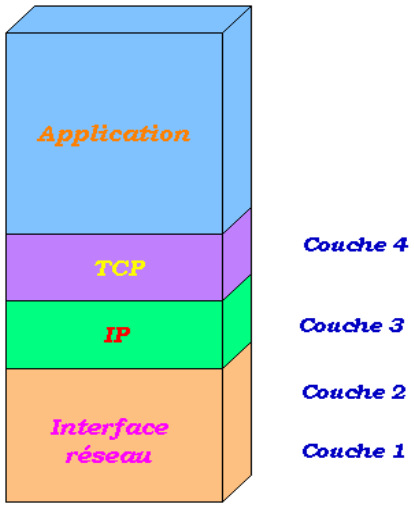
5.1 Présentation du protocole IP

Le **protocole IP (Internet Protocol)** est un des protocoles les plus importants d'Internet. Il permet de spécifier la destination par une adresse (IP). Cependant il n'assure pas la livraison : aucun message n'est mis en œuvre par IP pour s'assurer qu'un datagramme arrive à la destination c'est pourquoi on dit que IP est un protocole **non fiable**.

Par contre un routeur qui reçoit un datagramme IP et ne sachant pas comment le transférer utilise un message **ICMP** pour informer l'émetteur comme on le verra dans la paragraphe contenant **ICMP**.

Pour dialoguer sur un réseau une machine à besoin :

- **D'une adresse IP**
- **D'un masque** de sous réseau.
- **D'une passerelle** : Machine à qui on doit transmettre le paquet lorsque le destinataire n'est pas dans le réseau local.



Standard TCP/IP

5.2 Le paquet IP

- On parle de **paquet IP** ou de **datagramme IP**.
- Le paquet IP ne fait que contenir les informations nécessaires à la réalisation d'une interconnexion.
- Le **champ de données** de la **trame Ethernet** correspond au **paquet IP**.

Un datagramme IP est encapsulé dans une trame Ethernet avant d'être transmis dans un réseau local comme le montre la figure 5.2

- trame Ethernet II contenant un datagramme IPv4 (EtherType vaut 0x0800) :

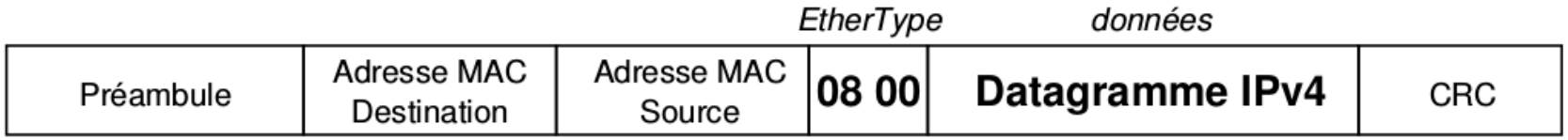


figure 5.2 Trame Ethernet II contenant un datagramme IPv4 (Ethernet vaut 0x0800)

A la réception d'une trame Ethernet par la couche accès réseau du TCP/IP, les données sont extraites et transmises au bon protocole de la couche Internet selon le champ EtherType comme le montre la figure 5.3

Ce processus se basant sur le champ EtherType pour transmettre au bon protocole de la couche supérieure est appelé **démultiplexage**

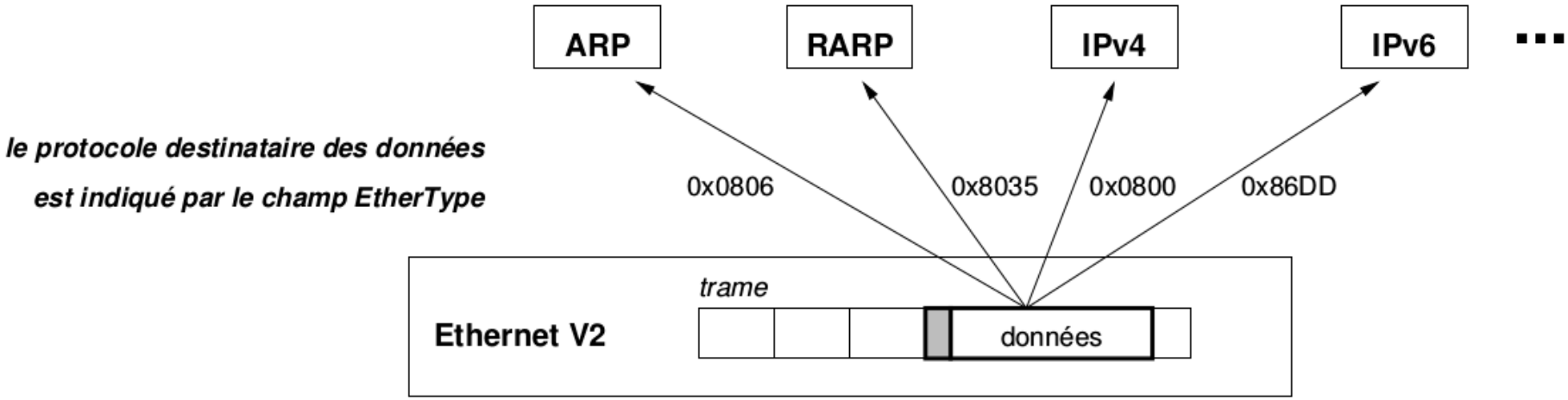


figure 5.3 Démultiplexage du champ données d'une trame Ethernet

Composition d’un datagramme IP

Un datagramme IP est composé d’une :

- **en-tête** de taille comprise entre 20 et 60 octets
- **données IP** nombre quelconque d’octets (limité à 65 315) comme le montre la figure ci-dessous:

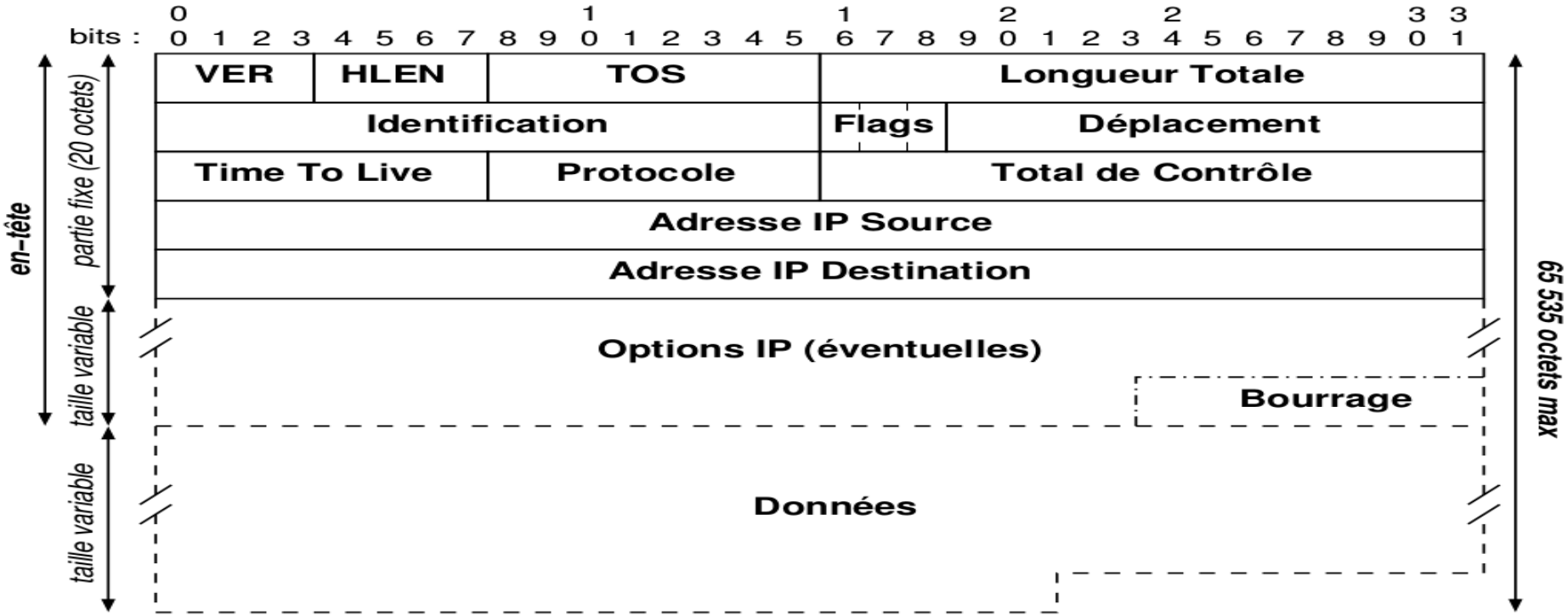


figure 5.4 Composition d’un datagramme IP

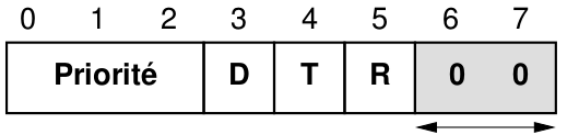
Description des différents champs de l'en-tête IP :

- Champ **Version :**
  - codé sur 4 bits
  - identifie la version du (format du) datagramme
  - actuellement, la version est 4 (codée 0100 en binaire)
  - dans le datagramme IPv6, ce champ est maintenu et vaut 6
  - permet de s'assurer que le datagramme sera correctement interprété
- Champ **Adresses :**
  - **adresse IP Source : (32 bits)**
    - identifie l'hôte à l'origine du datagramme
  - **adresse IP Destination : (32 bits)**
    - identifie le destinataire final du datagramme

figure 5.5 Exemple d'adresse IP source et destination dans un datagramme IP

- Champ **Longueur d'en-tête (HLEN)**
  - (internet) Header LENgth
  - codée sur 4 bits
  - indique le nombre de mots de 32 bits de l'en-tête (comprenant les options) :
    - en-tête de 20 à 60 octets
    - $5 \leq HLEN \leq 15$
  - si HLEN > 5 alors il y a des options

- Champ **Longueur Totale**
  - codée sur 16 bits
  - indique le nombre de total d'octets du datagramme (en-tête + données)
  - comprise entre 20 et 65 535
- Champ **Type Of Service (TOS)**
  - codé sur 8 bits :



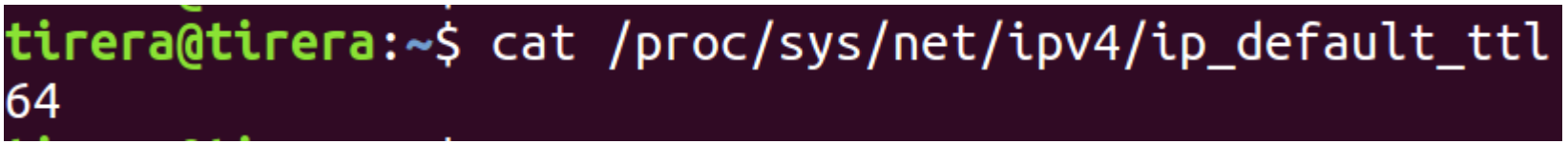
Priorité	
val_2	signification
000	routine
001	priority
010	immediate
011	flash
100	flash override
101	critic
110	internetwork control
111	network control

  - Priorité : de 0 à 7
    - distinction entre "normal" et "contrôle"
    - routeurs : infos trafic 6 et 7
  - bits D, T et R : type d'acheminement désiré :
    - **D**(elay) : délai d'acheminement court
    - **T**(hroughput) : débit de transmission élevé
    - **R**(eliability) : grande fiabilité

    - le TOS est un **souhait** que les routeurs peuvent ignorer

À la fin des années 1990, remplacé par le champ Differentiated Services, offrant plus de finesse pour exprimer une qualité de service (QoS) désirée.

- Champ **Time To Live (TTL)**
    - codé sur 8 bits
    - indiqué par l'émetteur pour limiter :
        - la "durée de vie" du datagramme en secondes (plus vraiment d'actualité)
        - le nombre de routeurs traversés par le datagramme (nombre de sauts)
    - décrémenté par routeurs et stations traitant le datagramme :
        - de 1 à chaque traversée d'un routeur (saut)
        - du temps passé en file d'attente
    - si atteint 0, le datagramme est détruit, et l'émetteur est informé par un message ICMP
    - évite qu'un datagramme ne circule indéfiniment
    - évite que des fragments d'un datagramme ne soient gardés inutilement
    - Souvent fixé bien en deçà de la valeur maximale. Par défaut, sur Linux :



- Champ **Protocole**

  Ce champ est codé sur 8 bits et indique le protocole devant recevoir les données du datagramme IP au moment du démultiplexage.

  Voici quelques valeurs officielles de ce champ :

val_{10}	protocole
0	IP
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Principe du démultiplexage par la couche IP

A la réception d'un datagramme IP par la couche IP celle-ci consulte ce champ pour savoir s'il doit remettre à le datagramme IP à ICMP, UDP, TCP etc comme le montre la figure suivante:

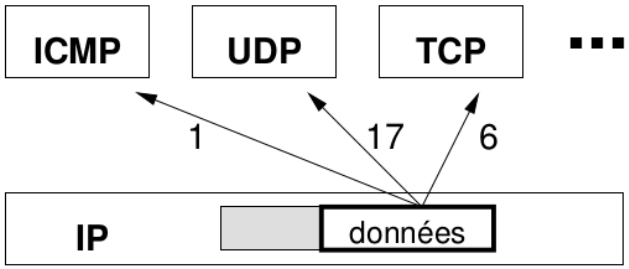


figure 5.6 Démultiplexage du champ données d'un datagramme IP par la couche Internet

- Champ Total de Contrôle d'en-tête (checksum)

Ce champ est codé sur 16 bits et permet de contrôler l'intégrité de l'en-tête IP.
Il faut noter que :

- Ce champ est calculé par l'émetteur et utilisé comme suit par les routeurs et le destinataire :
- On stocke le checksum du datagramme reçu et on recalcule la nouvelle valeur du checksum
- si les deux résultats sont différents on détruit le datagramme IP

- Sinon on traite le datagramme IP.

- Il est à noter que chaque routeur de recalculer ce champ car ce dernier décrémente la valeur de TTL qui fait parti de l'en-tête IP.

La figure 5.7 ci-après décrit le calcul du champ checksum :

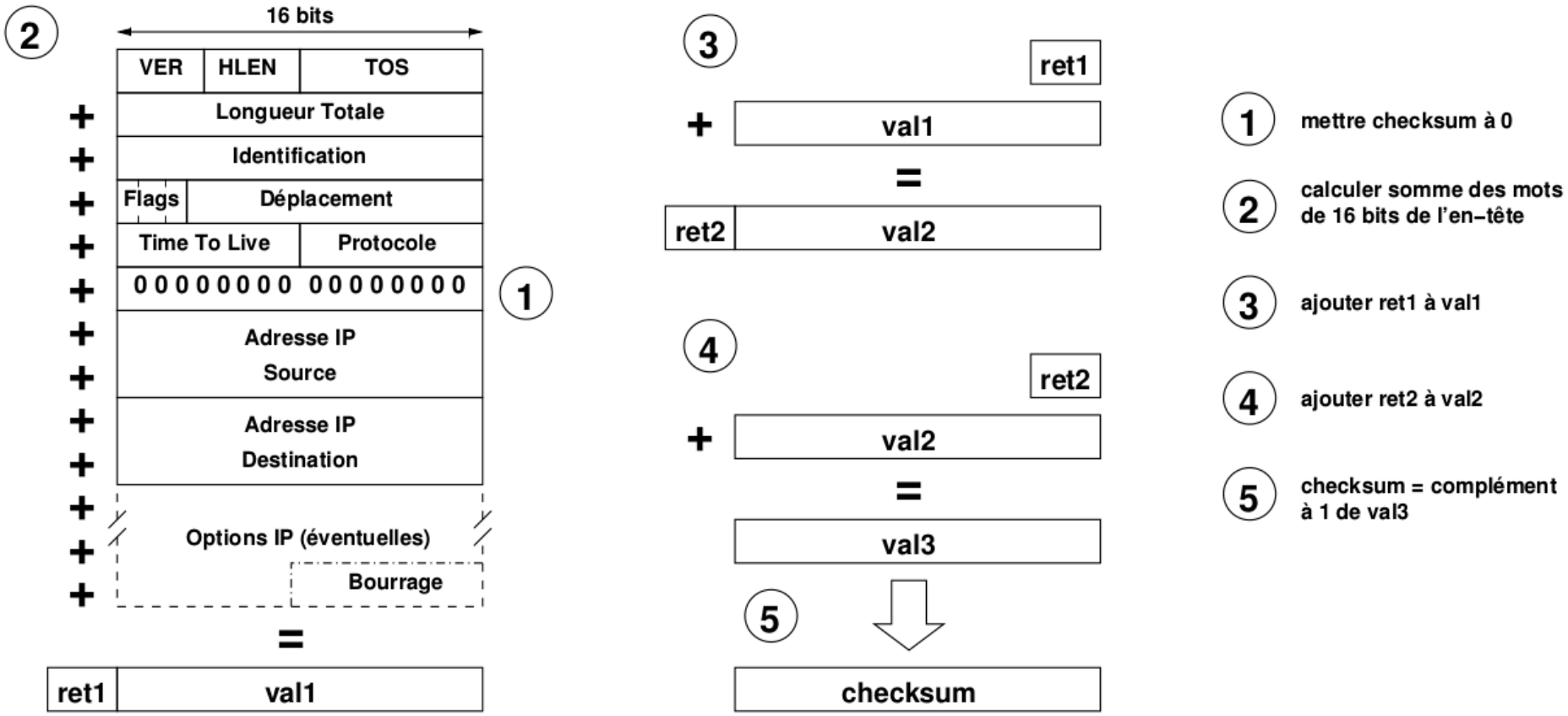


figure 5.7 Calcul du champ Checksum d'un datagramme IP

- **MTU et fragmentation**

Il est à noter qu'un routeur tient compte de le taille maximale des données transportées sur un réseau physique, appelée aussi **charge utile** (payload).

Réseaux physiques	Valeur MTU
Ethernet	1 500 octets
Token Ring	4 ou 16 Ko
X.25	128 octets recommandés (max 255)
Frame Relay	1 600 octets

IP fragmente tout datagramme plus grand que le MTU du réseau qui doit le transporter comme le montre la figure ci-aprés :

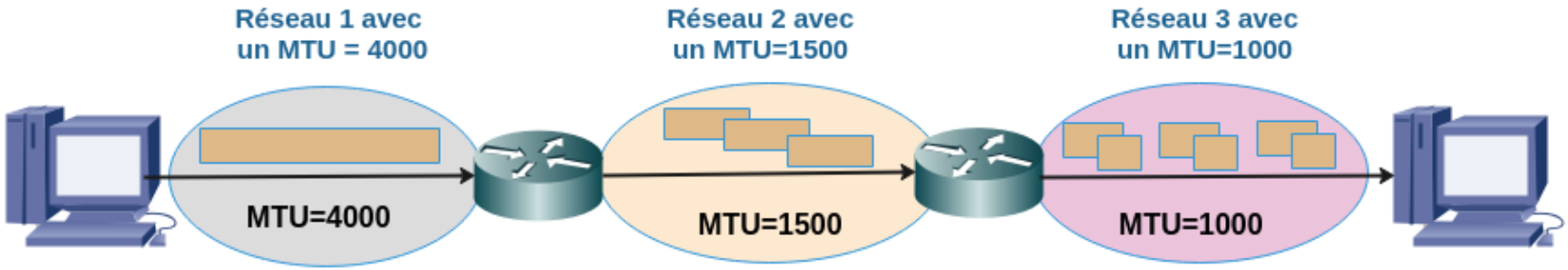
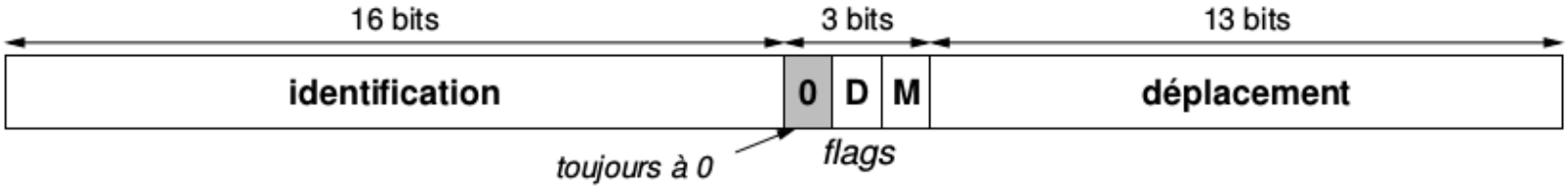


figure 5.8 Fragmentation d'un datagramme IP selon les MTU des différents réseaux trasversés

Chaque **fragment** est un datagramme acheminé indépendamment (peut suivre une route différente des autres fragments) et peut être à son tour fragmenté



- **identification** : valeur identifiant le datagramme d'origine. Les fragments d'un datagramme ont les mêmes identification, IP Source, IP Destination et Protocole
- **bit D(on't Fragment)** : le datagramme ne doit pas être fragmenté (détruit et message ICMP si impossible)
- **bit M(ore)** : à 0 si datagramme non fragmenté ou si c'est le dernier fragment ; sinon vaut 1
- **déplacement (Offset)** : déplacement × 8 est la position absolue (ou numéro) du premier octet de données de ce datagramme dans le datagramme d'origine.

  Vaut 0 si pas de fragmentation

Réassemblage

Il est à noter que le réassemblage des fragments n'est jamais fait au niveau des routeurs intermédiaires mais réaliser par le destinataire final.

En effet le destinataire final met en attente les fragments des datagrammes incomplets les réordonne pour avoir le datagramme d'origine.

Il détruit tous les fragments d'un datagramme IP si le TTL de l'un d'eux passe à 0(zéro) et informe l'émetteur par un message de type ICMP.

5.2 Systèmes de numérotation

L'objectif de ce paragraphe est de permettre aux apprenants de pouvoir passer un nombre dans une base en une autre base.

Le système décimal (base 10)

Les nombres que nous utilisons habituellement sont ceux de la base 10 (système décimal). Nous disposons de dix chiffres différents de 0 à 9 pour écrire tous les nombres. D'une manière générale, toute base N est composée de N chiffre de 0 à N-1. Soit un nombre décimal N = 2348. Ce nombre est la somme de 8 unités, 4 dizaines, 3 centaines et 2 milliers. Nous pouvons écrire N = (2 x 1000) + (3 x 100) + (4 x 10) + (8 x 1)

$$2348 = (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (8 \times 10^0)$$

10 représente la base et les puissances de 0 à 3 le rang de chaque chiffre. Quelque soit la base, le chiffre de droite est celui des unités. Celui de gauche est celui qui a le poids le plus élevé.

Le binaire (base 2)

Dans les domaines de l'automatisme, de l'électronique et de l'informatique, nous utilisons la base 2. Tous les nombres s'écrivent avec deux chiffres uniquement (0 et 1). De même que nous utilisons le système décimal parce que nous avons commencé à compter avec nos dix doigts, nous utilisons le binaire car les systèmes technologiques ont souvent deux états stables.

- Un interrupteur est ouvert ou fermé
- Une diode est allumée ou éteinte
- Une tension est présente ou absente

A chaque état du système technologique, on associe un état logique binaire. La présence d'une tension sera par exemple notée **1** et l'absence **0**.

Avec un bit nous pouvons coder deux états

0
1

Avec deux bits nous pouvons coder quatre états

0	0
0	1
1	0
1	1

Avec trois bits nous pouvons coder huit états

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

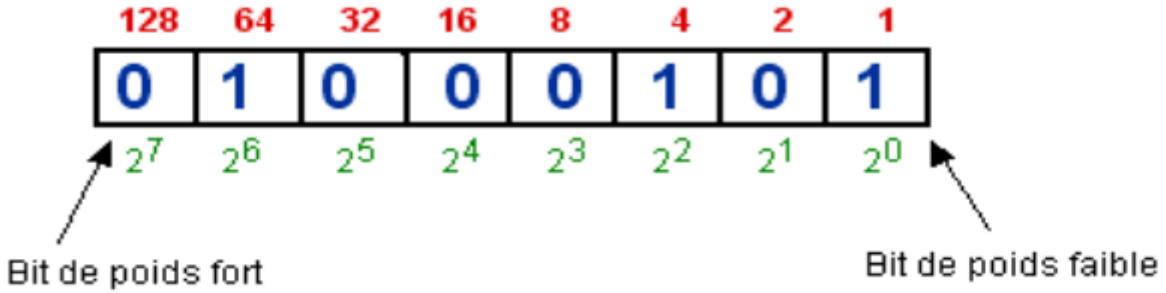
A chaque nouveau bit, le nombre de combinaisons possibles est doublé. Ce nombre est égal à **2 puissance N** (N étant le nombre de bits). Un groupe de bits est appelé un mot, un mot de huit bits est nommé un octet (byte).

0	1	0	0	0	1	0	1

Avec un octet, nous pouvons écrire **2 puissance 8 = 256** nombres binaires de **0 à 255** Les règles sont les mêmes que pour le décimal.

$$1011_{(2)} = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$$
$$1011_{(2)} = (1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1)$$
$$1011_{(2)} = 11_{(10)}$$

Description d'un octet.



Un 1 dans une case représente la valeur décimale qui est au dessus.

Correspondance entre binaire et décimal.
Conversion d'un nombre binaire en décimal. Il suffit de faire la somme des poids de chaque bit à 1 Le nombre ci dessus est égal à 64 + 4 + 1 = 69

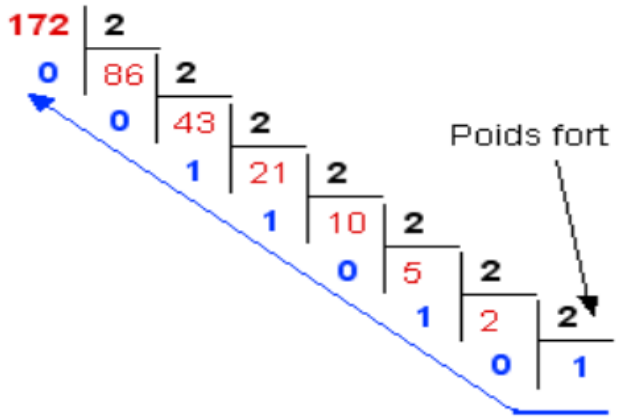
Conversion d'un nombre décimal en binaire (exemple : N = 172).
Méthode par soustractions.

172	44	12	4
- 128	- 32	- 8	- 4
44	12	4	0

$172 = 128 + 32 + 8 + 4$

$172_{(10)} = 10101100_{(2)}$

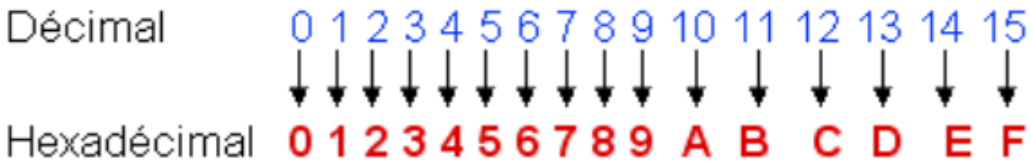
Méthode par divisions



172 / 2 = 86, il reste 0 ...

L'hexadécimal

La manipulation des nombres écrits en binaire est difficile pour l'être humain et la conversion en décimal n'est pas simple. C'est pourquoi nous utilisons de préférence le système hexadécimal (base 16). Pour écrire les nombres en base 16 nous devons disposer de 16 chiffres, pour les dix premiers, nous utilisons les chiffres de la base 10, pour les suivant nous utiliserons des lettres de l'alphabet.



Les règles sont ici aussi les mêmes que pour le décimal.

$$A3F_{(16)} = (A \times 16^2) + (3 \times 16^1) + (F \times 16^0)$$

$$A3F_{(16)} = (10 \times 256) + (3 \times 16) + (15 \times 1)$$

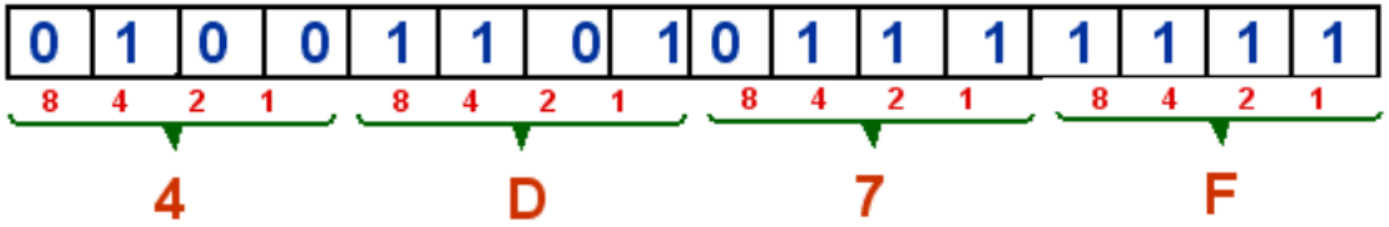
$$A3F_{(16)} = 2560 + 48 + 15 = 2623_{(10)}$$

Correspondance entre binaire et hexadécimal.

La conversion du binaire en hexadécimal est très simple, c'est d'ailleurs la raison pour laquelle nous utilisons cette base.

Il suffit de faire correspondre un mot de quatre bits (quartet) à chaque chiffre hexadécimal.

Conversion d'un mot de 16 bits entre binaire et hexadécimal



$4D7F_{(16)} = 0100110101111111_{(2)}$

Correspondance entre décimal et hexadécimal.

La méthodes par divisions s'applique comme en binaire (exemple : N = 2623).

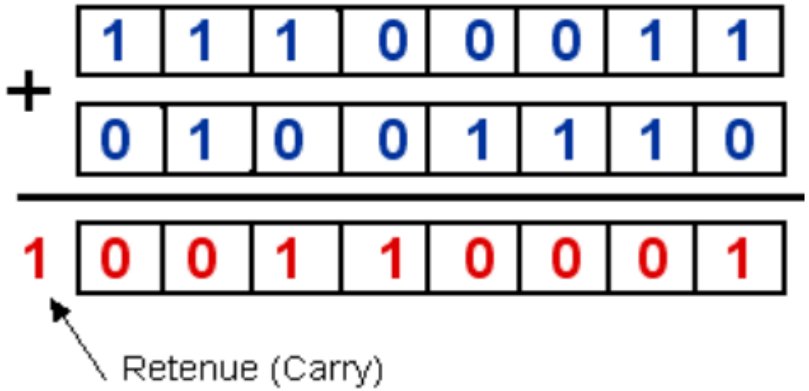
2623 / 16 = 163, il reste 15...

Opérations arithmétiques et logiques

Addition en binaire

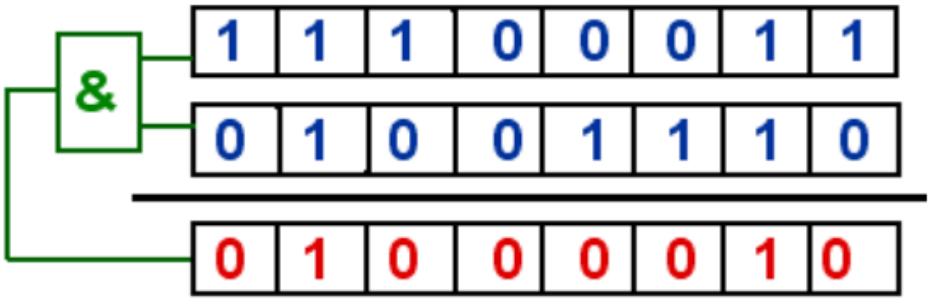
L'addition est réalisée bit à bit.

$1 + 0 = 1$
$1 + 1 = 10$
$1 + 1 + 1 = 11$



Produit logique en binaire

La fonction ET est appliquée bit à bit



5.3 Adressage IP

- Une **adresse IP**, universelle ou publique ou routable, est **unique au niveau mondial**.

- Elle est codée sur **32 bits** soit **4 octets**, la notation la plus courante consiste à indiquer chaque octet en décimal et à les séparer par des points.
  - *Exemple : **196.1.95.5***

- Plus précisément, l'adresse IP d'un ordinateur est composée de deux parties :
  - *La première partie correspond à **l'adresse du réseau ou Net-ID.***
  - *La deuxième partie correspond à **l'adresse de la machine ou Hots-ID** sur le réseau.*

Attribution d’une adresse IP publique

- L’adresse publique d’ordinateur pour une connexion à Internet est attribuée parmi celles dont dispose votre **Fournisseur d’Accès à Internet (FAI).**

- Celle-ci a été demandée au préalable à un organisme officiel, garantissant ainsi l’unicité des adresses de réseau au niveau mondial. C'est **l'ICANN (Internet Corporation for Assigned Names and Numbers)**, qui est chargée d'attribuer des adresses IP publiques.

Le masque de réseau

- Un **masque** de sous-réseau a la **même forme** qu'une **adresse IP** (32 bits). Il a pour rôle de **distinguer** le **numéro du réseau**, du **numéro de l'ordinateur** dans ce réseau.

- Dès lors qu'un **équipement** possède une **adresse IP**, il est extrêmement important de connaître le masque associé afin de déterminer le réseau dans lequel appartient cette machine.

Par convention, les bits de gauche d'un masque sont à 1 et les bits de droite sont à 0.

- Exemple : 11111111 11111111 11111111 00000000 ce qui correspond à 255.255.255.0
- Pour connaître le réseau dans lequel appartient une machine, on fait un & logique entre le masque de sous réseau et l'adresse IP de la machine.
- Exemple:

Une machine possède l'adresse IP :	194 . 214 . 110 . 35
Elle posséde un masque :	255 . 255 . 255 . 0
L'adresse du réseau est :	194 . 214 . 110 . 0
L'adresse de cette machine dans le réseau est :	35

- Dans chaque réseau, les adresses dont les bits de machine sont tous à 0 (valeur 0) ou tous à 1 (valeur 255) ne peuvent être attribuées :
  - Tous les bits à '0' désigne le réseau dans son ensemble

- Tous les bits à ‘1’ représente l’adresse de diffusion (**broadcast**) à destination de tous les nœuds du réseau.
- Une machine dans un réseau IP utilise son masque de réseau pour savoir si une autre machine dont elle connaît l'adresse IP se trouve dans son réseau ou non. De manière précise, si une machine A dont l'adresse IP est IPA veut initier une communication vers une machine dont l'adresse IP est IPB alors le processus suivant est mis en œuvre par A pour savoir si B se trouve sur ce réseau ou pas :
- A applique son masque de réseau NMA à son adresse IP pour obtenir son adresse réseau notée IPRXA
- A applique aussi son masque de réseau NMA à l'adresse IP de B pour obtenir l'adresse réseau de B notée IPRXB
- SI IPRXA = IPRXB alors A se dit qu'elle se trouve dans même réseau que B.
- Si IPRXA est **différente** IPRXB alors A se dit qu'elle ne se trouve pas dans le même réseau que B; dans ce cas la communication entre A et B n'est possible que par l'intermédiaire d'une passerelle(par défaut).

Comme l’illustre la figure ci-dessous :

A veut initier une communication vers la machine B

- A applique son masque de réseau à **son adresse IP** pour obtenir **son adresse réseau** comme suit:

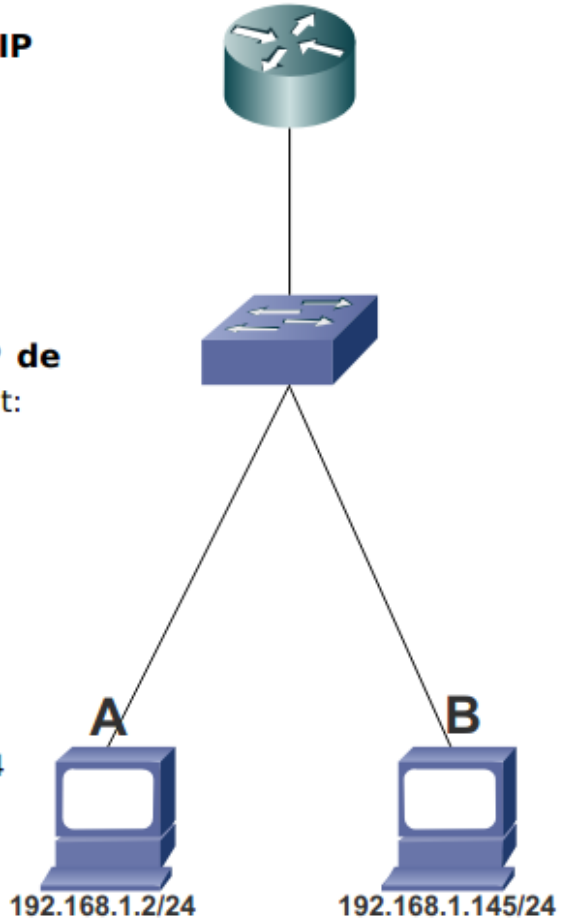
```
  11000000 . 10101000 . 00000001. 00000010
&
  11111111 . 1111 1111 . 1111 1111. 00000000
-----------------------------------------------
Rest1 = 192    .   168  .   1   .   0
```

- A applique son masque de réseau **à l'adresse IP de B** pour obtenir **l'adresse réseau de B** comme suit:

```
  11000000 . 10101000 . 00000001. 10010001
&
  11111111 . 1111 1111 . 1111 1111. 00000000
-----------------------------------------------
Rest2 = 192    .   168  .   1   .   0
```

Conclusion:

- Si **Rest1 est égal à Rest2** alors A et B se trouvent dans le meme réseau 192.168.1.0/24
- Si **Rest1 est différent de Rest2** alors A et B ne se trouvent pas dans le meme réseau donc la communication n'est possible que par l'intermédiaire d'une passerelle (routeur).



Remarque

Il est important de noter que la connaissance du masque de réseau permet de déterminer la partie netID et la partie hostID d'une adresse IP.

La longueur de la partie netID de l'IP est donnée par le nombre de bits à 1 dans le masque de réseau

Classification des adresses IP

On définit la notion de classe d'une adresse IP en utilisant le **premier octet** de cette adresse :

On dit qu'une adresse IP est de :

- classe A si la valeur de son **premier octet** en bit est de la forme : **0XXXXXXX**
  Le premier bit du premier octet est **0**

- Classe B si la valeur de son **premier octet** en bit est de la forme**:** **10XXXXXX**
  Les deux premiers bits du premier octet de l'adresse IP sont respectivement **1** et **0**.

- Classe C si la valeur de son premier octet en bit est de la forme : **110XXXXX**
  les trois premiers bits du premier octet de l'adresse IP sont respectivement **1,1** et **0**

- **Classe D si la valeur de son premier octet en bit est de la forme:** **111XXXXX**

Les trois classes d'adresses les plus couramment utilisés sont **A, B** et **C**

adresses potentielles de réseaux par classe :

classe	Adresse Minimale	Adresse Maximale
A	**1.0.0.0**	**126.0.0.0**
B	**128.0.0.0**	**191.255.0.0**
C	**192.0.0.0**	**223.255.255.0**

Adresses IP particulières

Les adresses particulières suivantes sont à connaître

Diffusions locale et distante

• 255.255.255.255 : adresse de broadcast sur le réseau IP local (ne passe pas le routeur, traduit en broadcast ARP)

•

<NET_ID><111...111> : adresse de broadcast dirigée vers le réseau de numéro NET_ID

(exemple : 132.227.255.255 = diffusion dans le réseau 132.227.0.0 traduit en broadcast ARP par le routeur destination) !

Rebouclage local (loopback) : 127.x.y.z

- généralement 127.0.0.1 (localhost)
- permet de tester la pile TCP/IP locale sans passer par une interface matérielle

l'adresse 0.0.0.0 est utilisé par le protocole RARP par les terminaux sans disque au démarrage et aussi utilisé comme adresse de route par défaut par les routeurs.

On rappelle les formules mathématiques suivantes :

$$1+q^2+q^3+.....+q^k=\frac{q^{(k+1)}-1}{q-1}$$

où **k** est un entier naturel et **q** un réel différent de **1**

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8
1	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**

Adresse IP privée

- Les adresses dites privées ne sont jamais utilisées sur Internet, car non « **routées** ». C'est-à-dire qu'aucun paquet d'un ordinateur possédant une adresse privée ne sera transmis aux autres ordinateurs.

- Dans le cas où on met un réseau à usage privé (non connecté à internet), choisir de préférence les adresses réservées à l'usage privé :
  - 10.0.0.1 à 10.255.255.254
  - 172.16.0.1 à 172.31.255.254
  - 192.168.0.1 à 192.168.255.254 (les plus courantes)
- Le nombre d'adresses IP est devenu trop faible par rapport au nombre de terminaux pouvant être connectés à internet on fait recours aux adresses IP privées et la translation d'adresses pour permettre aux machines ayant des adresses IP privées d'**accéder** à internet sans pour autant que les paquets transmis aient comme adresse source de type privé.
- En effet à la sortie du réseau privé un routeur remplace l'adresse IP source des paquets par des adresses IP routables avant d'envoyer les paquets sur internet: c'est le principe de translation d'adresse appelé **NAT (Network Address Translation).**

Remarque

Le nombre d'adresses IP étant devenu trop faible, compte tenu de la multiplication des utilisateurs, la solution de transaction d'adresse est utilisé aujourd'hui par les boxs fournis par les FAI pour la connexion Internet.

Aujourd'hui on se rend compte que le passage à IPv6 semble être la solution la plus pérenne avec son adressage codé sur 128 bits au lieu de 32 bits en IPv4

Sous-réseaux IP : motivation

- dans la version d'origine d'IP, une adresse IP est constituée d'un NETID et d'un HostID dans ce réseau
- le routage classique (sans masque) n'utilisait que le NETID

> à chaque réseau physique doit être affectée une adresse de réseau unique

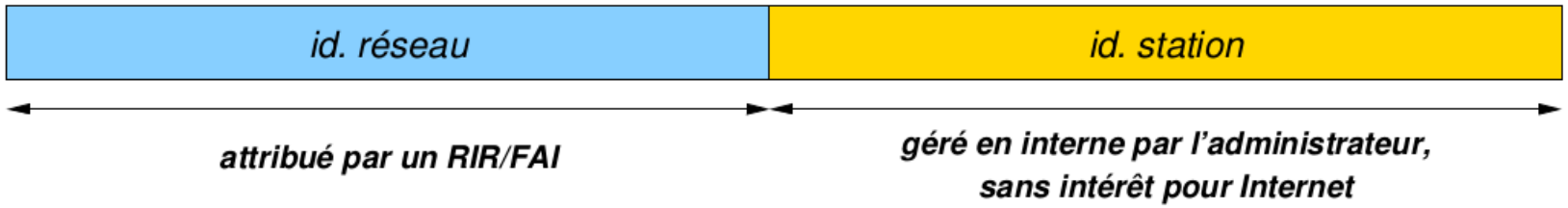
- or, la multiplication des réseaux pose plusieurs problèmes :
  - la gestion de toutes les adresses de réseau devient très lourde
  - les tables de routage deviennent gigantesques

- le schéma d’adressage peut être saturé

il est devenu nécessaire de réduire le nombre de réseaux à gérer, notamment en permettant à plusieurs réseaux physiques de partager la même adresse de réseau

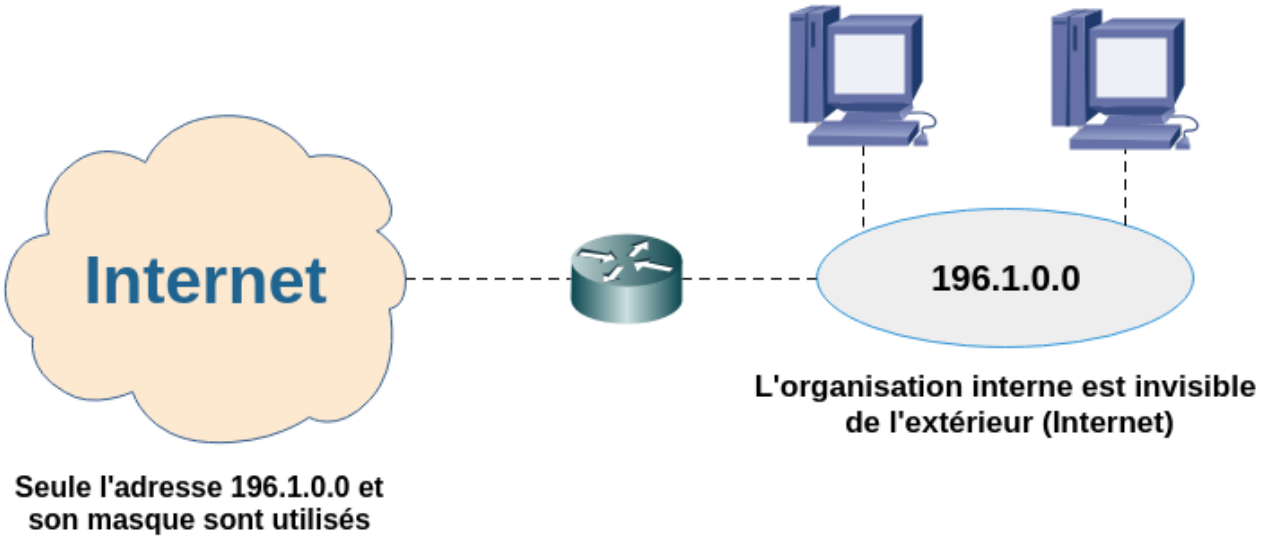
Pour y parvenir on procéde comme suit :

- pour connecter un réseau à Internet, un administrateur demande une adresse de réseau publique (associée à un masque)
- l’organisation interne du réseau est à la charge de l’administrateur :
  - plan d’adressage (affectation des adresses IP). En particulier, l’administrateur utilise comme il le souhaite la partie id. station :



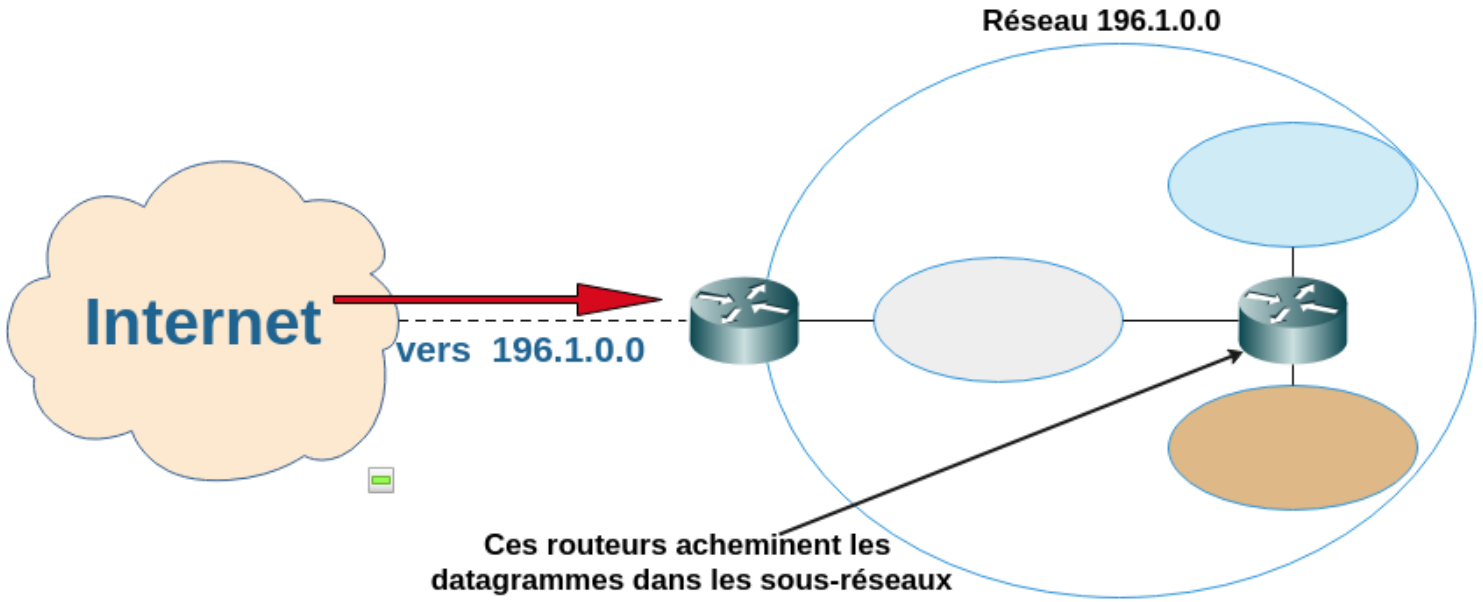
  - définition des routes sur les ordinateurs et routeurs de ce réseau

- du point de vue (des autres réseaux et routeurs) d’Internet, seule l’adresse du réseau est prise en compte



Principe

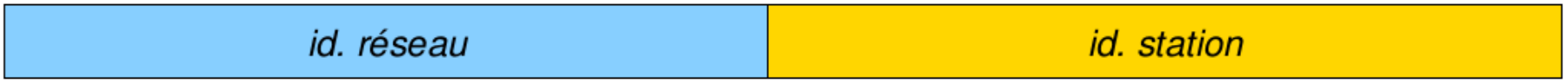
- l’administrateur dispose d’une adresse de réseau (telle que 196.1.0.0) mais de plusieurs réseaux physiques, appelés sous-réseaux
- la présence de plusieurs réseaux physiques est une question interne
- les routeurs d’Internet se contentent d’acheminer les datagrammes vers "le réseau” 196.1.0.0
- à charge des routeurs internes d’acheminer les datagrammes à travers les sous-réseaux



identifiant de sous-réseau

Pour distinguer les sous-réseaux, l'administrateur réserve une partie de Host-ID, appelée l'identifiant de sous-réseau :

un seul réseau (pas de sous–réseau)



avec des sous–réseaux

- les stations et routeurs internes doivent en tenir compte pour leurs décisions de routage (accessibilité directe ou indirecte de la destination)
- la taille de l'id. sous-réseau dépend du nombre de sous-réseaux, notamment si on suit la recommandation obsolète suivante :
  - l'id. sous-réseau tout à 0 est réservé (conflit avec l'adresse du réseau)
  - l'id. sous-réseau tout à 1 est réservé (conflit avec l'adresse de diffusion dirigée)

d'où : $2^n - 2$ sous-réseaux avec n bits pour l'*id. sous-réseau*

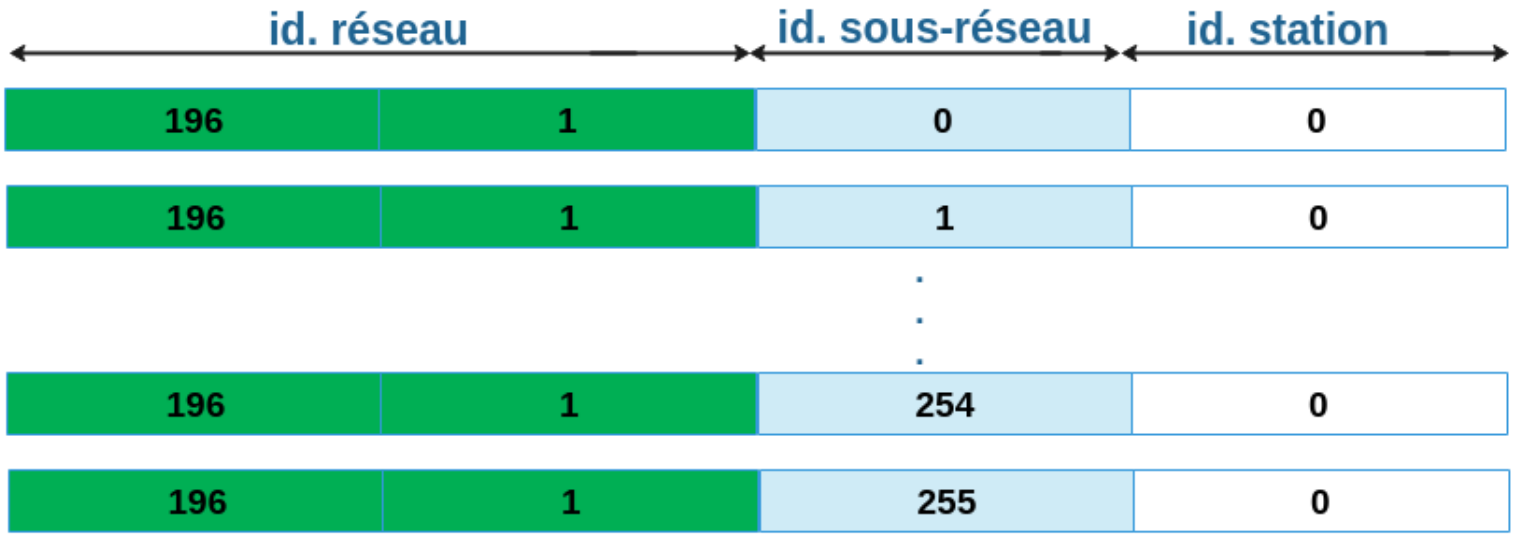
De nos jours, le all-zeros subnet et le all-ones subnet sont utilisables. . .

Exemples d’identifiant de sous-réseau

Dans le réseau 139.124.0.0 :

- si on utilise un octet pour l’id. sous-réseau, on peut avoir jusqu’à 256 sous-réseaux d’au plus 254 stations.

  Les sous-réseaux auront pour adresses :



si on n’utilise que 3 bits, on peut avoir jusqu’à 8 sous-réseaux d’au plus 8 190 stations.

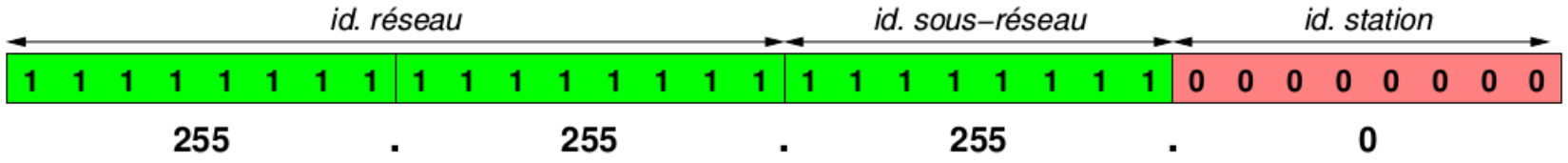
Les sous-réseaux auront pour adresses :

196.1.0.0, 196.1.32.0, 196.1.64.0, 196.1.96.0, 196.1.128.0, 196.1.160.0, 196.1.192.0 et 196.1.224.0

Masques de sous-réseaux

- en interne, pour identifier le sous-réseau, il faut prendre en compte la partie id. sous-réseau
- or, le masque du réseau (ou sa classe) ne permet que d'extraire la partie id. réseau
- d'où l'usage généralisé des masques de sous-réseaux :
  - ses bits à 1 indiquent où se trouvent les parties id. réseau et id. sous-réseau
  - ses bits à 0 indiquent où se trouve la partie id. station

exemple : pour le réseau 196.1.0.0 et l'id. sous-réseau sur un octet, le masque est 255.255.255.0 (autre notation /24) :



flexibilité du sous-adressage

- il est possible de regrouper plusieurs entrées de la table de routage en une seule, en jouant sur le masque associé (agrégation d'adresse)
- on n'est pas limité au subnetting d'un réseau : le subnetting d'un sous-réseau est tout aussi possible
- les exemples précédents n'ont montré que le subnetting avec des sous-réseau de taille égale (la taille de la partie id. sous-réseau était la même pour tous les sous-réseaux). Mais on peut tout aussi bien subnetter avec des sous-réseaux de taille variable par la technique du **sous-adressage variable** .

CIDR : Classless Inter-Domain Routing ou l'adressage hors classe

Idée générale

Prenons le cas d'une entreprise qui voulait avoir une plage d'adresse de classe B alors il en existe rarement.

La solution de lui attribuer 256 plages d’adresses de classes C mais ces 256 adresses nécessitent 256 entrées dans les tables de routage des routeurs d’internet.

La solution à ce problème est que les 256 plages d’adresses attribuées doivent être contigues pour être regroupées en une plage.

Par exemple, de 196.1.0.0 à 196.1.255.0 qui forment un bloc de adresses.

Ces 256 entrées peuvent être agrégées en seule entrée 196.1.0.0 de masque 255.255.0.0

En notation CIDR, cette entrée s’écrit 196.1.0.0/16

Conséquences:

- Les routeurs ne doivent plus utiliser les masques par défaut (par classe).
- les tables de routage doivent contenir obligatoirement un masque pour chaque destination (l’usage du format par classe est obsolète)

C’est ces deux principes énumérés ci-dessus qui constituent le principe de CIDR ou de routage hors classe.

Conséquence du routage CIDR :

- Si pour une adresse destination, plusieurs entrées peuvent correspondre alors c’est l’entrée ayant le préfixe le plus grand qui est retenue.

Par exemple si on a deux entrées dans la table de routage d’un routeur

196.1.0.0/14	192.168.1.1
196.1.0.0/16	**192.168.1.2**

Dans cet exemple c’est la deuxième route qui sera prioritaire.

5.4 Présentation du protocole ARP

- **L'adresse Ethernet** *(MAC)* est une **adresse unique** sur 48 bits *(6 octets)* associée à une carte Ethernet.

- Lorsqu'une machine A *(adresse EthA, adresse IPA)* veut émettre un paquet IP vers une machine B *(adresse IPB),* A doit connaître l'adresse Ethernet de B *(adresse EthB)* de façon à construire la trame Ethernet.

- Pour retrouver l’adresse Ethernet à partir de l’adresse IP du récepteur B, l'émetteur A utilise **le protocole ARP** *(Protocole de résolution d’adresse).*

Principe de résolution d’adresse par ARP

1. L'émetteur A envoie une **trame Ethernet** de diffusion (broadcast) contenant un message ARP « ARP Request » demandant **qui est adresse *IPB* ?**

2. Toutes les machines du réseau local reçoivent la requête. Seul B d'adresse adresse *IPB* se reconnaît, et elle répond à A *(adresse IPA)* dans une trame destinée à adresse *EthA*
« ARP Reply ».

3. La machine A retrouve l’adresse *EthB* de la machine B dans la trame Ethernet répondue.

4. Chaque machine maintient en mémoire une **table ARP de correspondances adresse *IP / adresse Eth*** pour éviter trop de requêtes *ARP*. Chaque entrée de la table à une durée de vie limitée.

Exemple:

1er cas :

A voulant envoyé une donnée à B, applique son masque de réseau à **son adresse IP** et à **l'adresse IP de B.** Il trouve la meme l'adresse **réseau 192.168.1.0/24** donc **A** et **B** se trouvent ds le meme réseau. Pour construire la trame à transmettre, A lance une requete ARP (**ARP request**) pour obtenir **l'adresse MAC de B** en fonction de l'adresse IP de la machine B. La machine B répond en fournissant son adresse MAC (**ARP replay**). A fabrique la trame aprés avoir obtenu de l'adresse MAC de B (@MAC-B) et transmet la trame ethernet à la machine B.

@MAC-B	@MAC-A	type	datagramme IP	crc

figure donnant le structure de la trame

On parle de **routage direct.**

2eme cas :

A voulant envoyer des données à C, applique son masque de réseau à **son adresse IP** et à **l'adresse IP de C.** Il trouve des adresses **réseaux différentes** qui sont **192.168.1.0/24 et 172.16.0.0/24** donc **A** et **C** se trouvent dans des réseaux **différents**. Dans ce cas la trame va etre envoyée vers la passerelle. Pour cela A lance une requete ARP (**ARP request**) pour obtenir **l'adresse MAC de l'interface f0/0** du routeur en fonction de l'adresse IP de l'interface f0/0. Le routeur répond en fournissant son adresse MAC (**ARP replay**). A fabrique la trame aprés obtention de l'adresse MAC de l'interface f0/0 du roueur (@MAC-f0/0) et lui transmet.

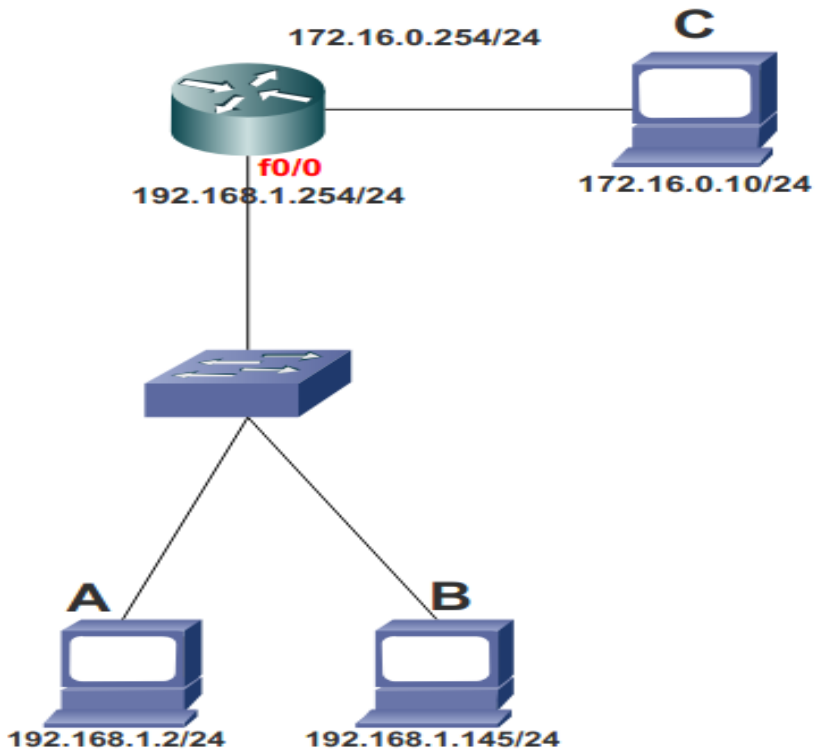
@MAC-f0/0	@MAC-A	type	datagramme IP	crc

figure donnant le structure de la trame

La trame arrive au niveau du routeur, il désencapsule et récupére le datagramme IP et il regarde l'adresse IP destination et si le réseau desinataire est directement connecté au routeur, il lance une requete ARP pour obtenir l'adresse MAC de C et fabrique la trame en y ajoutant le datagramme IP desencapsulé et l'envoie à C.

@MAC-C	@MAC-f0/0	type	datagramme IP	crc

On parle de **routage indirect**



Message ARP

Le message ARP est encapsulé dans une trame Ethernet comme le montre la figure 5.9 :

figure 5.9 Encapsulation d'un message ARP dans une trame ARP

La figure ci-dessous décrit les différents champs du message ARP.

Il faut noter que la différence qu'il y a entre la requête ARP (ARP Request) et la réponse ARP (ARP Reply) se situe au niveau du code d'opération vaut **1** pour la premiere et **2** pour la seconde.

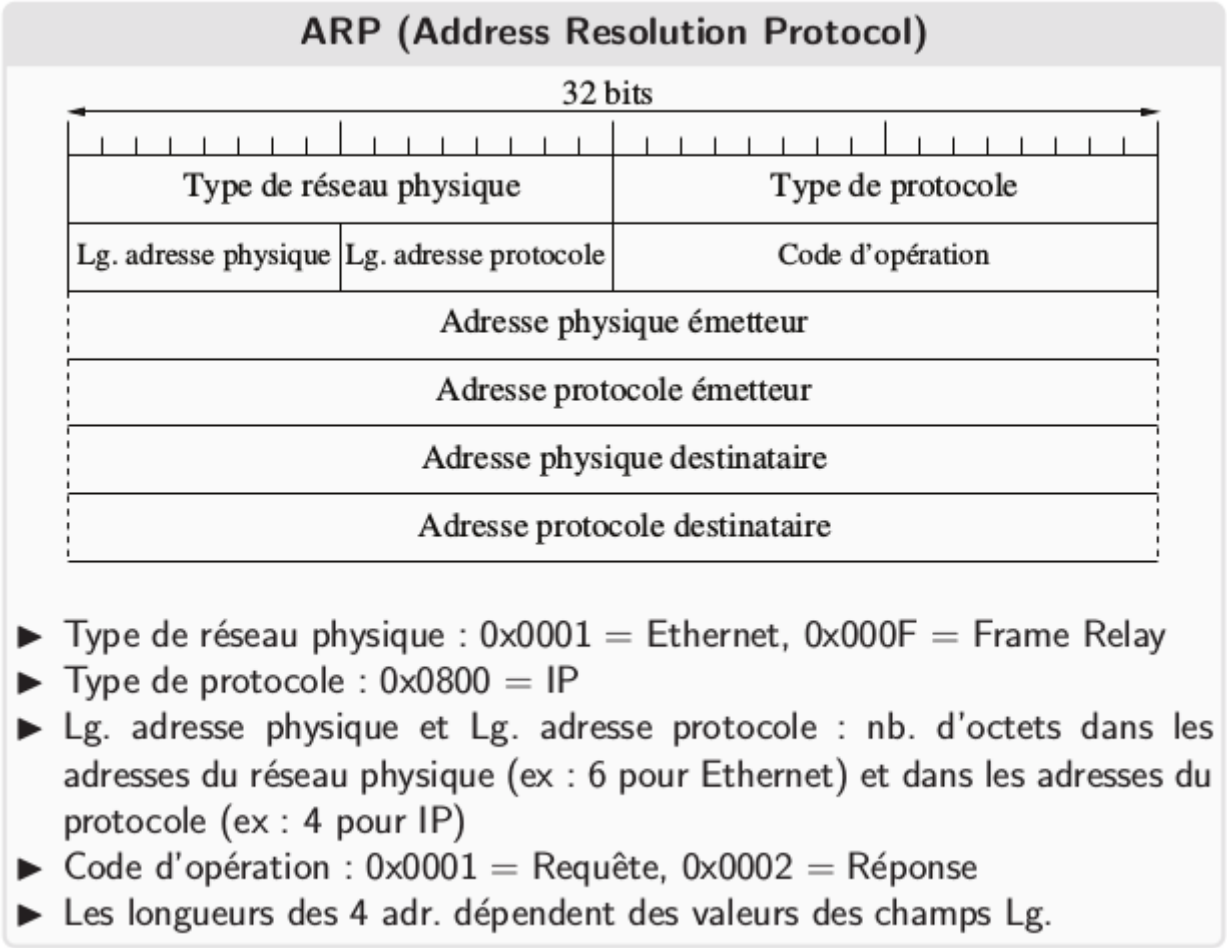


Figure 5.1O Structure d'un méssage ARP

Il est à noter que la table ARP d'une machine est fabriquée de manière dynamique et régulièrement mise à jour.

La table ARP

Tout envoi de paquet devrait théoriquement être précédé d'un échange ARP pour que l'émetteur connaisse l'adresse physique du destinataire.

Cela génère beaucoup de trafic sur le réseau local

C'est pourquoi chaque machine construit une table ARP d'associations (adresse IP, adresse MAC) qu'elle consulte si elle ne contient pas déjà l'information par rapport à un destinataire avant d'émettre une nouvelle requête ARP.

À la réception d'une réponse ARP, l'adresse MAC est ajoutée à la table.

Pour voir la table ARP sous linux

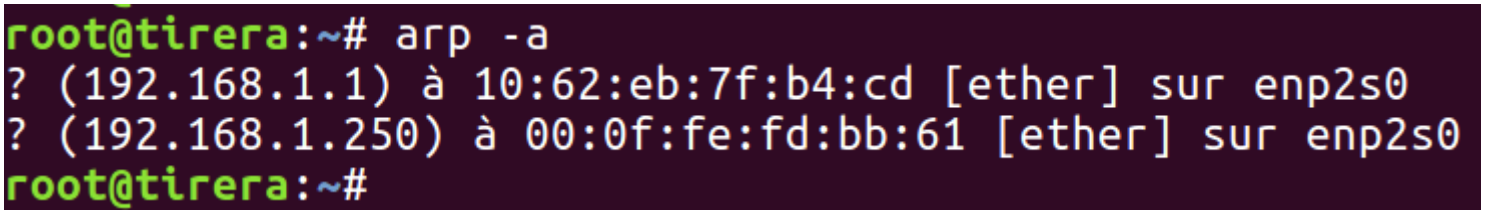


figure 5.11 Affichage d'une table ARP sous Linux

5.6 Présentation du protocole ICMP

- Le protocole ICMP est un protocole qui permet de gérer les informations relatives aux erreurs des machines connectées. Etant donné le peu de contrôles que le protocole IP réalise, il permet non pas de corriger ces erreurs mais d’en avertir les couches voisines.

- Ainsi, le protocole ICMP est utilisé par tous les routeurs, qui l'utilisent pour signaler une erreur.

Le message ICMP

Bien qu'il soit à un niveau équivalent au protocole IP un message ICMP est néanmoins encapsulé dans un datagramme IP comme le montre la figure 5.12

Figure 5.12 Encapsulation d’un message ICMP dans un datagramme IP en vue de son transfert dans un réseau local

Le champ ICMP est donné dans le figure 5.13

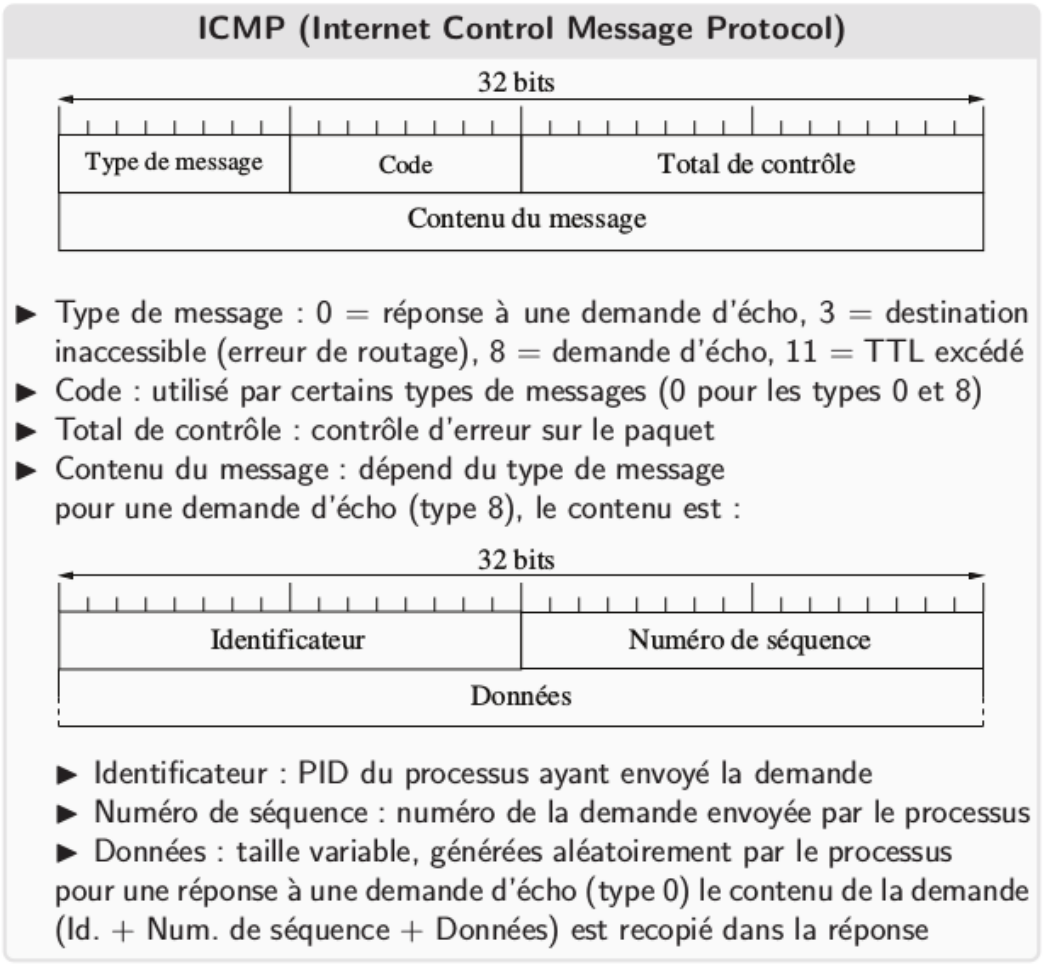


figure 5.13 Format d'un message ICMP

La commande **ping**

- Elle utilise les **paquets ICMP** de demande et de réponse d'écho afin de déterminer si une machine d'un réseau fonctionne.

- L'utilitaire **ping** est utilisé pour **diagnostiquer** les défaillances au niveau d'un réseau IP ou des routeurs.

Les 2 tableaux ci-dessous donnent respectivement quelques **Type et code** et **leur signification.**

Tableau 5.1 tableau de type

valeur (décimal)	signification
0	réponse à une demande d'echo
3	destination inacessible
4	limitation de production à la source
5	redirection (changement de route)
8	demande d'echo
9	annonce de routeur
10	sollicitation de routeur
11	TTL de datagramme expiré
12	problème de paramètre d'un datagramme
13	demande d'horodatage
14	réponse à une demande d'horodatage
17	demande de masque de sous–réseau
18	réponse à une demande de masque de sous–réseau

tableau 5.2 tableau de codes ICMP

valeur (décimal)	signification
0	réseau inaccessible
1	ordinateur inaccessible
2	protocole inaccessible
3	port inacessible
4	fragmentation nécessaire mais bit Don't Fragment positionné
5	échec de routage à la source
6	réseau de destination inconnu
7	ordinateur de destination inconnu
8	ordinateur source isolé
9	communication avec le réseau de destination interdite par l'administrateur réseau
10	communication avec l'ordinateur destinataire interdite par l'administrateur réseau
11	réseau inacessible pour le service demandé
12	ordinateur inaccessible pour le service demandé

- **Ping @IP**

```
tirera@PDC:~$ ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
64 bytes from 192.168.1.5: icmp_seq=1 ttl=255 time=1.64 ms
64 bytes from 192.168.1.5: icmp_seq=2 ttl=255 time=0.809 ms
64 bytes from 192.168.1.5: icmp_seq=3 ttl=255 time=0.682 ms
64 bytes from 192.168.1.5: icmp_seq=4 ttl=255 time=0.678 ms
64 bytes from 192.168.1.5: icmp_seq=5 ttl=255 time=0.692 ms
64 bytes from 192.168.1.5: icmp_seq=6 ttl=255 time=0.672 ms
64 bytes from 192.168.1.5: icmp_seq=7 ttl=255 time=0.744 ms
^C
--- 192.168.1.5 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6001ms
rtt min/avg/max/mdev = 0.672/0.845/1.640/0.328 ms
```

Ping nom

```
tirera@PDC:~$ ping ec2lt.sn
PING ec2lt.sn (37.187.56.30) 56(84) bytes of data.
64 bytes from ns3317153.ip-37-187-56.eu (37.187.56.30): icmp_seq=1
64 bytes from ns3317153.ip-37-187-56.eu (37.187.56.30): icmp_seq=2
64 bytes from ns3317153.ip-37-187-56.eu (37.187.56.30): icmp_seq=3
64 bytes from ns3317153.ip-37-187-56.eu (37.187.56.30): icmp_seq=4
64 bytes from ns3317153.ip-37-187-56.eu (37.187.56.30): icmp_seq=5
64 bytes from ns3317153.ip-37-187-56.eu (37.187.56.30): icmp_seq=6
^C
--- ec2lt.sn ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5001ms
rtt min/avg/max/mdev = 101.683/129.330/188.918/35.689 ms
```

Chapitre 6 : Le routage IP

Objectifs spécifiques du chapitre 6 : Le routage IP

1. Comprendre et mettre en œuvre le routage direct
2. Comprendre et mettre en œuvre le routage indirect
3. Comprendre le caractère fastidieux de routage statique quand un réseau dispose un nombre élevé de routeurs
4. Savoir utiliser et mesurer le rôle du protocole interne RIP en routage
5. Savoir utiliser et mesurer le rôle du protocole interne OSPF en routage

Sommaire

6.2 Le routage statique ou dynamique

Le routage IP

- Le routage est une fonction essentielle qui consiste à choisir le chemin pour transmettre un **datagramme IP** à travers les divers réseaux.

- On appelle **routeur** un équipement relié à **au moins deux réseaux** (cet équipement pouvant être un ordinateur, au sens classique du terme, qui assure les fonctionnalités de routage).

- Un routeur ré-émettra des datagrammes venus d'une de ses interfaces vers une autre.

Le routage direct

- Le **routage direct** correspond au **transfert** d'un datagramme au sein d'un **même réseau.** La démarche suivante est suivie :

  - L'expéditeur vérifie que le destinataire final partage le **même réseau** que lui. On utilise pour cela le masque de sous réseau (principe d'application de masque de réseau aux adresses IP).

  - Si c'est le cas, le **routage direct** est suffisant :

    1) l'émetteur lance une requête ARP pour connaître l'adresse MAC du destinataire à partir de son adresse IP.

    2) encapsule le datagramme à transférer dans une trame Ethernet et le transmet dans le réseau.

    3) L'émission se fera en **encapsulant le datagramme** dans une **trame Ethernet**.

Le **routage indirect**

Le **routage indirect** est mis en œuvre dans tous les autres cas, c'est-à-dire quand au moins un **routeur** sépare **l'expéditeur initial** et le **destinataire final**. La démarche suivante est suivie :

- L’expéditeur doit déterminer vers quel routeur envoyer un datagramme IP en fonction de sa destination finale.

- Ceci est rendu possible par l'utilisation d'une table de routage spécifique à chaque routeur et qui permet de déterminer le prochain routeur destinataire pour transmettre le paquet.

Table de routage

- L'essentiel du contenu d'une table de routage est constitué de quadruplets :

  - **Réseau destination** : C’est l'adresse d'un réseau de destination.

  - **Passerelle (gateway)** : C’est l'adresse IP du prochain routeur vers lequel envoyer le datagramme pour atteindre cette destination

  - **Masque :** C’est le masque associé au réseau de destination

  - **Interface :** Cela désigne l'interface physique par laquelle le datagramme doit réellement être expédié.

Tableau 6.1 tableau de routage d’un routeur

Destination	Passerelle	Masque	Interface
C’est l’adresse IP du réseau de destination	*C’est l’adresse IP du prochain routeur vers lequel le est envoyé*	*C’est le masque associé au réseau de destination*	*Désigne l’interface physique (carte réseau) par laquelle datagramme doit réellement être expédié.*

Une table de routage peut contenir une route par défaut qui spécifie un routeur par défaut vers lequel sont envoyés tous les datagrammes pour lesquels il n'existe pas de route dans la table.

Tous les routeurs mentionnés dans une **table de routage** doivent être **directement accessibles** à partir du routeur considéré.

Aucune machine, ni routeur ne connaît le chemin complet de routage des paquets.

Chaque routeur connaît seulement le prochain routeur à qui le datagramme doit être envoyé.

Processus du routage

- Supposons qu’une machine A souhaite envoyer une information à une machine B.
- A compare la partie réseau de son adresse avec la partie réseau de l’adresse de B.

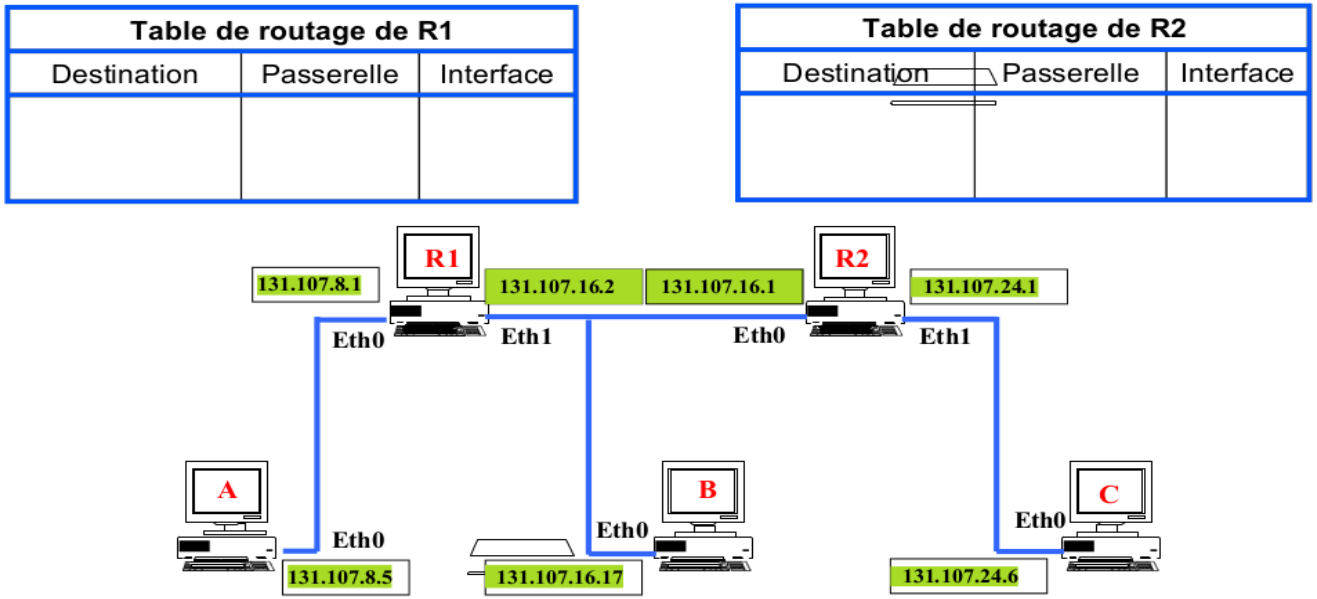
1. =>Si A et B font partie du même réseau (ou sous-réseau). A envoie son information directement à B (routage direct).

2. => Si A et B ne font pas partie du même réseau. A cherche, dans sa table de routage, la passerelle pour transmettre le datagramme.

3. => Si A n'a toujours rien trouvé dans sa table de routage, il envoie l'information à la passerelle par défaut (default gateway).

Exemple de routage direct et indirect

A (131.107.8.5 / 24) veut envoyer un paquet à C (131.107.24.6 / 24).



- A encapsule le datagramme IP à envoyer à C dans une trame ayant comme adresse MAC destinataire l'adresse MAC de l'interface Eth0 de **R1.**
- **R1** désencapsule la trame pour y retirer le datagramme IP émis par A et constate que celui-ci est destiné à C pour qu'il peut atteindre en passant par **R2.**
- **R1** encapsule le datagramme reçu de A dans une trame ayant comme destination l'adresse MAC de Eth0 de R2 et l'envoie dans le réseau.
- **R2** désencapsule la trame reçu de R1 retire le datagramme IP et constate que celui-ci est destiné à la machine C qui se trouve dans son réseau.
- **R2** crée une nouvelle trame et encapsule le datagramme IP venant de A et l'envoie dans le réseau commun avec C.

Le Routage statique

On dit qu’un routage est statique lorsque la table de routage est entrée manuellement par l’administrateur.

Dans le cas le routeur ne partage les informations manuellement entrées par l’administrateur avec les autres routeurs.

Lorsque un nouveau réseau est ajouté, il faut reconfigurer l'ensemble manuellement. De plus, pour prévenir tout dysfonctionnement (panne d'un routeur, ligne coupée, etc.), il faut effectuer une surveillance permanente et **reconfigurer chaque routeur** le cas échéant.

Plus un réseau est important, plus cette tâche devient fastidieuse.

Pour gérer la table de routage dans un réseau de grand taille on fait appel au protocoles de routages qui permettront aux routeurs d’échanger dynamiquement des informations sur tables de routage sans que l’administrateur n’intervienne.

Le routage dynamique

Quand on utilise un protocole de routage, le routeur construit lui-même sa table de routage en fonction des informations qu'il reçoit de ce protocole de routage.

Le routeur sélectionne la route la mieux adaptée à un paquet circulant sur le réseau en utilisant les informations d'état du réseau transmises d'un routeur à l'autre.

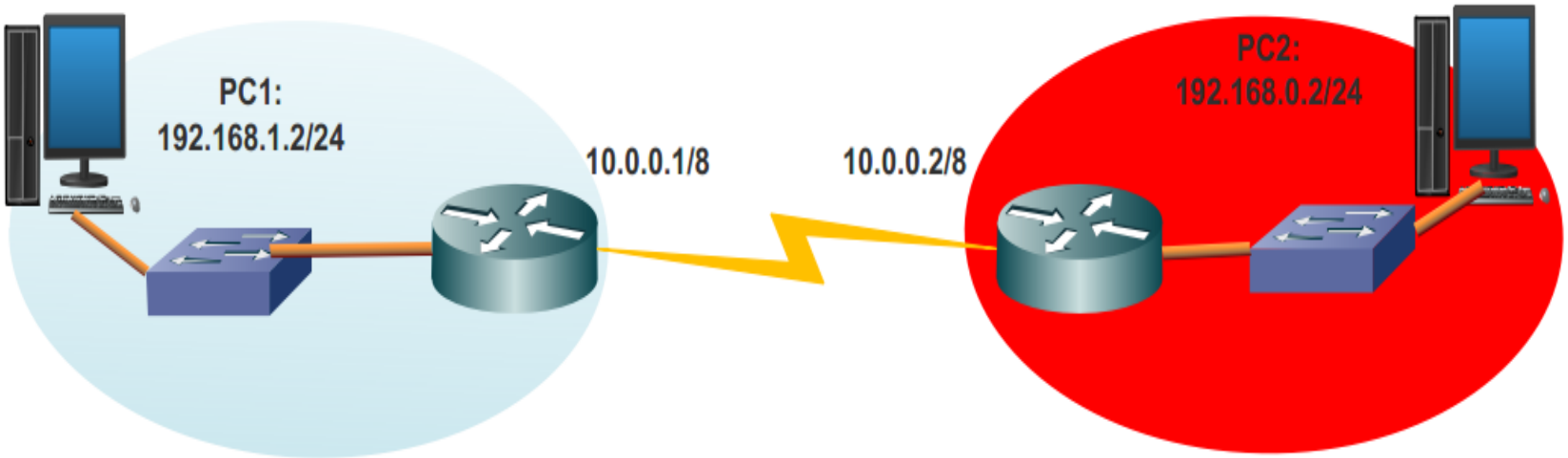


Figure 6.1 Echange d'information sur les sous-réseaux gérés par des routeurs utilisant des protocoles de routage

Le protocole de routage RIP

Avec RIP, un routeur transmet à ses voisins les adresses réseaux qu'il connaît ainsi que la distance pour les atteindre.

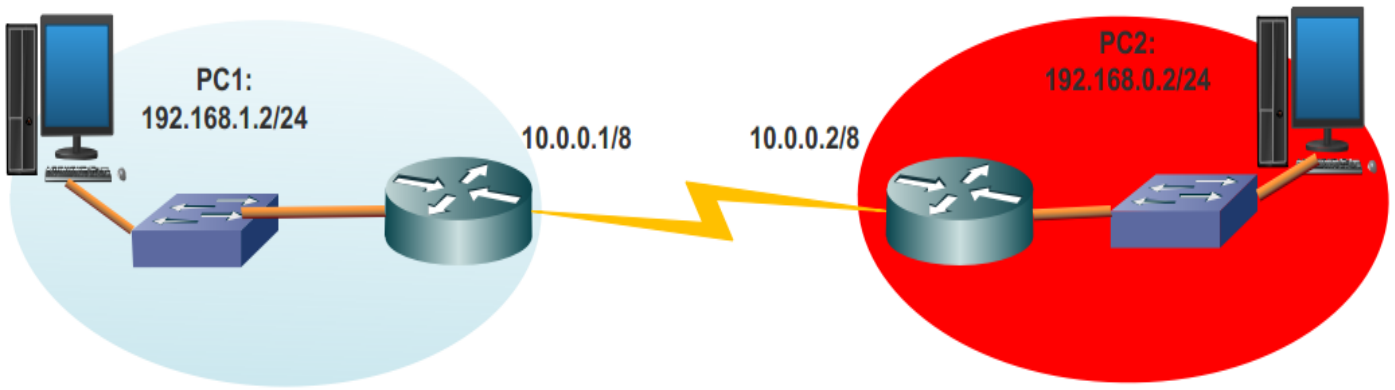
Ces couples **adresse/distance** sont appelés **vecteurs de distance.**

La métrique utilisée par RIP est la distance correspondant au nombre de routeurs à traverser (hop ou nombre de sauts) avant d'atteindre un réseau.

Si plusieurs routes mènent à la même destination, le routeur doit alors choisir **la meilleure route** vers une destination donnée.

Sur chaque routeur, si des routes redondantes apparaissent, il retient celle qui traverse **le moins de routeur.**

Exemple:
On appelle R1 le routeur qui est à gauche et R2 le routeur qui est droite.



- R1 transmet à R2 le vecteur de distance (192.168.1.0/24;1) et R2 transmet à R1 le vecteur de distance (192.168.0.0/24;1)
- Aucune information n'est transmise concernant le réseau 10.0.0.0/8, supposé connu par les deux routeurs.

R1 et R2 actualisent leurs tables avec les informations reçues.

Table de routage de R1

Destination	Passerelle	Interface	Distance
		Eth1	1
		Eth0	1
		Eth1	2

Table de routage de R2

Destination	Passerelle	Interface	Distance
10.0.0.0/8	x	Eth1	1
192.168.0.0/24	x	Eth0	1
192.168.1.0/24	10.0.0.1	Eth1	2

Chapitre 7 : Les protocoles de couche 4 (TCP, UDP…)

Objectifs spécifiques du chapitre 7: Couche 4, Les protocoles TCP et UDP

1. Comprendre l’identification des applications dans un réseau IP par couple (adresse IP, numéro du port)
2. Comprendre dans quel cas un service réseau à besoin du protocole TCP ou UDP
3. Comprendre la notion de socket
4. Savoir décrire les segments TCP et UDP
5. Comprendre et mettre en œuvre la translation d'adresse et de port

Sommaire

Introduction

Les protocoles de transports TCP et UDP ont été défini pour pallier aux limitations d'IP à savoir :

- la non garantie de livraison de datagramme IP
- les erreurs non signalées
- déséquencement possibles des datagrammes
- la non prise en charge des mécanismes de détection des erreurs sur les données
- la non gestion de contrôle de flux
- et surtout le non adressage des applications clients/serveurs

Ces protocoles de transport ont été définis pour pouvoir corriger les erreurs signalées par ICMP par pour celles qui ne sont pas signalées.

Sachant que plusieurs applications peuvent s'exécuter simultanément sur un même ordinateur, on utilise le numéro de port codé sur 16 bits pour identifier chaque application.

De manière précise l'adresse d'une application dans un réseau IP est le triplet (Adresse IP, Protocole de transport et numéro du port).

Quand la couche réseau reçoit un datagramme IP sur une machine destinataire, elle examine le champ protocole du datagramme IP pour savoir à quel protocole de transport remettre les données (17 si UDP, 6 si TCP).

Chaque protocole de transport consulte le port du service pour rediriger les données vers le bon service comme le montre la figure ci-après:

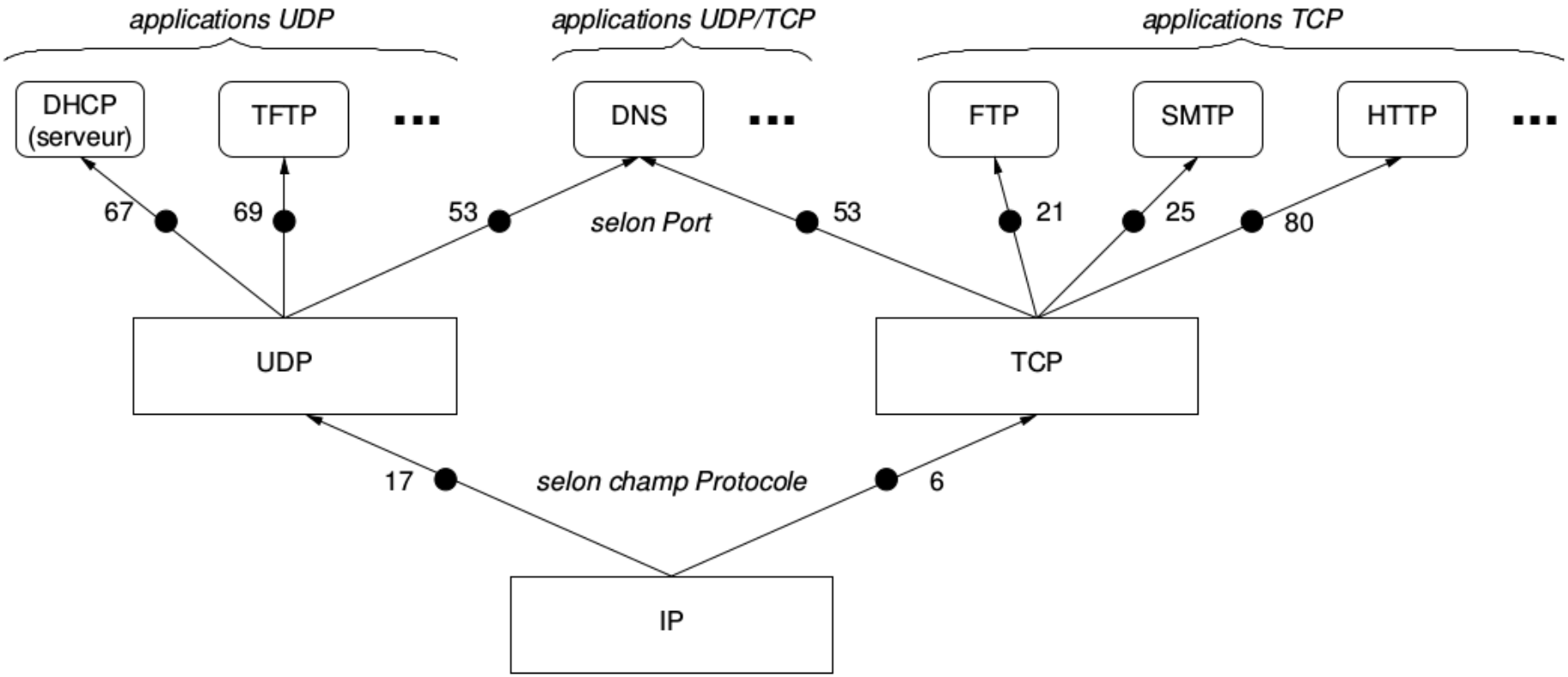


figure 7.1: Démultiplexage des ports au niveau IP et transport

7.1 Présentation du protocole UDP

Le protocole est défini dans RFC 768, fonctionne en mode non **connecté** (possibilité de pertes, déséquencement des messages, duplication, la non régulation des flux) et rend les services suivants :

- envoi/réception de messages entre applications (processus) à travers Internet
- adressage des applications par numéro de port
- multiplexage/démultiplexage par numéros de port
- contrôle facultatif de l'intégrité des données

NB: Les applications utilisant UDP doivent gérer elles mêmes les insuffisances de l'UDP.

7.2 Format des datagrammes UDP

Le datagramme UDP est constitué d'une en-tête de taille fixe de 8 octets et d'une partie données de taille variable de longueur maximale 65535 octets.

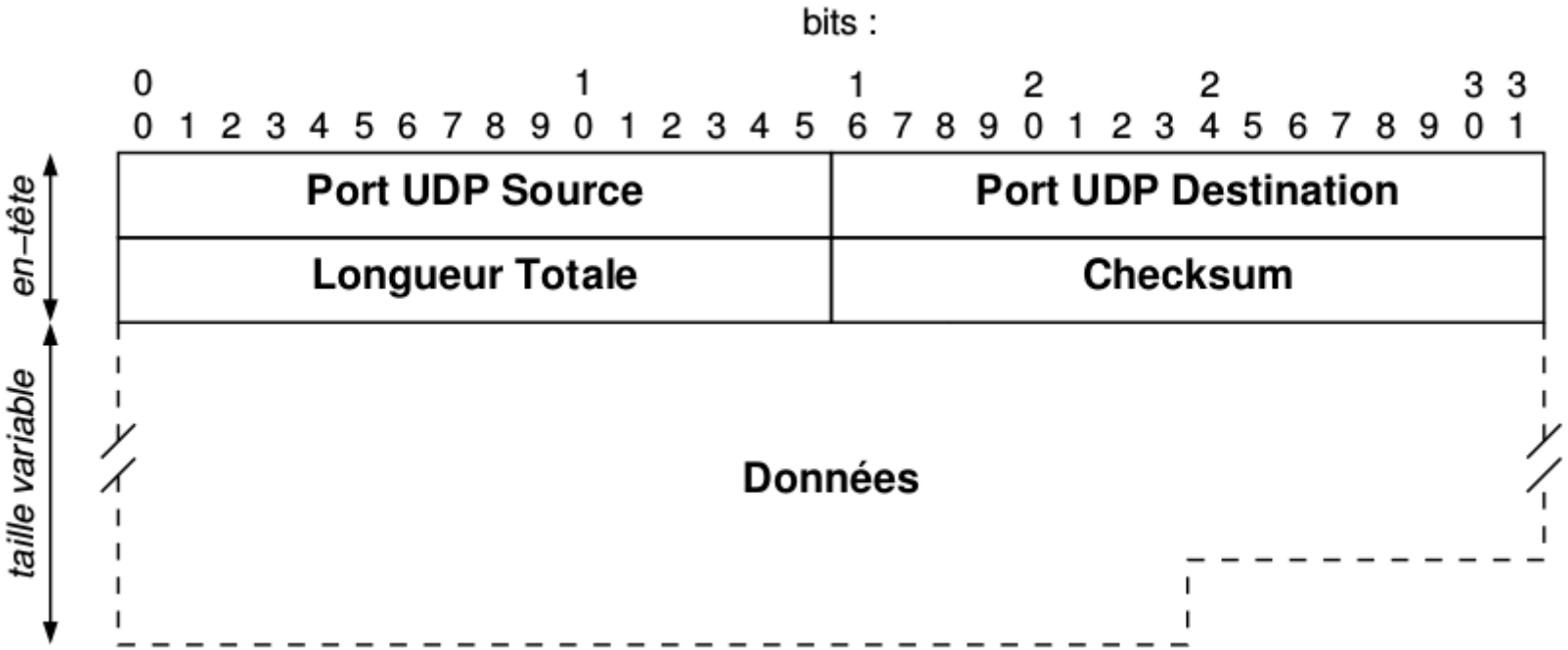


Figure 7.2 Formats des datagrammes UDP

- Le champ port UDP source indique le numéro de port de l'émetteur.

  Ce champ est mis à 0 si l'émetteur ne désire recevoir une réponse.

- Le champ port UDP destination : indique le numéro de port du destinataire.

- Si ce port n’a été alloué à aucun processus, UDP renverra un message ICMP de destination inaccessible car port non alloué (type 3, code 3) et détruit le datagramme.

UDP utilise le numéro du port pour démultiplexer les données et les envoyer vers les applications adéquates comme le montre la figure ci-après :

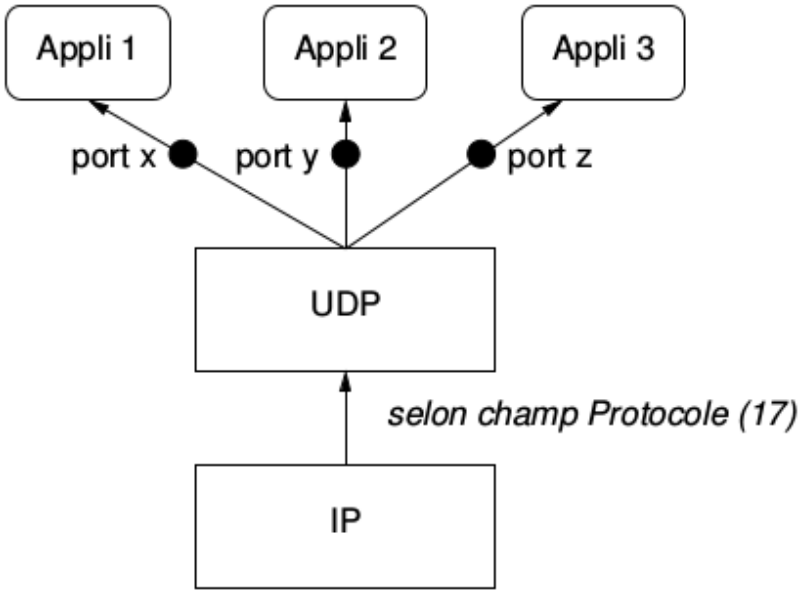


figure 7.3 Démultiplexage par UDP et transmission des données vers les applications adéquates

Certains ports UDP sont réservés pour des applications particulières

Exemple :

tableau 7.1: tableau d'application utilisant UDP et leur port

Num (décimal)	Application
7	Serveur echo
13	Serveur daytime
19	Serveur chargen
53	Serveur DNS
67	Serveur BOOTP/DHCP
68	Client BOOTP/DHCP
69	Serveur TFTP
123	Serveur NTP

- les ports [1024, 49151] sont enregistrés (mais peuvent être utilisés)
- les ports [49152, 65535] sont dits dynamiques et/ou à usage privé

Champs UDP : Checksum

Ce champ est facultatif et est fixé à zéro 0 si non calculé.

Ce champ permet de vérifier l'intégrité de la totalité du datagramme IP.

Il permet plus précisément de s'assurer que :

- Les données reçues sont correctes
- les ports sont corrects
- les adresses IP sont correctes

UDP utilise Pseudo en-tête sur 12 octets composé de champs comme le montre la figure ci-après :

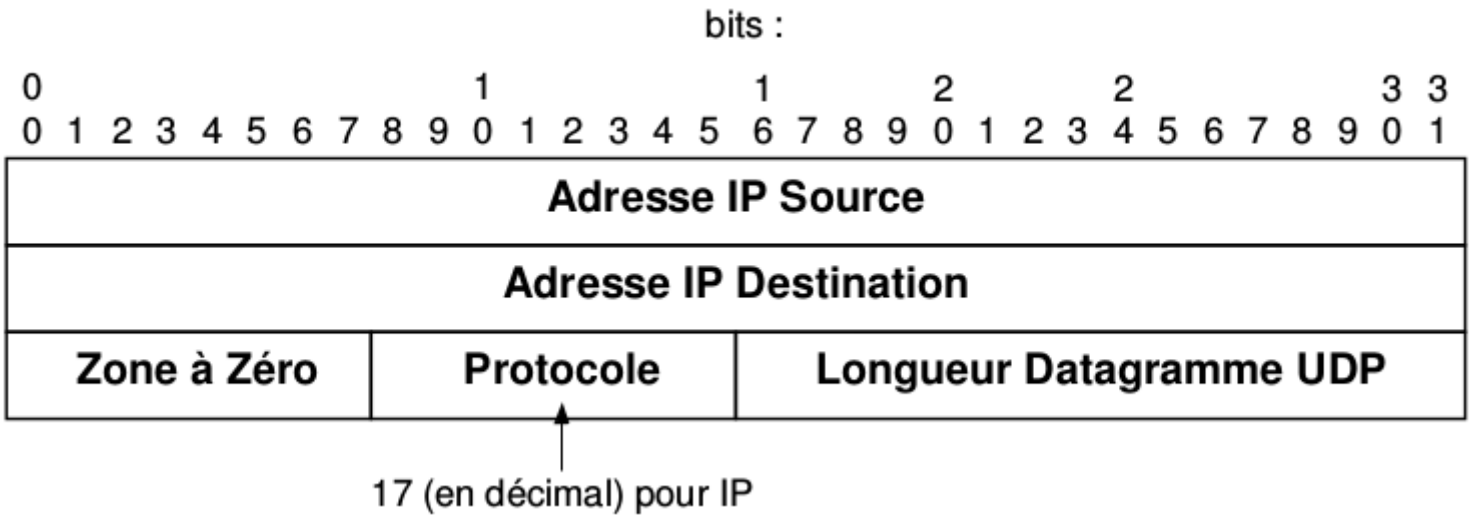


figure 7.4 Différents champ de l'en-tête UDP

Modèle Client/Serveur utilisant UDP

- Le serveur est démarré sur un ordinateur en écoute sur un port. Son adresse est le triplet : (adresse IP, UDP, port)

  L'écoute consiste à attendre qu'un message parvienne à ce port ; les clients devront connaître le port du serveur (intérêt des ports réservés)
- Le client envoie une requête au serveur (à son adresse) :
  - pour recevoir une réponse du serveur, il doit avoir acquis un numéro de port UDP. Le plus souvent, ce port est quelconque. Rares sont les clients (comme BOOTP/DHCP) qui nécessitent un port précis.
  - la requête est un message applicatif : suite d'octets constituant un PDU (Protocol Data Unit) du protocole qu'implémentent le client et le serveur.
  - UDP fabrique un datagramme UDP avec pour champ Données ce message, et pour champ Port Destination celui du serveur. Le Port Source est celui du client.
  - Le datagramme UDP est ensuite envoyé via un datagramme IP avec les adresses IP du serveur (destination) et du client (source).

- Le serveur reçoît le message du client, ainsi que son adresse. Il peut alors traiter le message et répondre au client.

- Selon le protocole, la discussion peut se poursuivre, ou s'arrêter là.

7.3 Le protocole TCP (Transport Control Protocol)

Le protocole de transport TCP est défini dans la RFC 793 corrigée par RFC 1122 et 1323, fonctionne en mode **connecté.**

Propriétés du service de TCP

Le protocole de transport TCP est un protocole **fiable** avec des propriétés suivantes**:**

- **Orienté connexion** : transfert de flots d'octets. La suite d'octets remise au destinataire est la même que celle émise.
- **Circuits virtuels** : une fois une connexion demandée et acceptée, les applications la voient comme un circuit dédié
- **Transferts tamponnés** : quelle que soit la taille des blocs de données émis par les applications, TCP est libre de les découper ou de les regrouper
- **Connexions non structurées** : pas de frontière placée par TCP entre les messages émis par les applications
- **Connexions full-duplex :** les données s'échangent dans les deux sens (mais un côté peut libérer un sens de transmission quand il n'a plus de données à émettre)

Adresse d’application, port et connexion

- Comme pour UDP, l’adresse d’une application est un triplet (adresse IP, TCP, port) : le serveur et le client doivent en posséder une. Le port du client est généralement quelconque.
- Mais à la différence d’UDP, on ne peut envoyer un message directement à une adresse : il faut que le client établisse une connexion avec le serveur. Ils ne peuvent échanger des messages que via une connexion
- **Établissement d’une connexion :**
  - Serveur : effectue une ouverture passive en écoutant sur un port, c’est à dire en demandant un port et en attendant qu’un client s’y connecte.
  - Client : effectue une ouverture active en demandant l’établissement d’une connexion entre son adresse et celle du serveur. Le serveur doit être en écoute. Les modules TCP du client et du serveur intéragissent pour établir cette connexion.
- Une fois la connexion établie, le serveur et le client doivent l’utiliser pour en voyer/recevoir des messages. TCP est chargé d’assurer la fiabilité de la connexion (notamment s’occupe des acquittements/ retransmissions)

Serveurs et ports réservés TCP

Certaines applications bien connues ont des ports TCP réservés [0, 1023].

Exemples :

Tableau 7.2 : tableau d'application utilisant TCP et leur port

Num (décimal)	Application
7	Serveur echo
13	Serveur daytime
20	Serveur FTP (données)
21	Serveur FTP (commandes)
22	Serveur SSH
23	Serveur TELNET
25	Serveur SMTP (transfert de mail)
53	Serveur DNS
80	Serveur HTTP (www)
119	Serveur NNTP (news)

Ports et connexions

- Plus complexe qu'UDP car un port peut être utilisé pour plusieurs connexions simultanément :
  - un serveur peut accepter plusieurs clients à la fois : chaque appel d'accept() retourne une nouvelle connexion utilisant le port du serveur
  - plus rare, un client peut aussi utiliser son port pour établir plusieurs connexions (mais pas versla même adresse serveur)
- En dehors des SAP d'ouverture passive, TCP gère surtout des "objets" connexion
- Une connexion est identifiée par le quadruplet formé avec l'adresse de ses deux extrémités :

  (adresse IP locale, port local, adresse IP distante, port distant)
- Les connexions sont gérées indépendamment les unes des autres Chaque connexion dispose de ses propres tampons en émission/réception et de chaque côté.

Flots d’octets et segments

- pour TCP une connexion sert à transmettre des flots d’octets dans les deux sens
- les flots sont transmis par des segments (PDU de TCP)
- un segment est transmis dans un seul datagramme IP (sauf fragmentation pendant l’acheminement)
- l’émetteur confie à (son) TCP des blocs de données de taille quelconque
- le récepteur récupère des blocs de données de taille quelconque
- mais le nombre d’octets transportés par un segment est décidé par TCP :
  - pour des raisons d’efficacité
  - pour la régulation de flux

Flots d’octets et segments : exemple

A chaque port TCP on associe 2 fils d’attente à l’émission (E) et la réception (R) comme le montre la figure 7.5

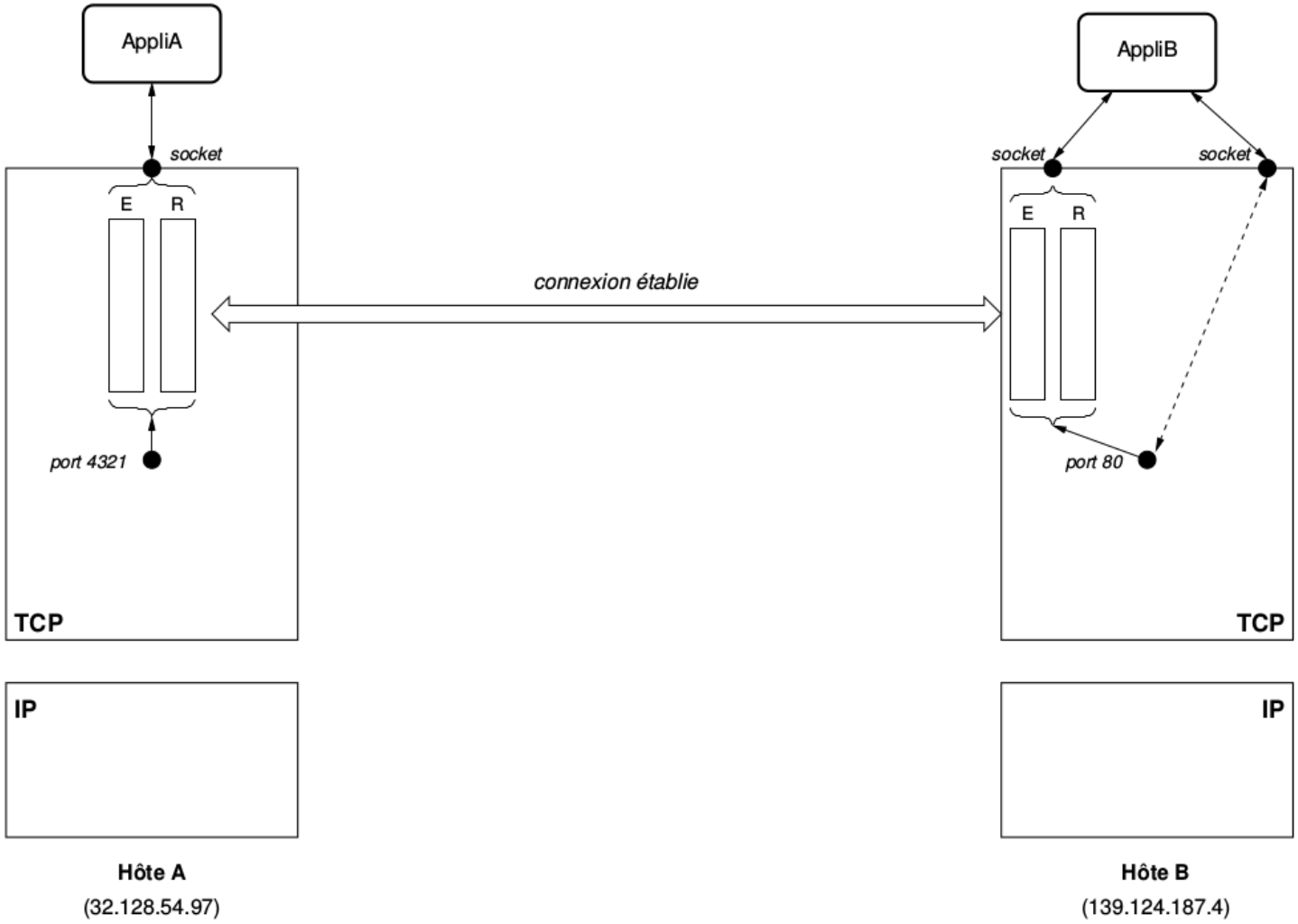


Figure 7.5 Files d’attente associées à un port TCP

Supposons que l'application A a des données à transmettre à l'application B, après établissement de connexion entre les deux applications, l'application A utilise la commande **send**() pour envoyer une donnée dans la file d'attente (E)

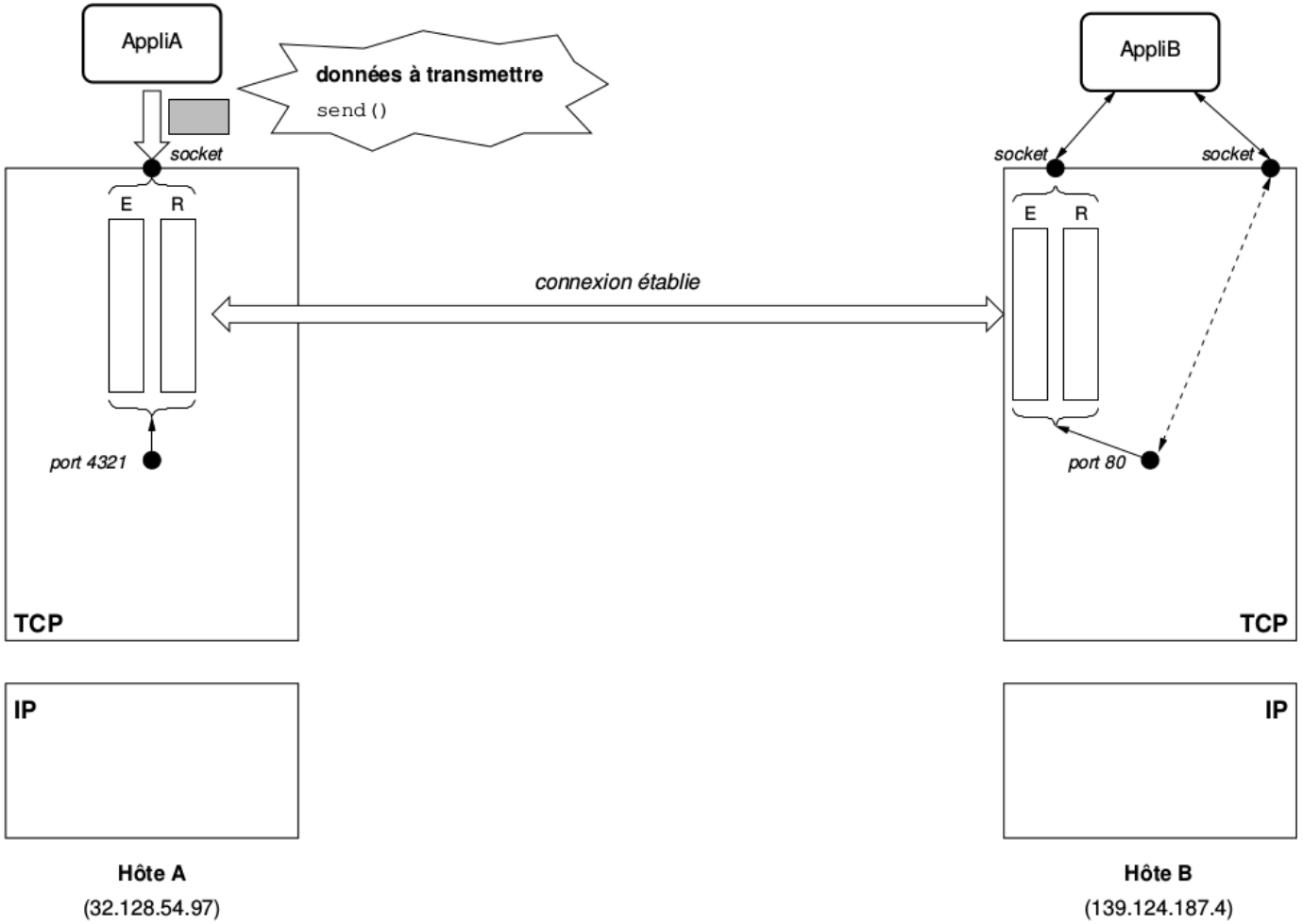


Figure 7.6 Début de remplissage d'une file d'attente à l'émission

La file d'attente (E) n'étant pas encore pleine, l'application A va envoyer toujours par la commande **send()** d'autres dans la file d'attente (E) comme le montre la figure 7.7

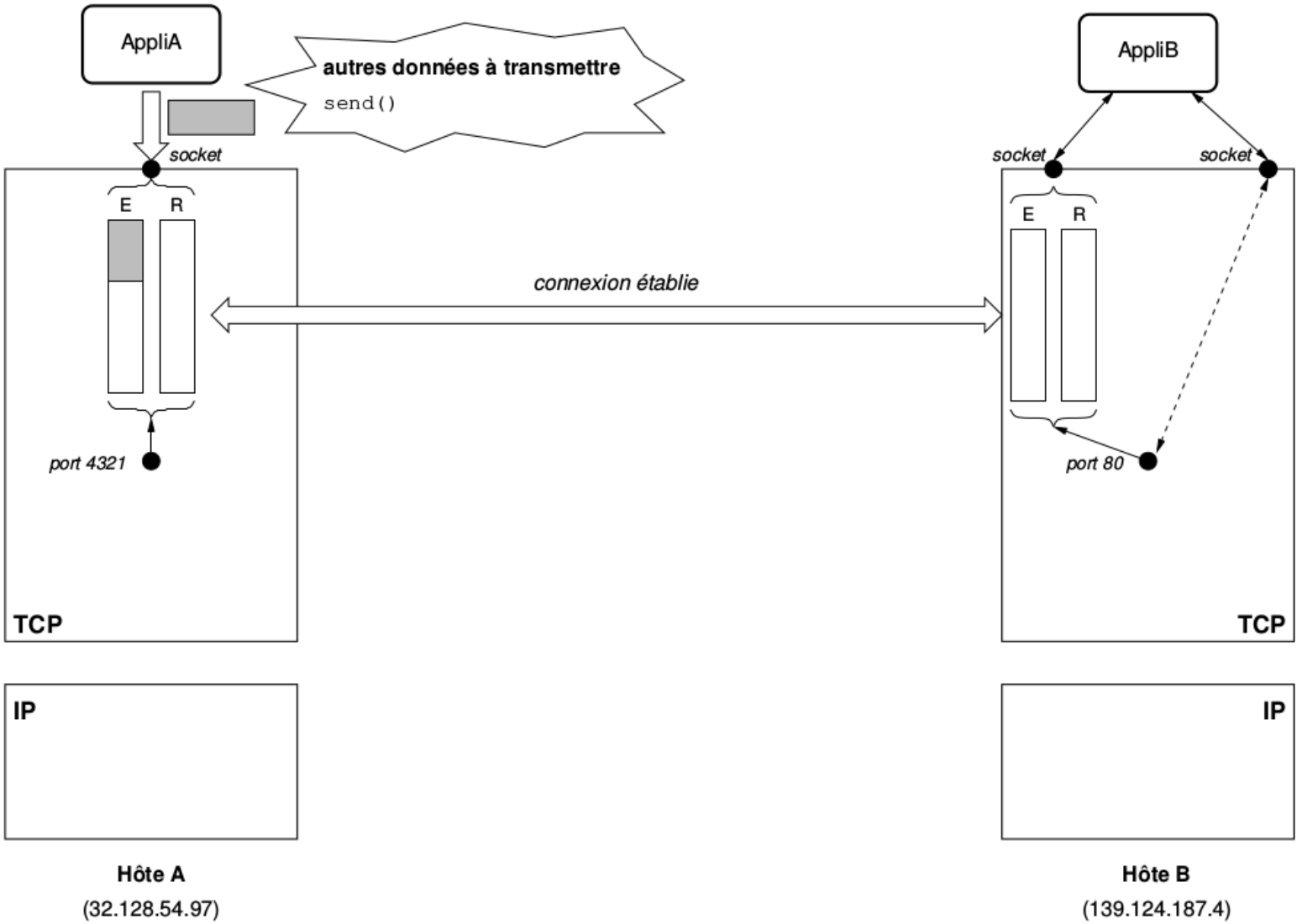


Figure 7.7 Ajout d'autres données sur la file d'attente

On obtient la figure 7.8

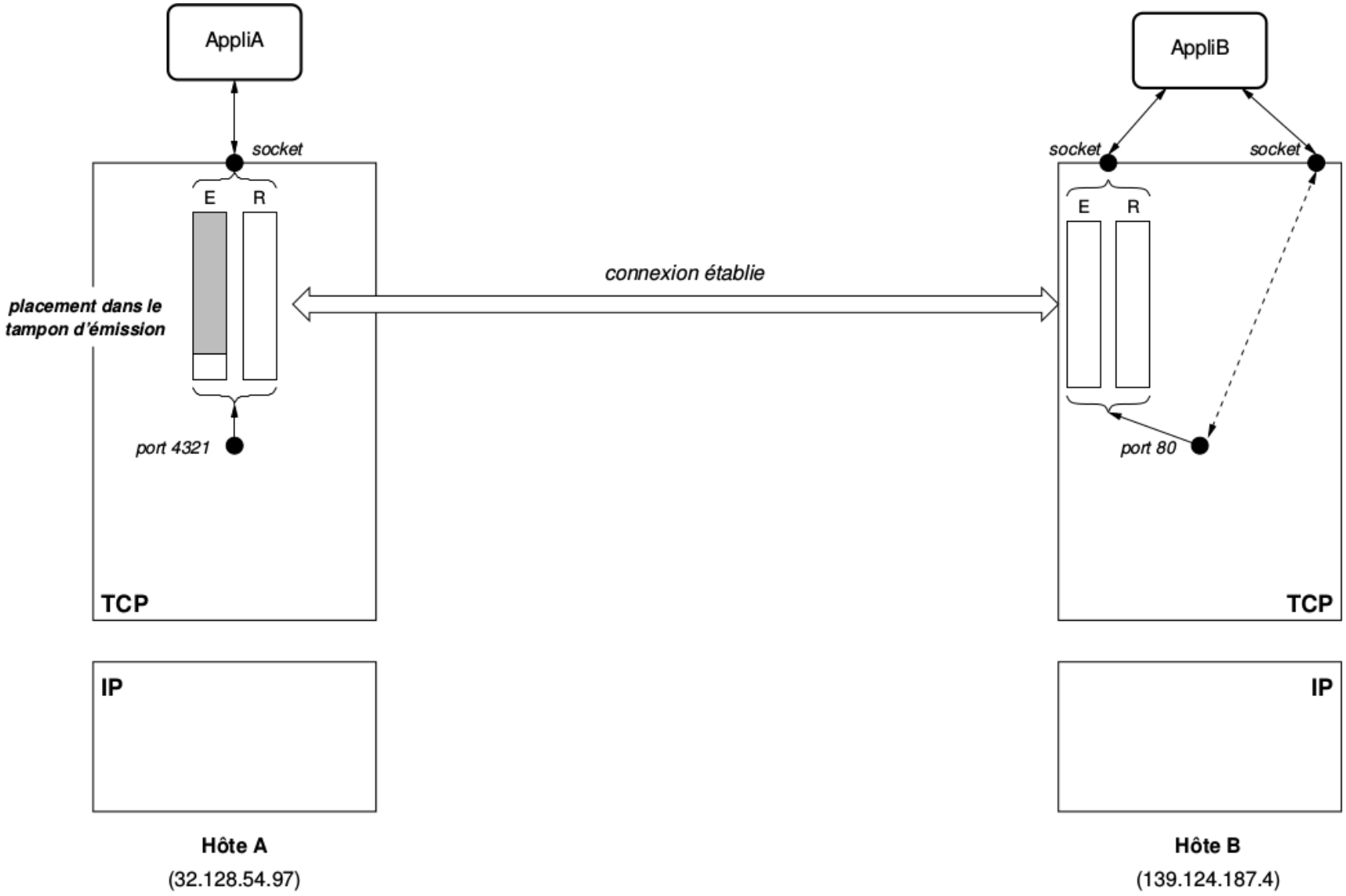


figure 7.8 Remplissage d'un segment de données

L’application A décide alors d’envoyer un segment à l’application B comme le montre la figure 7.9

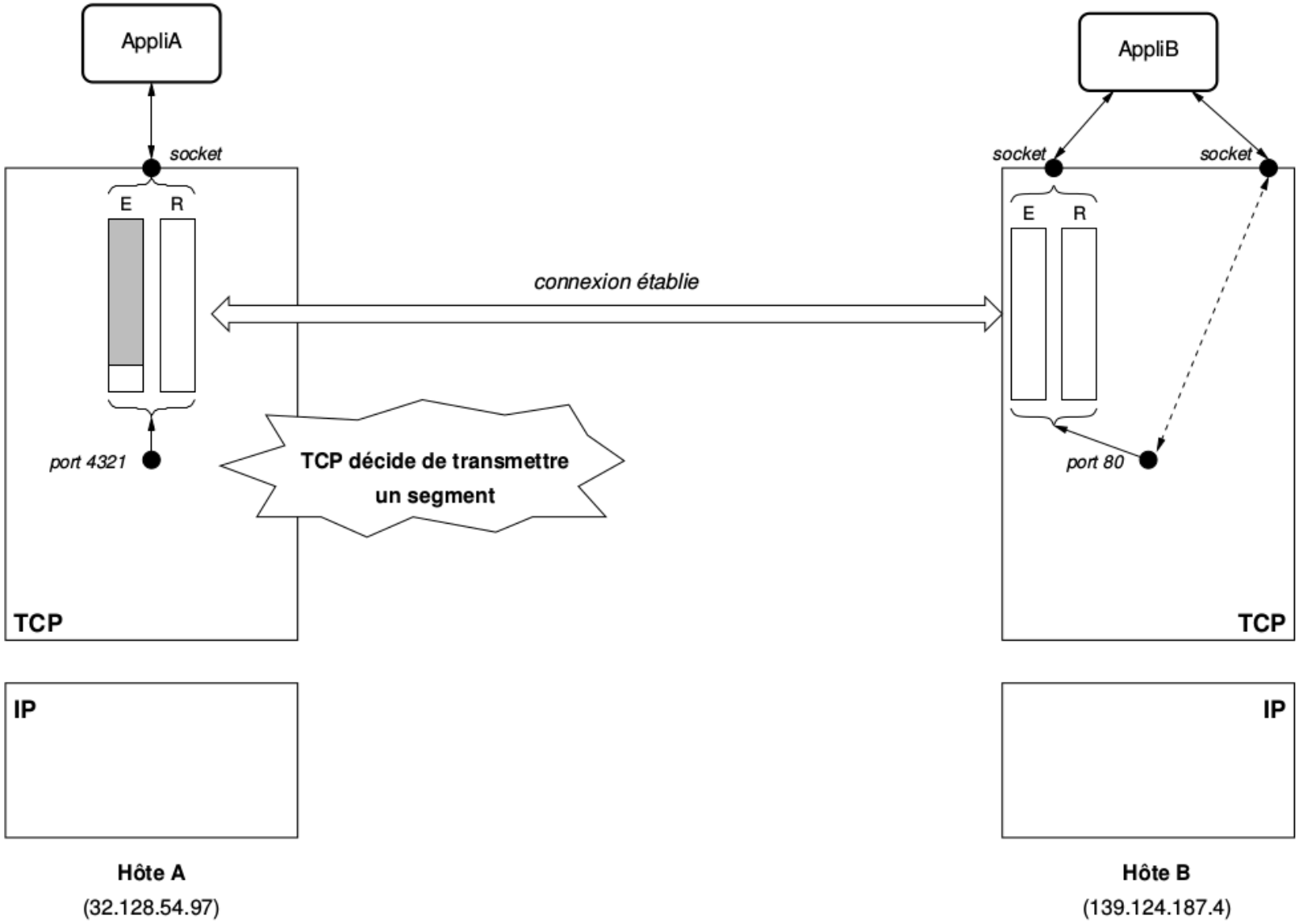


figure 7.9 Création d’un segment de données TCP prêt à envoyer

L’application A ajoute un en-tête TCP au segment comme le montre la figure 7.10

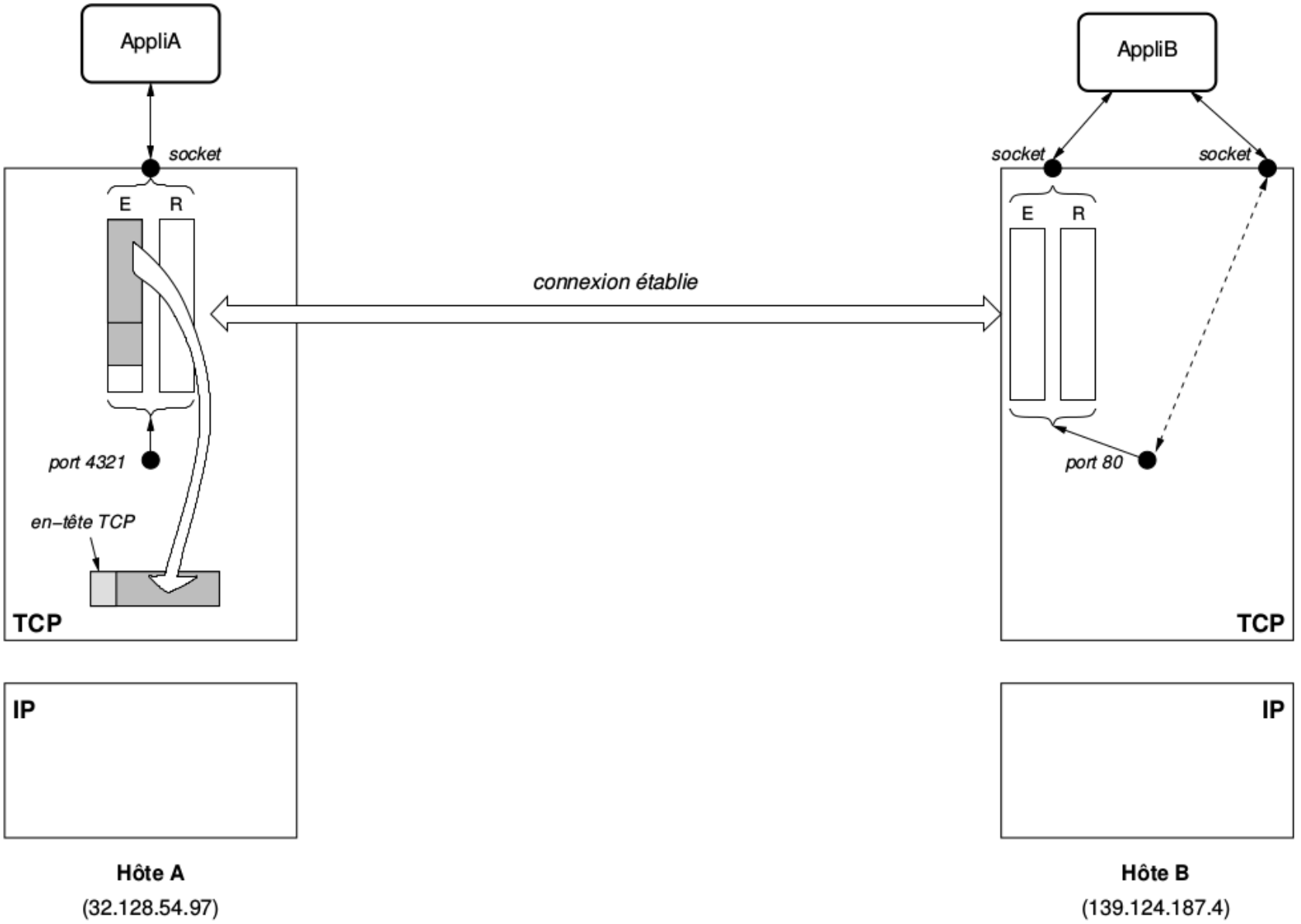


figure 7.10 Ajout d’un en-tête TCP à un segment de données

Ensuite l'application A transfert le segment avec en-tête au protocole IP qui, à son tour ajoute un en-tête IP comme le montre la figure 7.11

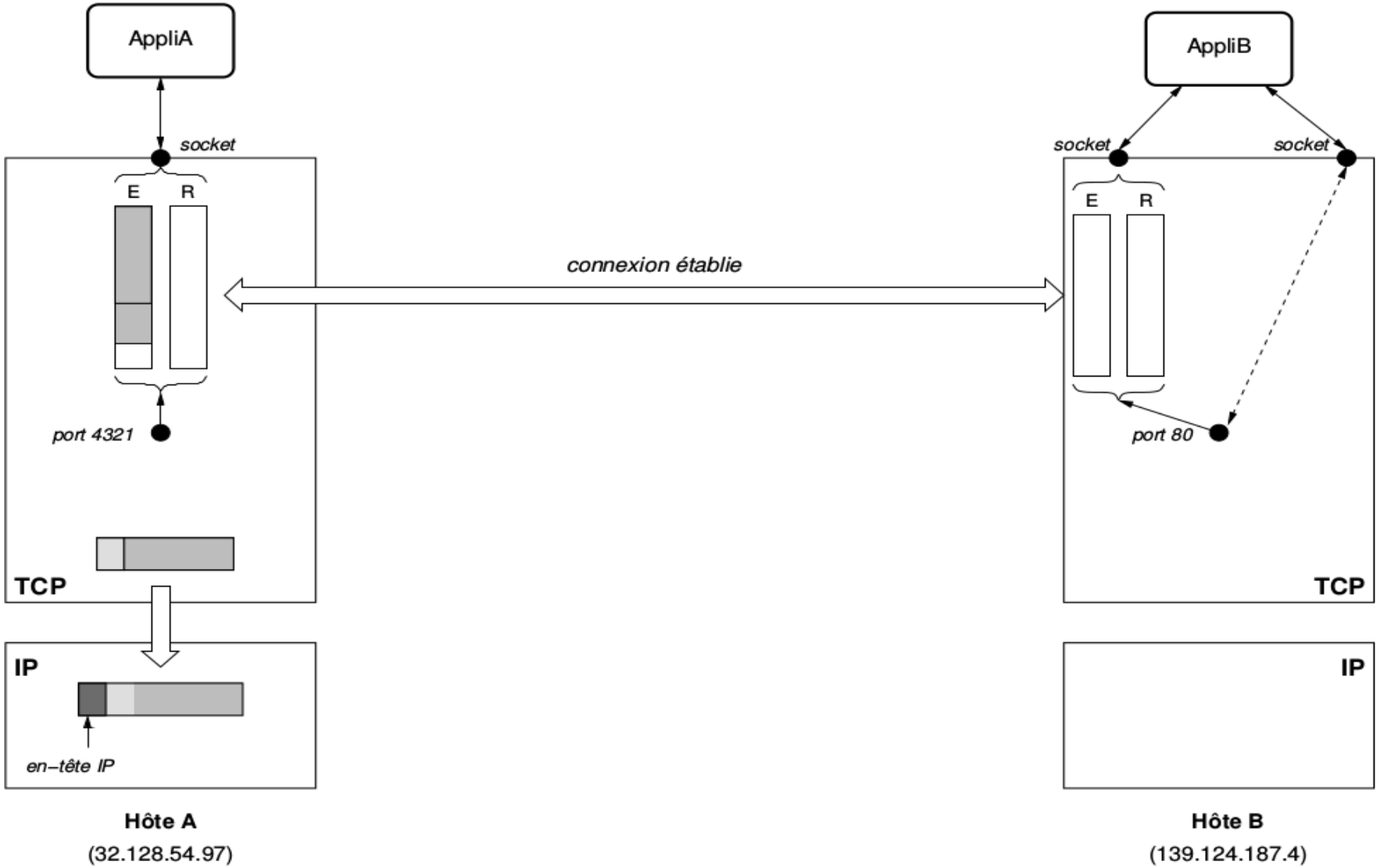


figure 7.11 Transmission d'un segment TCP à la couche IP

Le protocole IP est alors utilisé pour transmettre le datagramme IP de la machine A vers la machine B comme le montre la figure 7.12

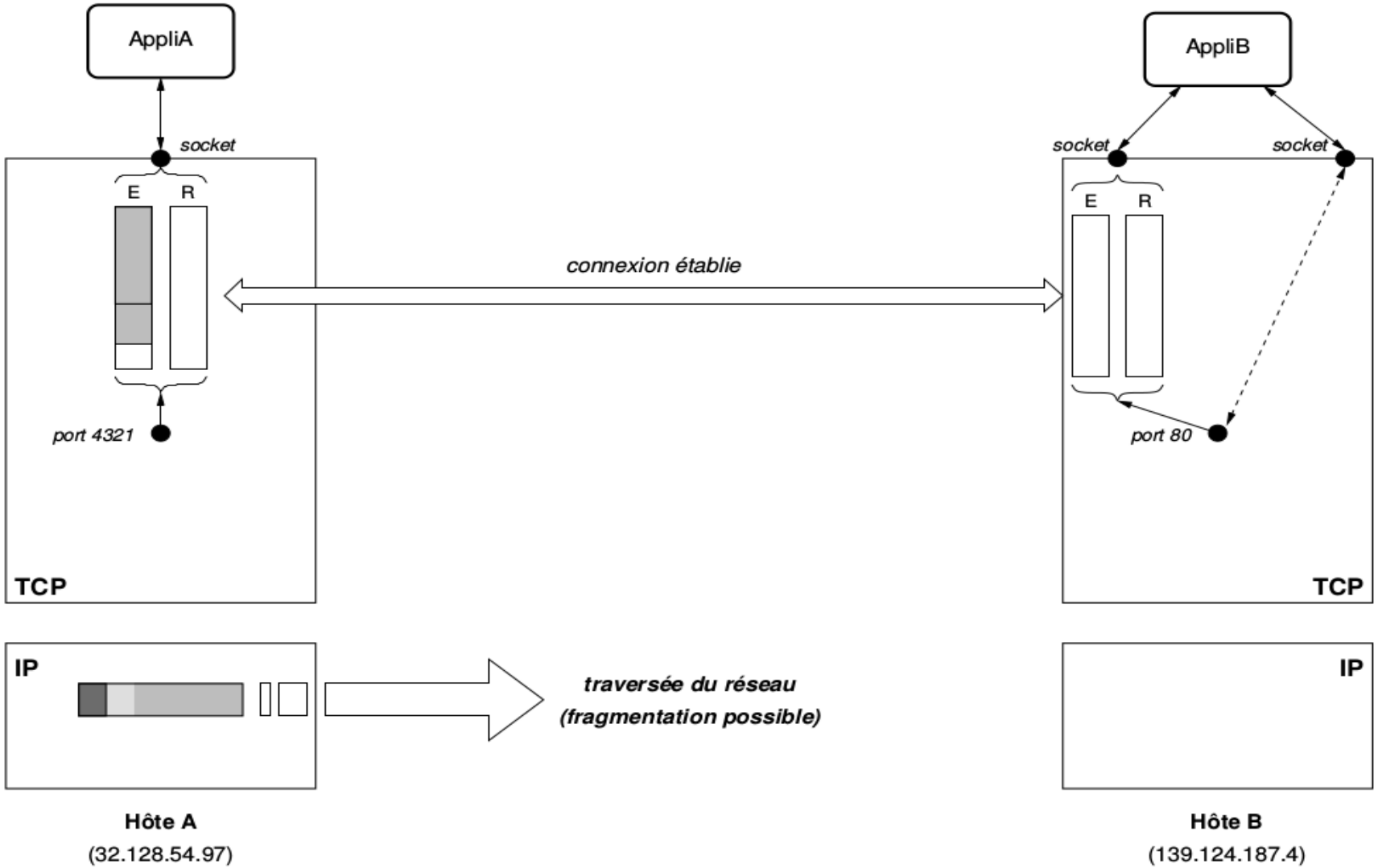


figure 7.12 Transmission d'un datagramme IP

Une fois le datagramme IP reçu par le protocole IP de la machine B, le segment y est extrait et transmis au protocole TCP comme le montrent les figures 7.13 et 7.14

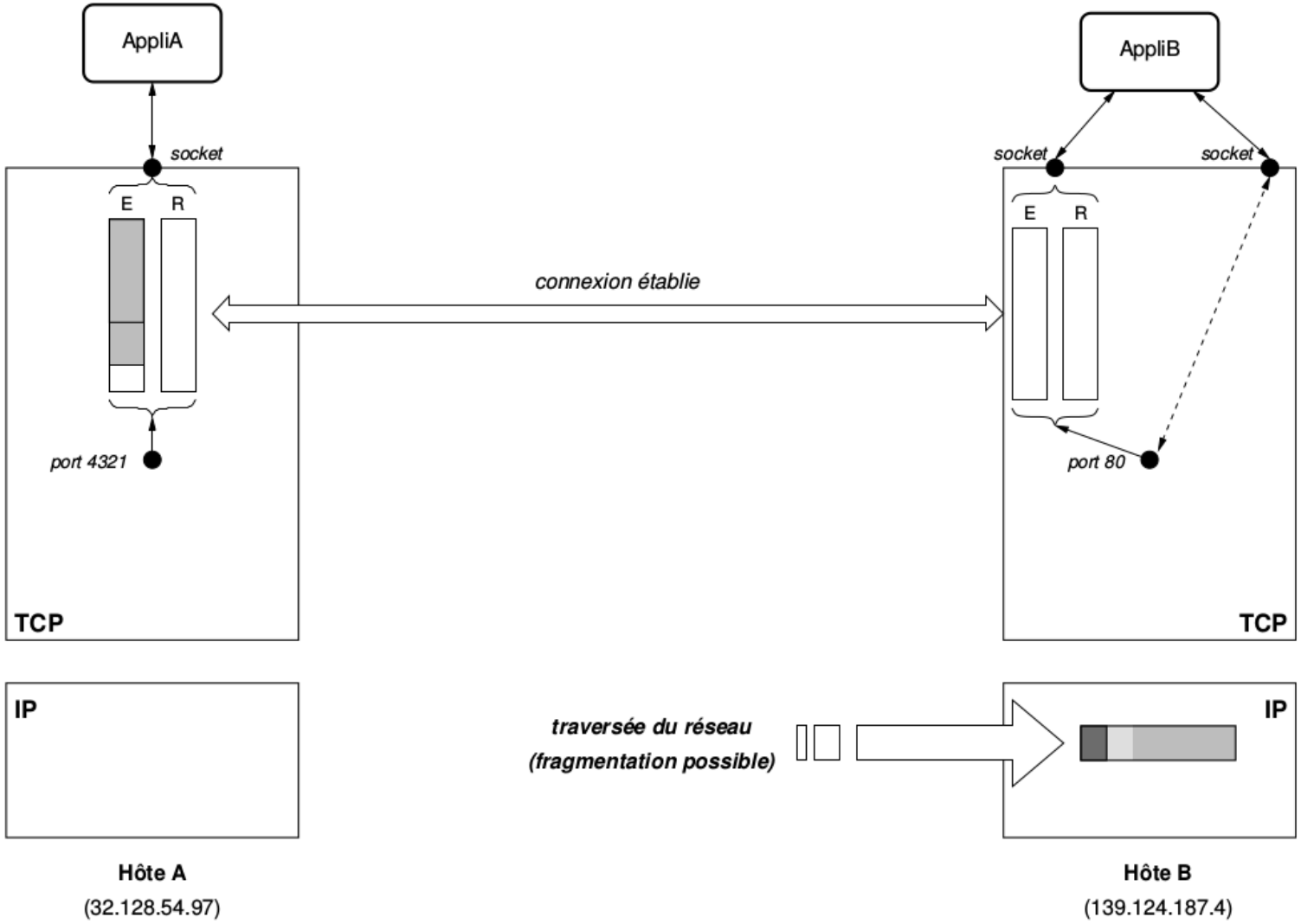


figure 7.13 Réception d’un datagramme IP

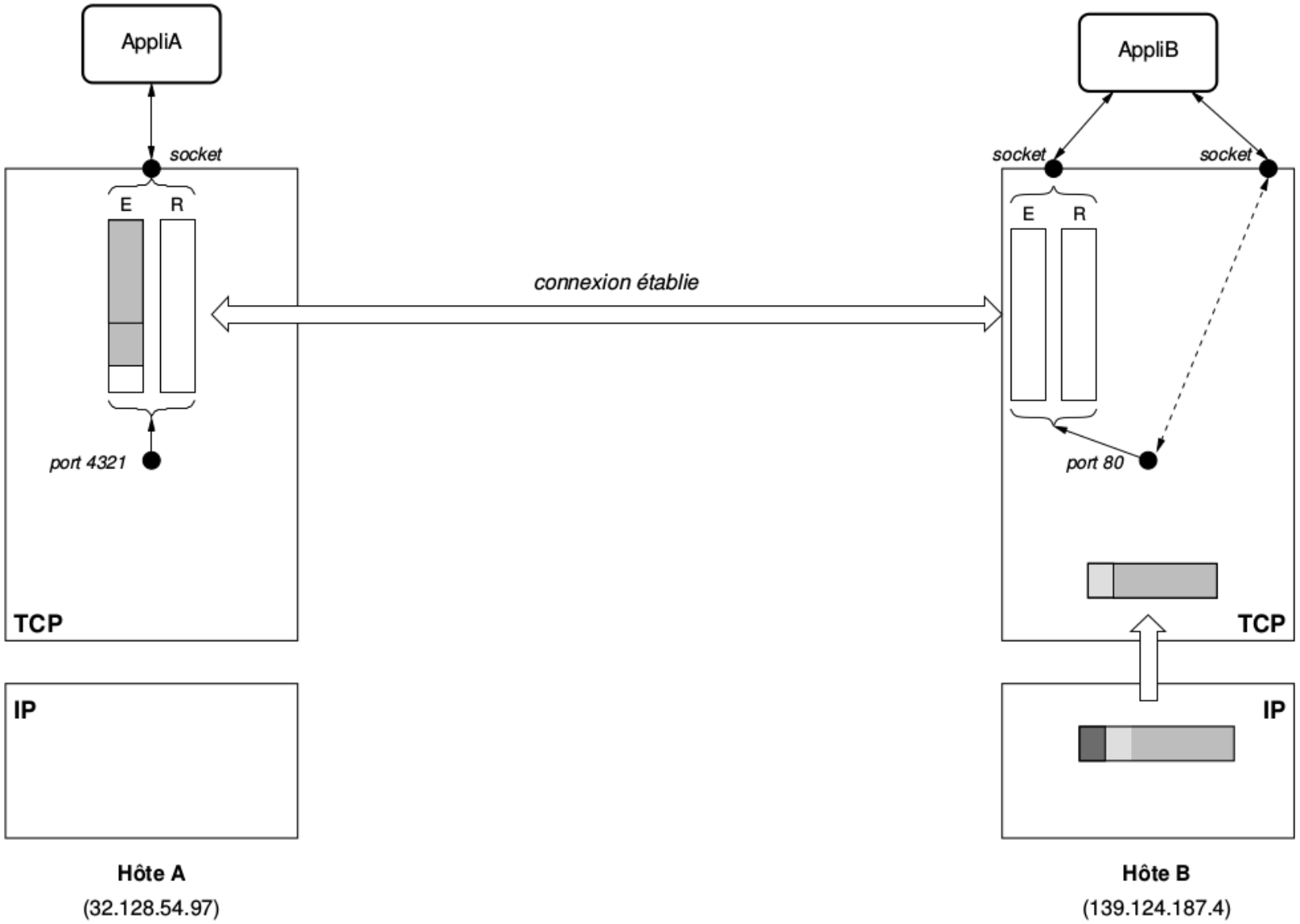


figure 7.14 Extraction et transmission d’un segment TCP

Le protocole TCP enlève l'en-tête du segment et place le segment sans en-tête dans sa file d'attente de réception (R) comme le montre la figure 7.15

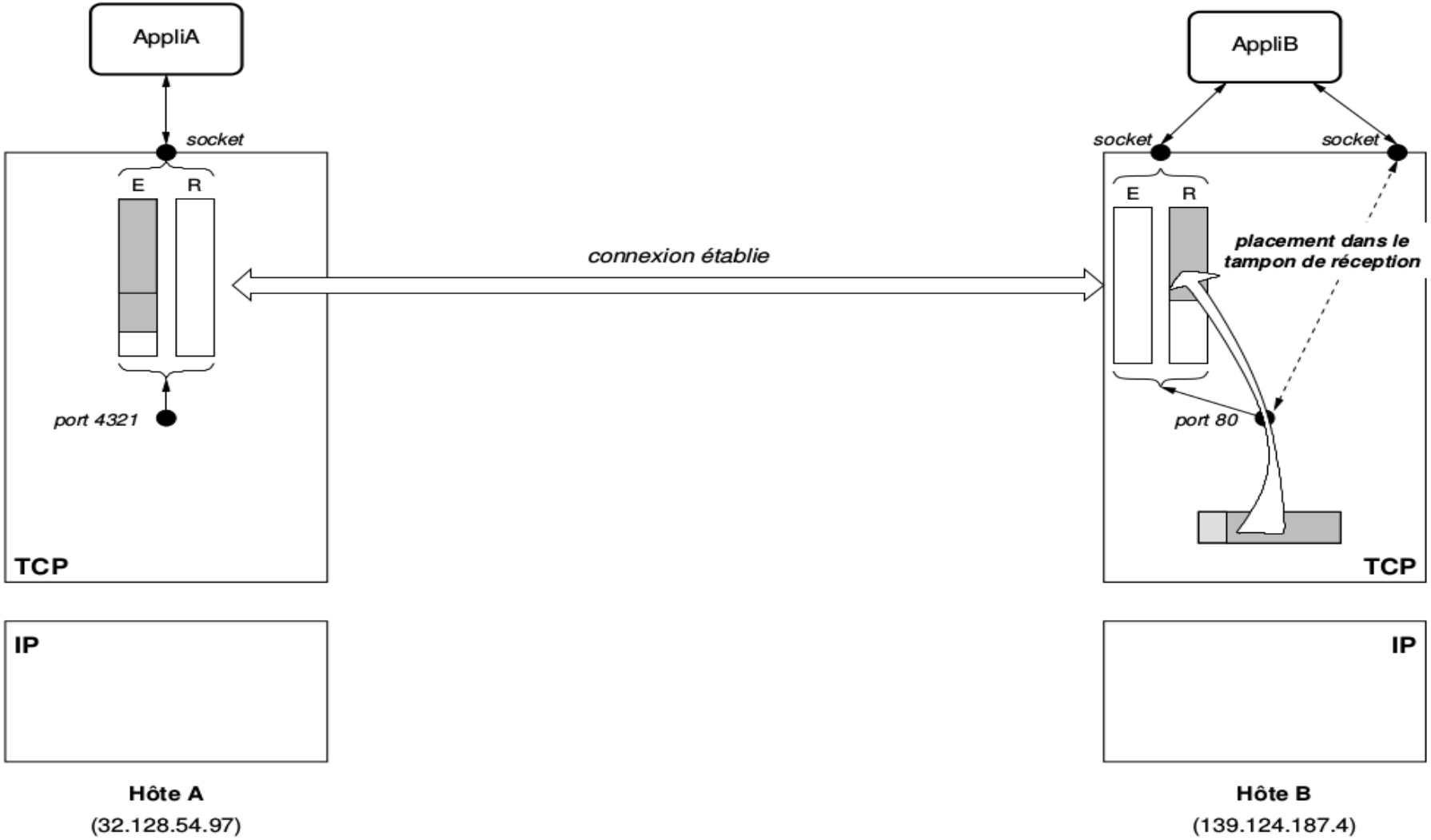


figure 7.15 Remplissage de la file d'attente à la réception

Dans le cas général, le protocole TCP de B envoie un accusé de réception au protocole TCP de A comme le montre la figure 7.16

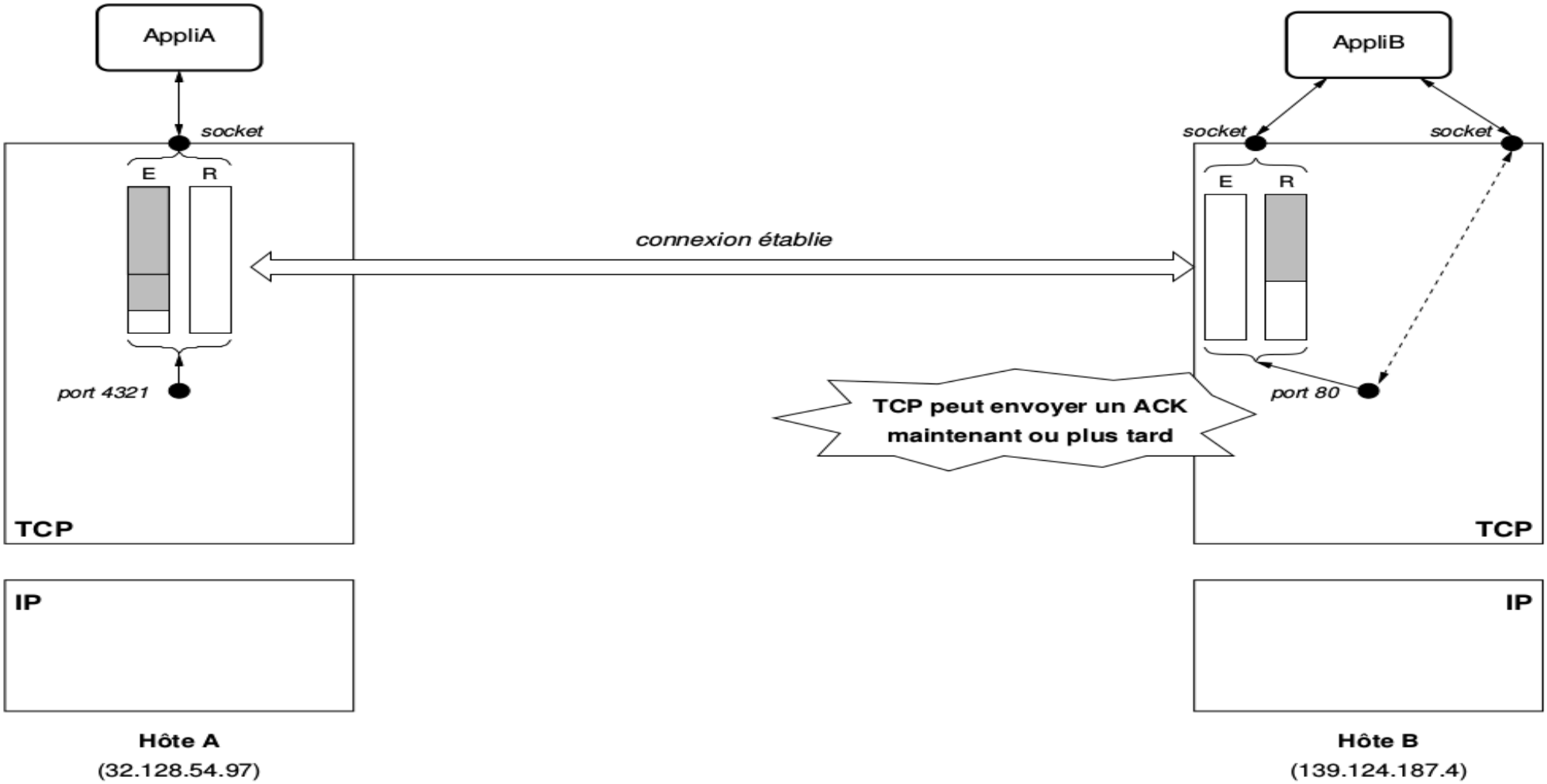


figure 7.16 Envoi d’un accusé de réception après réception des octets d’un segment TCP

L’application B va alors utiliser sa commande recev() pour lire les données transmises comme le montre la figure 7.17

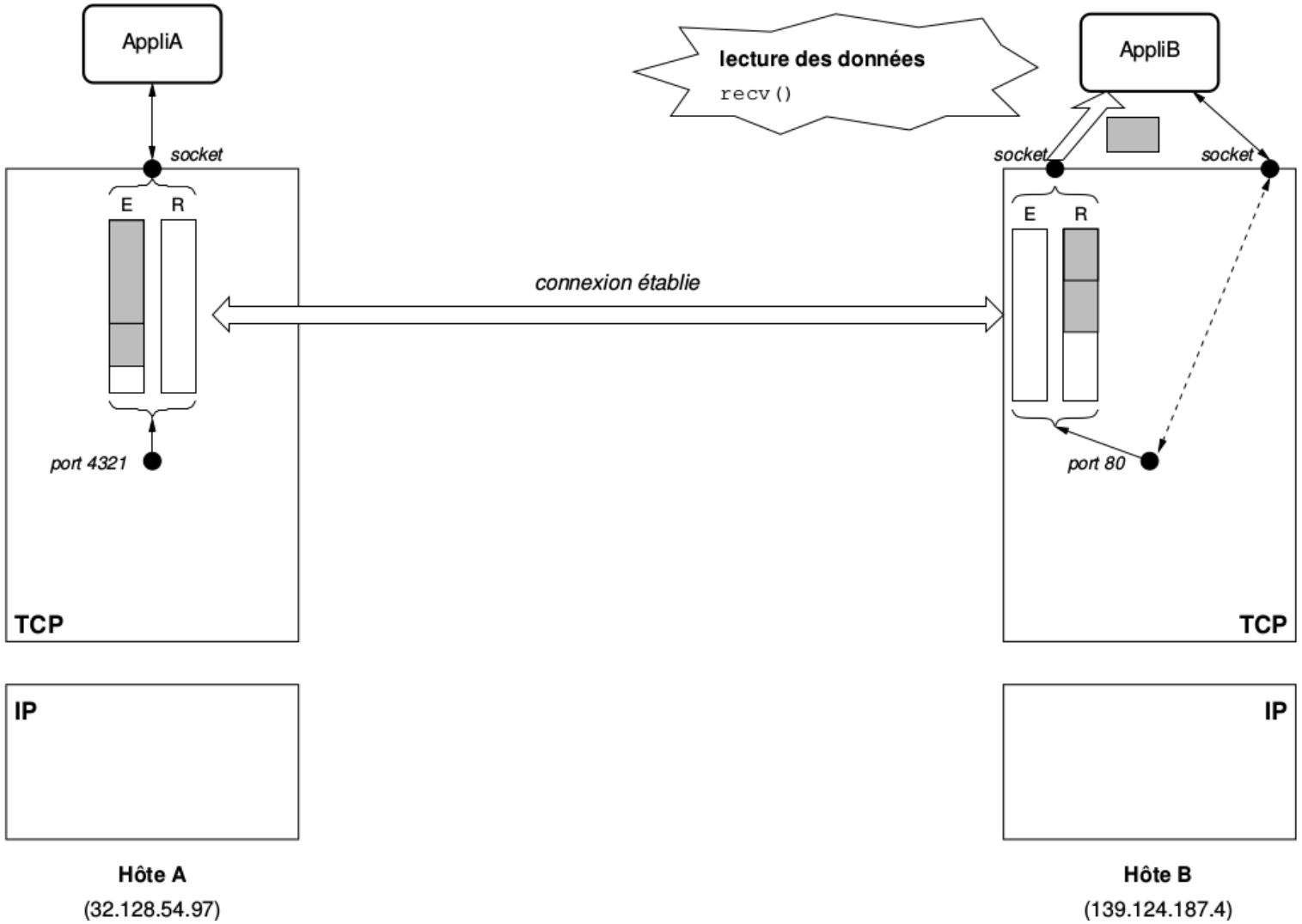


figure 7.17 Lecture des données par l’application B à partir de la file d’attente (R)

Les applications A et B mettent alors à jour leur tampon comme le montre la figure 7.18

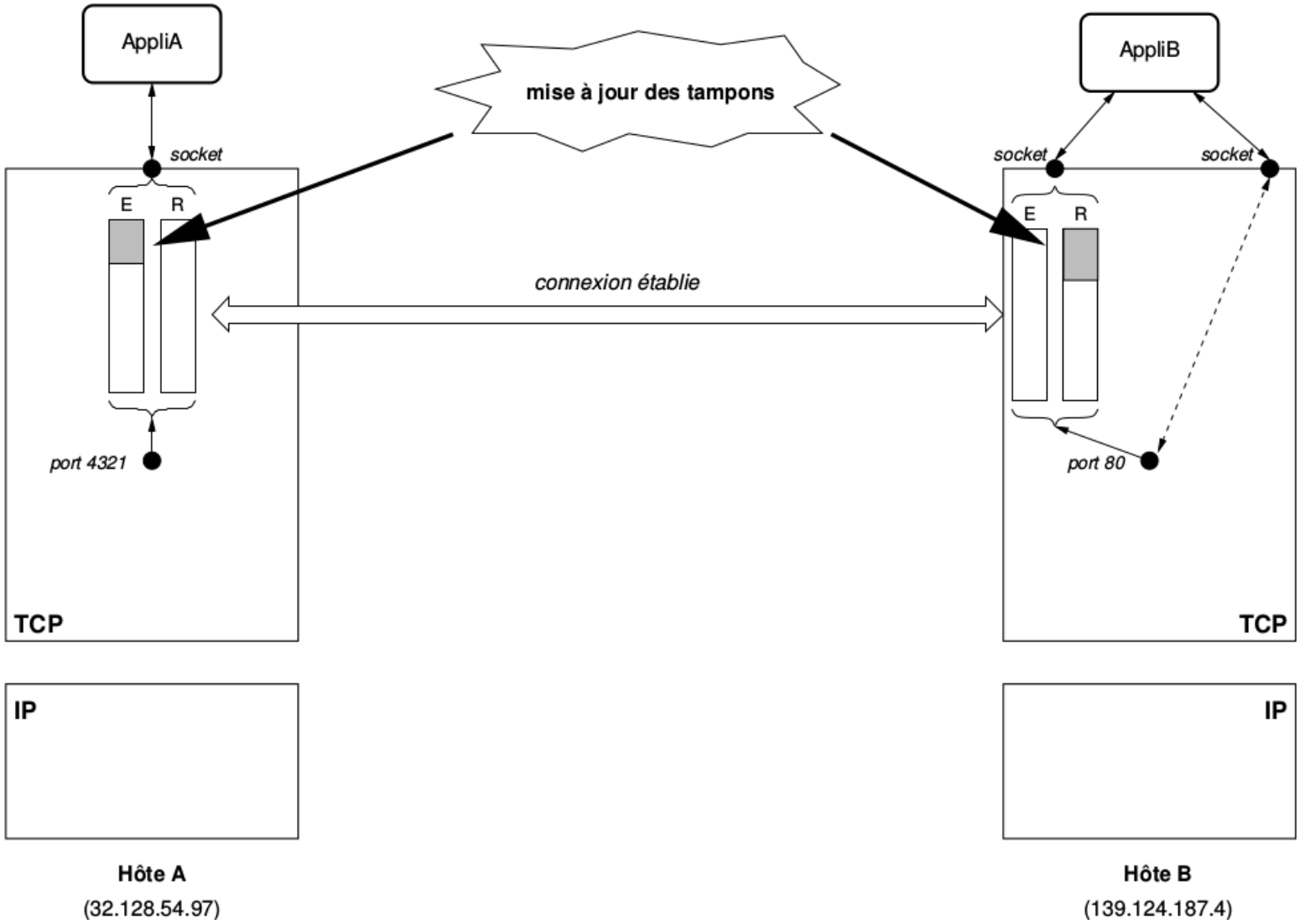
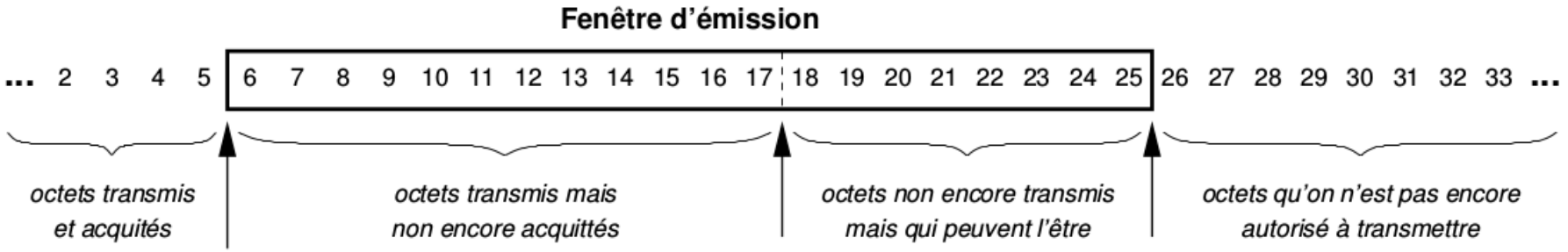


figure 7.18 Mise à jour des tampons

Numéro de séquence et fenêtre glissante

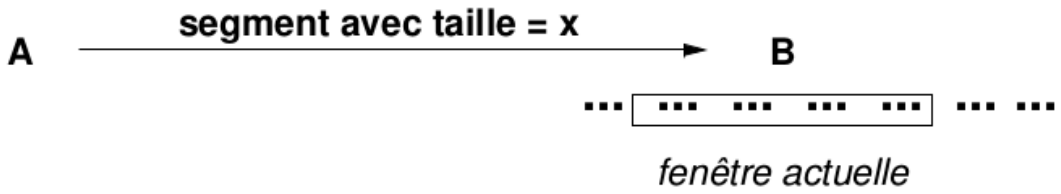
- Pour une transmission efficace, TCP utilise une fenêtre glissante
- TCP n’acquitte pas les segments mais les octets reçus
- Tous les octets de données transmis portent un numéro : le numéro de séquence
- Les acquittements indiquent le numéro du prochain octet attendu
- La fenêtre glissante (émission) comporte alors 3 pointeurs :



- Chaque côté de la connexion possède une fenêtre d’émission et une fenêtre de réception
- Les acquittements peuvent transiter avec les données (superposition)

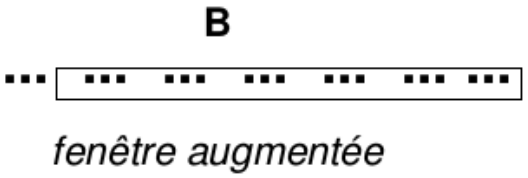
Taille de fenêtre variable et contrôle de flux

- La fenêtre glissante n'a pas une taille fixe
- Les segments (en particulier ACK) contiennent une information qui renseigne sur la taille de la fenêtre :

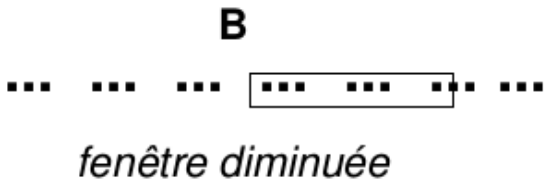


A indique à B la place disponible dans son tampon de réception

- La réaction de B dépend de la taille annoncée :
  - augmentation : B augmente sa fenêtre et envoie les octets supplémentaires qu'elle comprend



- diminution : lors du glissement, B diminue sa fenêtre (sans exclure les octets qui y étaient déjà)



7.4 Format des segments TCP

TCP utilise une unité de données de protocole appelé (PDU) pour échanger, pour établir et libérer une connexion ou pour transférer et acquitter les données comme le montre la figure7.

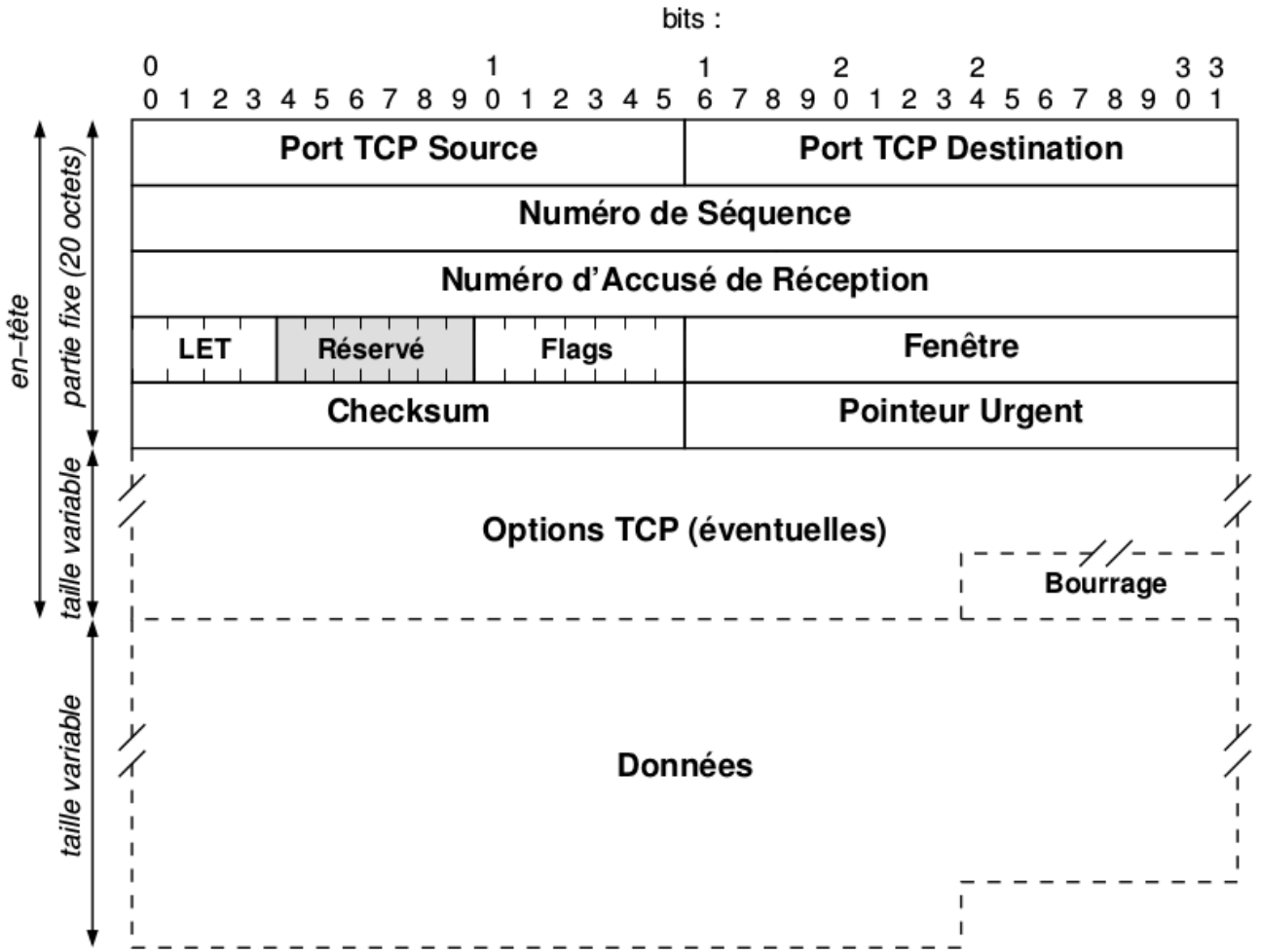
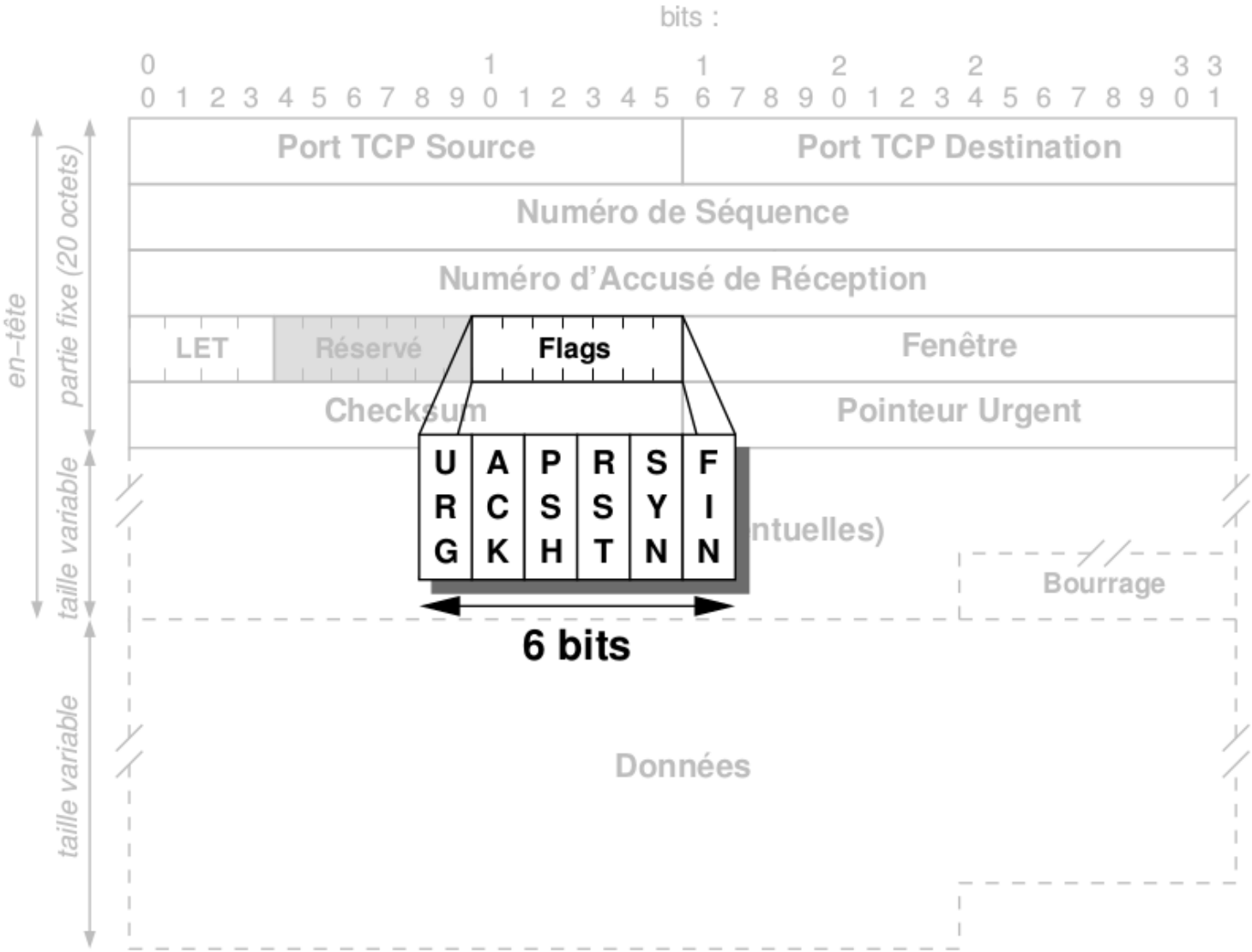


Figure 7.19 Format d'un segment TCP

Segment TCP : le champ flags



Segment TCP : champ Numéro de séquence

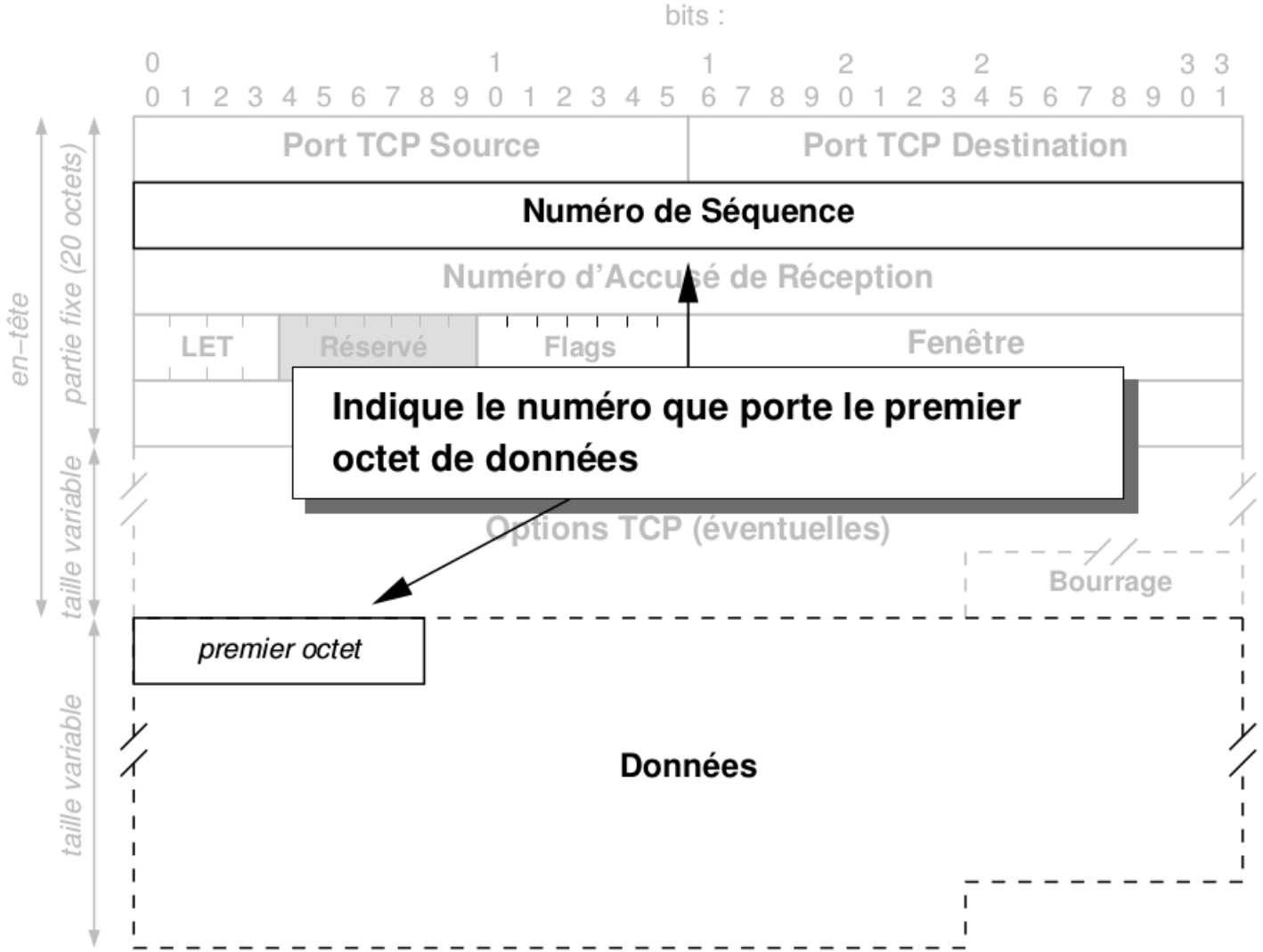


figure 7.20 Champ numéro de séquence d'un segment TCP

Segment TCP : champ LET

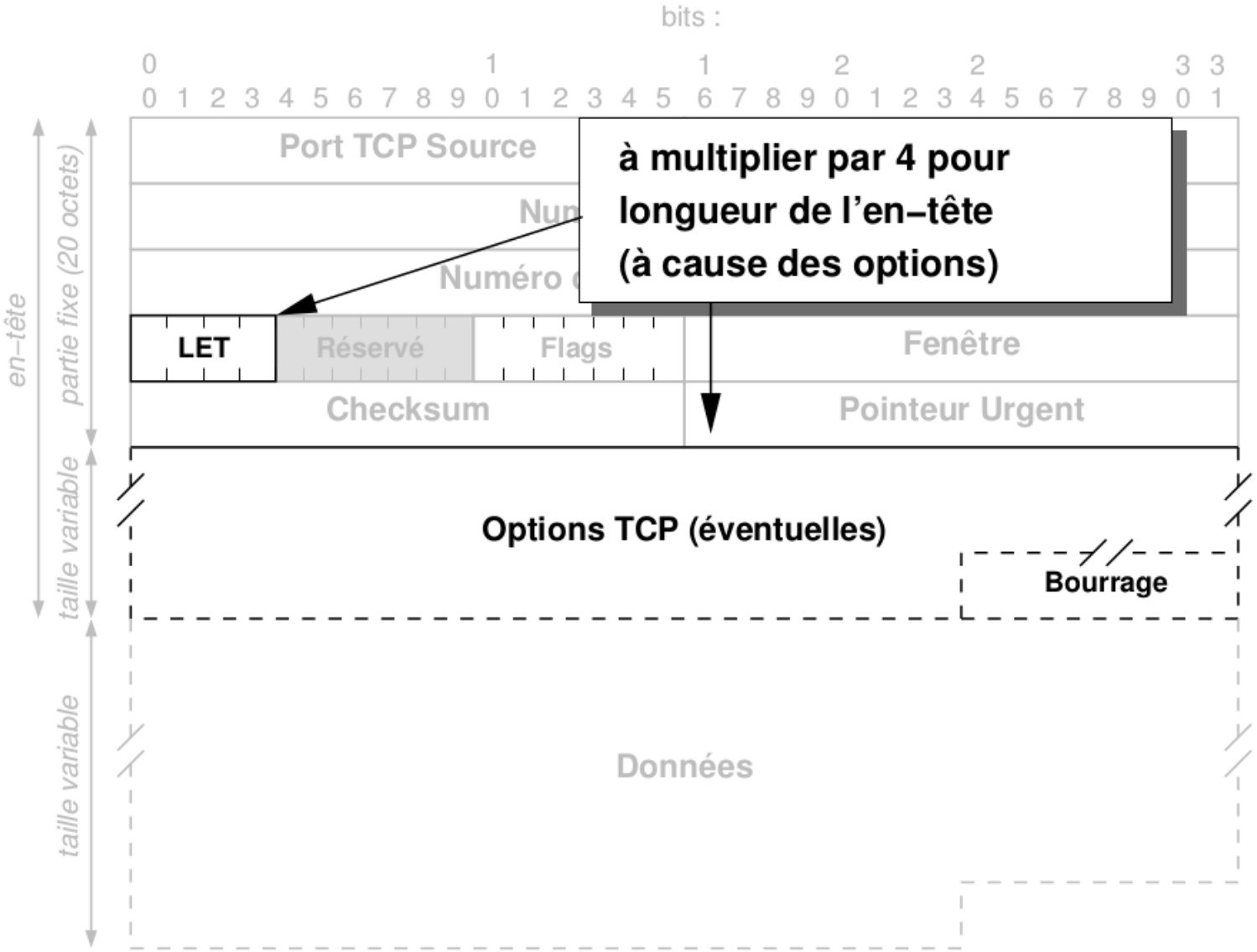


figure 7.21 Champ LET permettant de déterminer la longueur d'un en-tête TCP

Segment TCP : champ Fenêtre

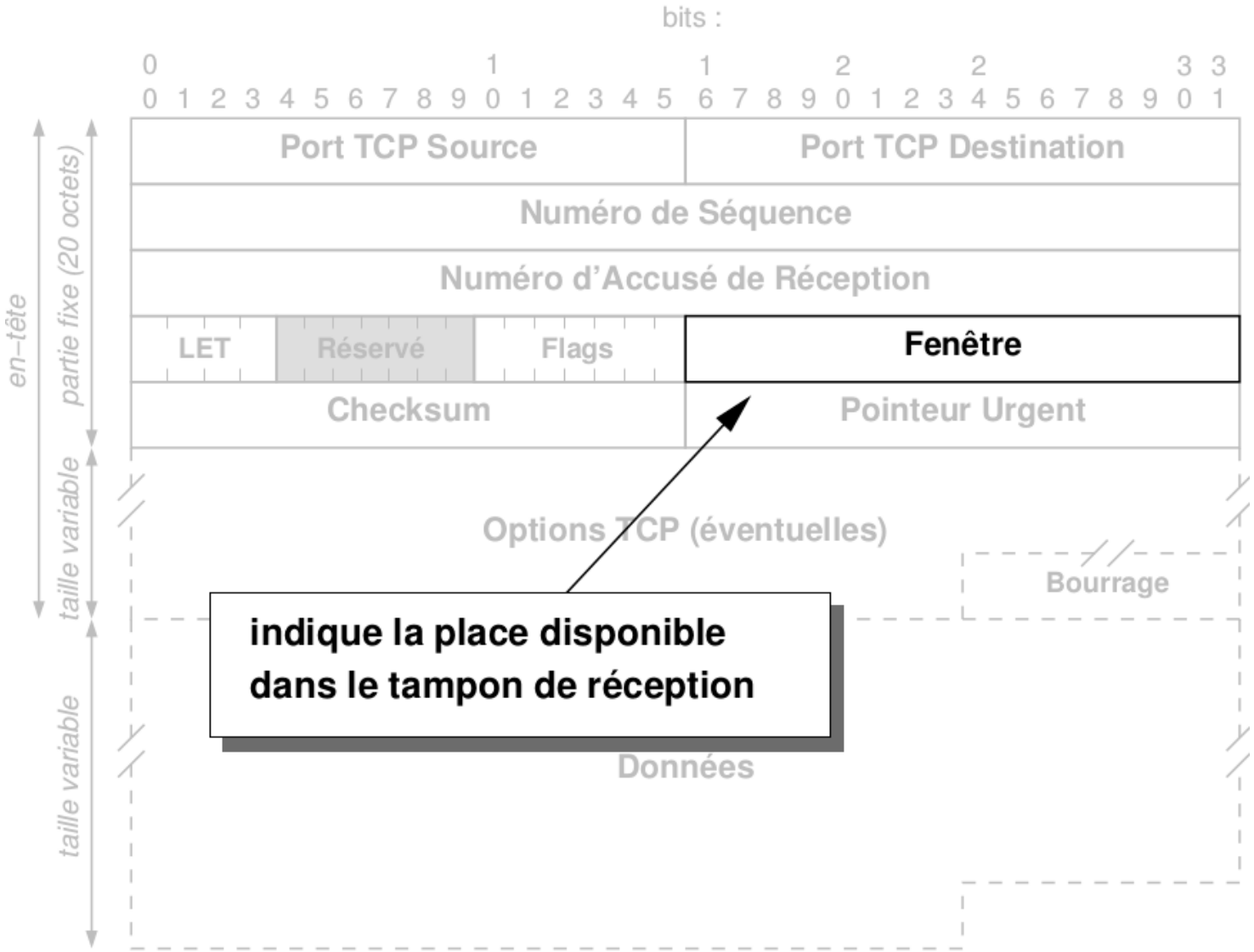


figure 7.22 Champ indiquant la place disponible dans le tampon de réception

Segment TCP : Checksum

Ce champ est obligatoire et permet de vérifie la totalité de l'intégrité du segment + son Pseudo en-tête TCP.
Comme pour UDP, ce champ permet de s'assurer :

- que les données sont correctes
- que les ports sont corrects
- que les adresses IP sont correctes

On utilise le même procédé de calcul que IP/UDP (bourrage éventuel 1 octet à 0) + pseudo en-tête TCP

Le Pseudo en-tête TCP est de taille 12 octets et l'interaction avec IP

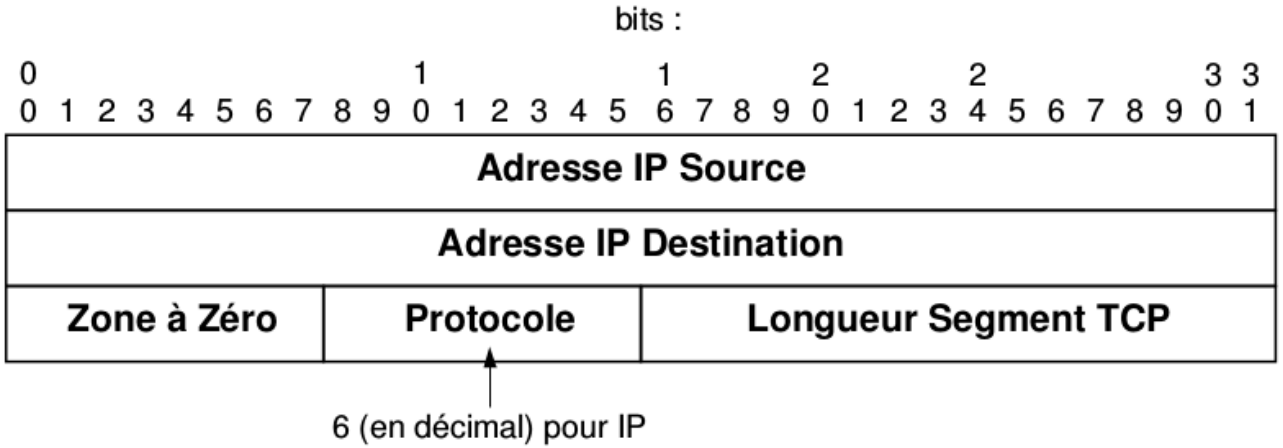


figure 7.23 Pseudo en-tête TCP

Le champ longueur segment TCP utilisé conjointement avec le champ LET permet de déterminer la taille des données utiles dans le segment.

Segment TCP : option MSS

Le champ option MSS n'est utilise que pendant la phase de connexion et sa taille est de 4 octets comme le montre la figure 7.24

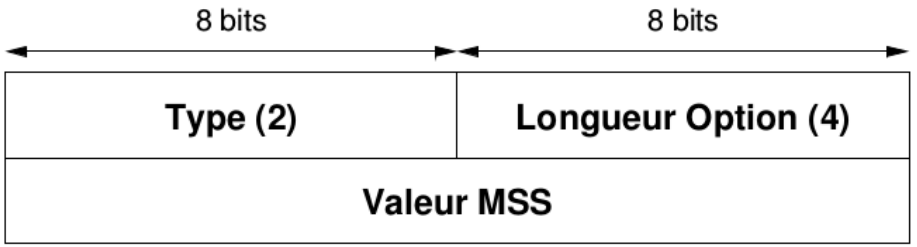


figure 7.24 Champ MSS

- Chaque côté de connexion indique la taille maximale des données des segments appelé MSS (Maximum Segment Size) qu'il veut recevoir :
  - généralement MSS est fixé à MTU - 40 (en-têtes IP et TCP sans option)
  - MSS dépend aussi de la taille des buffers de réception
  - par défaut sa taille est de 536 octets
- MSS est difficile à choisir sur l'Internet :
  - si elle est trop petite alors on peut avoir une perte d'efficacité
  - si elle est trop grande alors il y a risque de fragmentation des datagramme IP contenant des segments

- La taille idéal de MSS doit prendre la plus grande valeur tel qu’aucun datagramme ne soit fragmenté

Données Urgentes (Hors Bande)

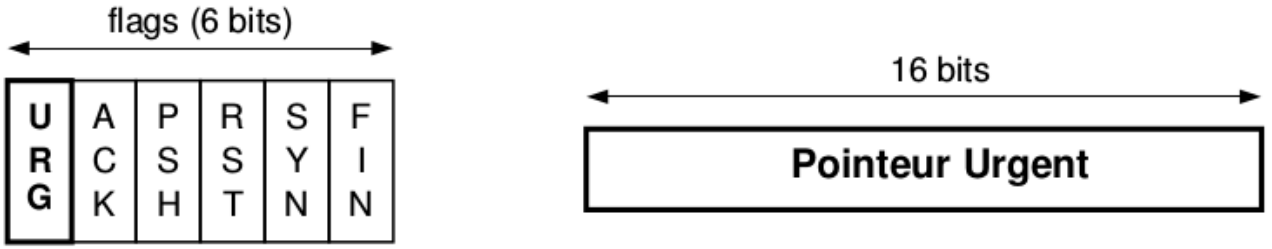
Il arrive q’un émetteur veuille envoyer des données en urgence, sans attendre que le récepteur ait lu les données précédentes comme c’est le cas de la commande Ctrl-C pour arrêter le traitement de l’application destinataire

Le TCP émetteur place les données urgentes et envoie immédiatement le segment

Le TCP récepteur interrompt l’application destinataire (sous Unix, signal SIGURG)

Les données urgentes sont mises en œuvre par le bit URG et Pointeur Urgent comme suit :

- si bit URG = 1 : alors les données urgentes sont présentes dans le segment et le Pointeur Urgent indique leur fin dans le segment
- si bit URG = 0 : alors il n’y a pas de données urgentes et le Pointeur Urgent est ignoré.



Remise forcée

Il arrive qu'une application émettrice demande de ne pas retarder l'émission des données, ce cas est utile dans le cas d'un terminal virtuel.

TCP émetteur place le bit PSH du champ flags à 1 comme le montre la figure 7.25 :

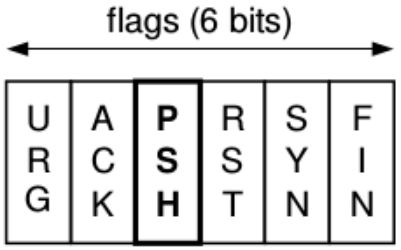
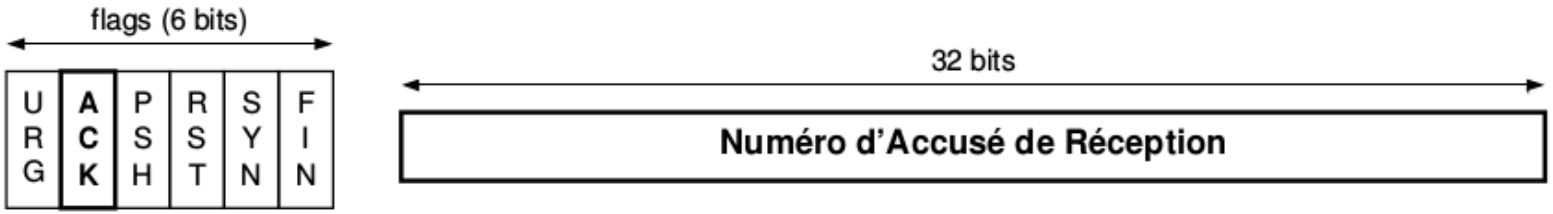


figure 7.25 Champ PSH

Dans ce cas le TCP récepteur doit remettre les données au plus vite plutôt de les tamponner dans sa file d'attente de réception.

Acquittements et Retransmissions

Lorsque le bit ACK à 1 alors le champ Numéro d'Accusé de Réception doit être utilisé



- Un segment non acquitté est retransmis après timeout
- Un segment retransmis peut contenir plus de données que le précédent
- Le champ Numéro ACK n'acquitte pas le segment mais indique le numéro du prochain octet attendu
- Le fait que l'ACK est cumulatif a des avantages et des inconvénients car il peut entraîner un rejet global ou selectif
- En pratique l'émetteur ne renvoie que le premier segment non acquitté

Établissement d’une connexion

L’établissement d’une connexion se fait en 3 temps, avec bits SYN et ACK comme le montre la figure 7.26

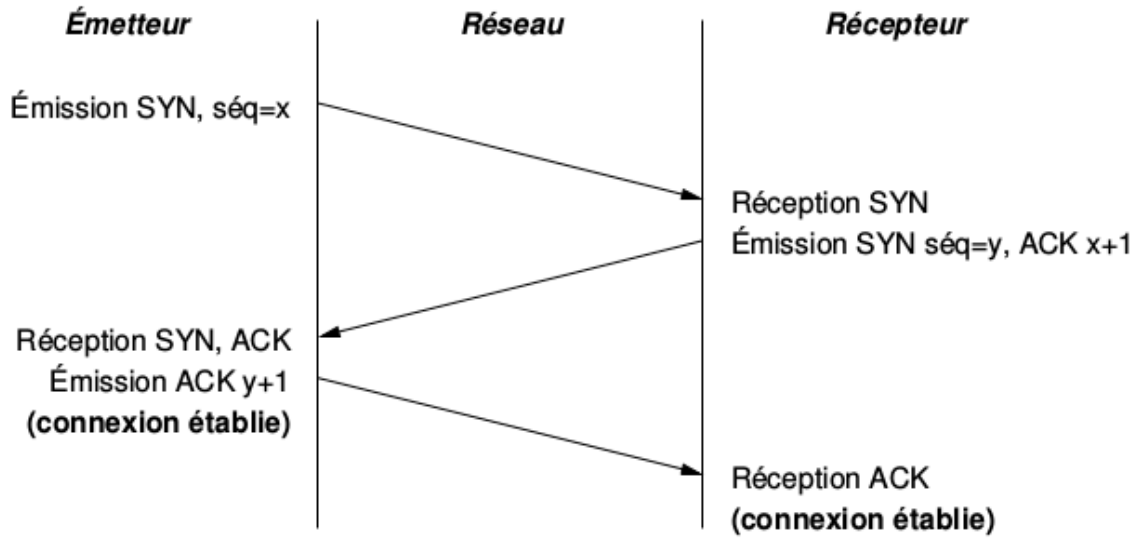


figure 7.26 Établissement d’une connexion

Libération d’une connexion

Le processus de libération se fait en trois temps, avec les bits FIN et ACK.

Une connexion est libérée lorsque chaque côté indique qu’il n’avait plus de données à émettre comme le montre la figure 7.27 :

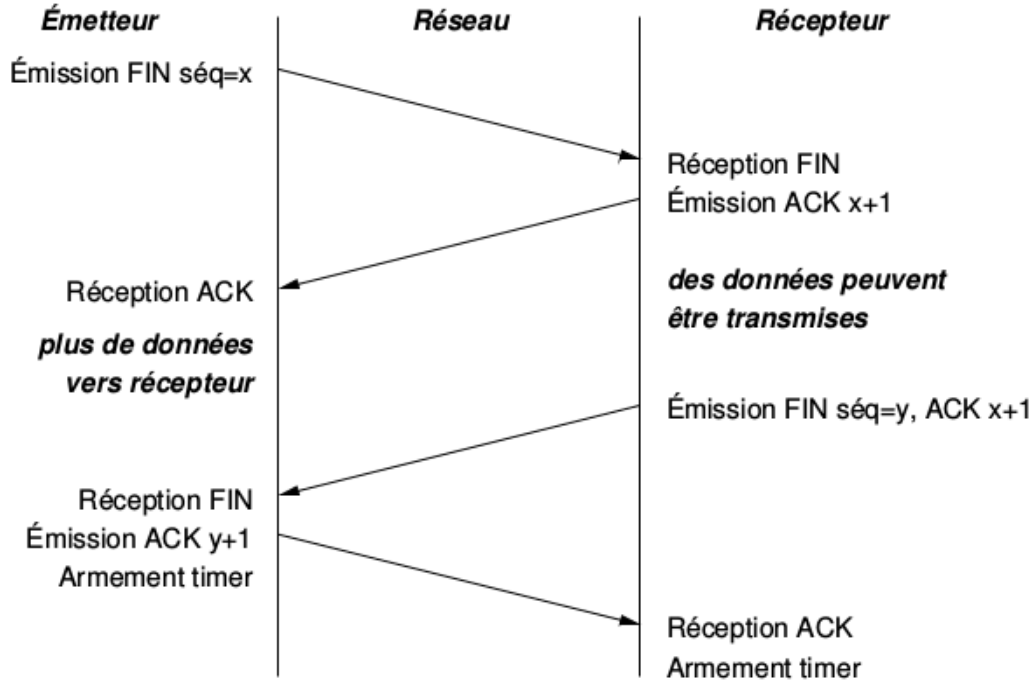


figure 7.27 Libération d’une connexion

- À la fin, un temporisateur est utilisé pour laisser le temps aux segments retardés d’arriver ou d’être détruits
- Puis les données relatives à la connexion sont détruites

Fermeture brutale d’une connexion

Si le bit RST est à 1 alors il y a eu un problème grave qui nécessite une libération immédiate de la connexion dans ce cas les données non traitées ou les segments retardés sont perdus.

Ce bit est aussi utilisé pour refuser une demande de connexion comme le montre la figure 2.28

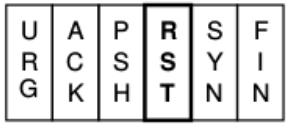


figure 7.28 Positionnement du bit RST

7.5 Translation d’adresses

Problématique

Le mécanisme de translation d'adresses (en anglais Network Address Translation noté NAT) a été mis au point afin de répondre à la pénurie d'adresses IP avec le protocole IPv4.

- Principe :

Cela consiste donc à utiliser une passerelle de connexion à internet, possédant au moins une interface réseau connectée sur le réseau privé et au moins une interface réseau connectée à Internet (possédant une adresse IP routable).

C’est le cas des routeurs ADSL utilisés au Sénégal qui ont une adresse IP publique et une interface connectée au réseau privé des abonnés de la SONATEL (Société Nationale des Télécommunications).

Schéma de translation et principe

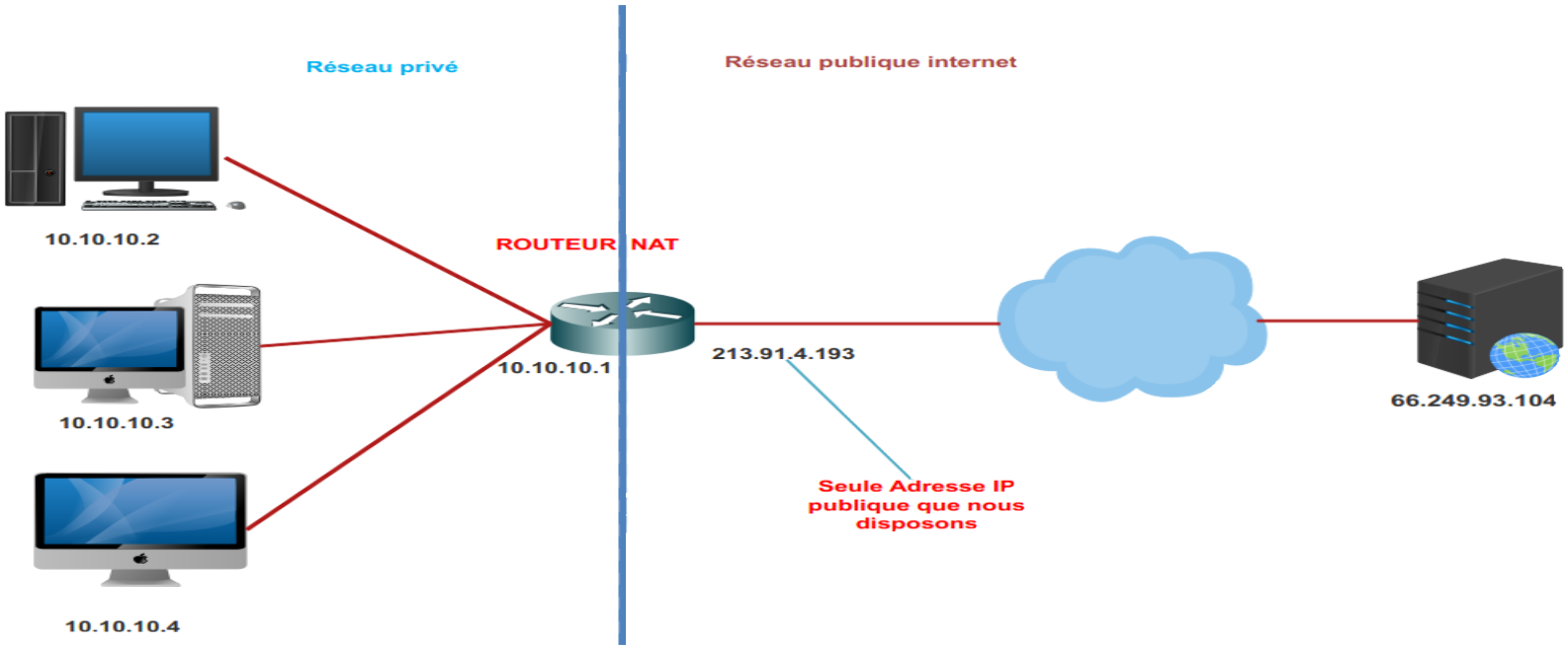


Figure 7.29 Principe de translation d'adresses

Lorsqu' une machine dans le réseau privé veut envoyer des paquets sur internet, l'adresse IP source du paquet sortant sur l'interface publique du routeur qui fait le NAT est remplacée par l'adresse IP publique du routeur.

En revanche, les paquets reçus d'internet par le routeur résultant des réquêtes émises du réseaux privé voient leur adresse de destination remplacée par une adresse du réseau privé avant que ceux ci ne soient retransmis à leur destinataire dans le réseau privé.

Les paquets sortants

La figure 7.30 illustre le principe de translation d'adresses de manière générale

Un paquet émis par la machine 10.10.10.2 à destination du site web à l'adresse 66.249.93.104 voit son adresse source remplacé par l'adresse IP publique du routeur qui est 213.91.4.193.

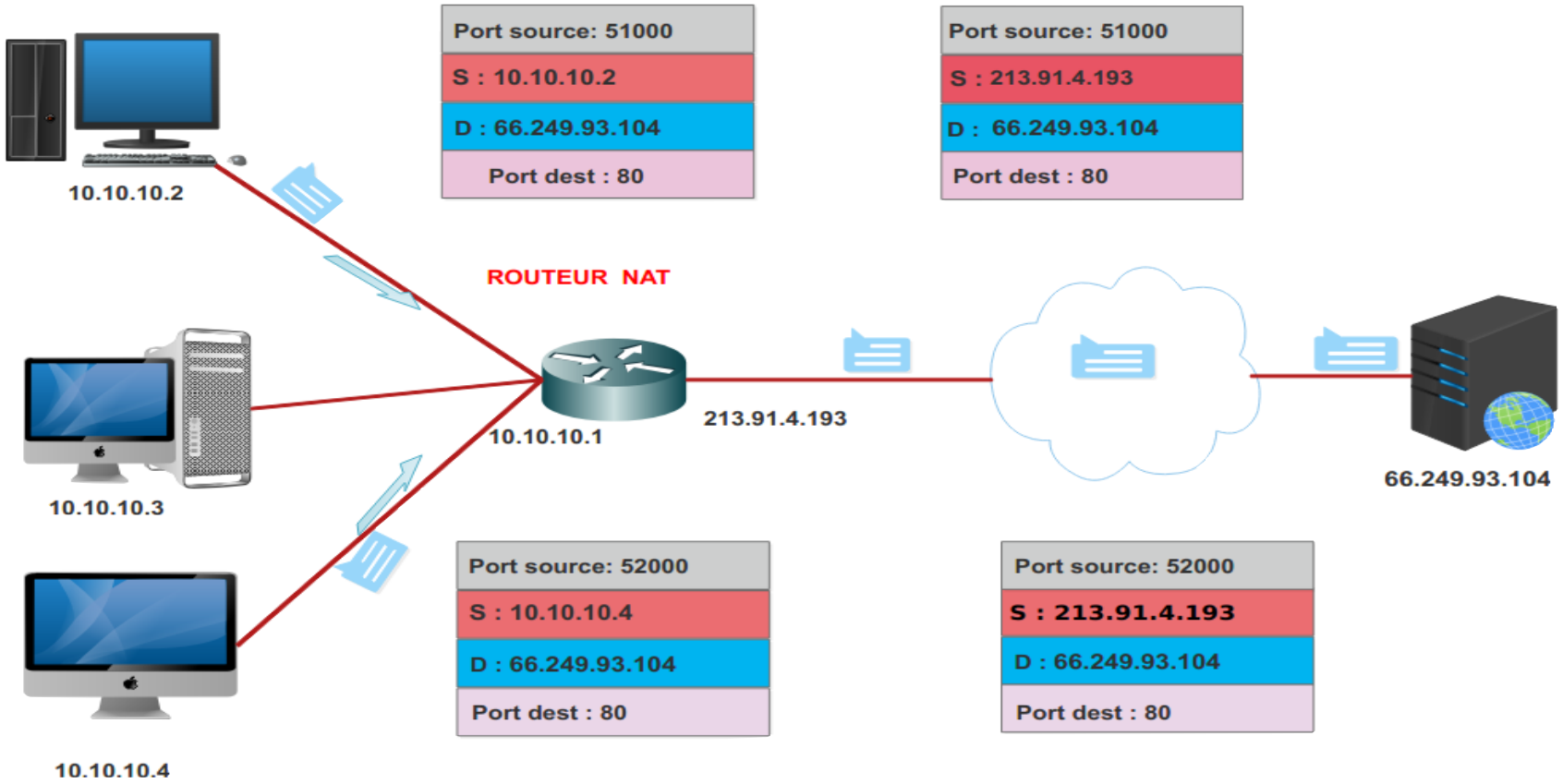


Figure 7.30 Remplacement de transport et translation d'adresses

Il faut noter une connexion vers une application nécessite le quadruplet (**adresse IP source, port source et adresse IP destination, port destination**) comme vu dans les paragraphes sur TCP et UDP.

Les paquets rentrants

Pour retrouver les adresses IP des paquets IP qui reviennent au client.

Le routeur NAT utilise une table de correspondance appelé table de NAT qui est construite de manière dynamique comme le montre le tableau 7.3

Tableau 7.3 : Table de translation d'adresses

Interne		**Externe**	
@IP	**Ports**	**@IP**	**Ports**
10.0.0.2	51000	213.91.4.193	51000
10.0.0.4	51000	213.91.4.193	52000

Par exemple si le routeur NAT reçoit des demandes de connexion des machines 10.0.0.2 et 10.0.0.4 avec des ports sources confondus 51000 alors pour éviter des confusions le routeur décide d'associer le port 52000 comme port source de requête de la machine 10.0.0.4.

Ainsi si un paquet venant d'internet allant vers le port 51000 du routeur NAT sera redirigé vers la machine 10.0.0.2 port 51000 alors un paquet venant d'internet et allant vers le port 52000 du routeur NAT sera redirigé sur le port 51000 de la machine 10.0.0.4.

En réalité nous n'avons traiter dans ce paragraphe que le mécanisme de translation de port appelé PAT utilisé par des routeurs ADSL ne possédant qu'une seule adresse IP publique à un instant donné.

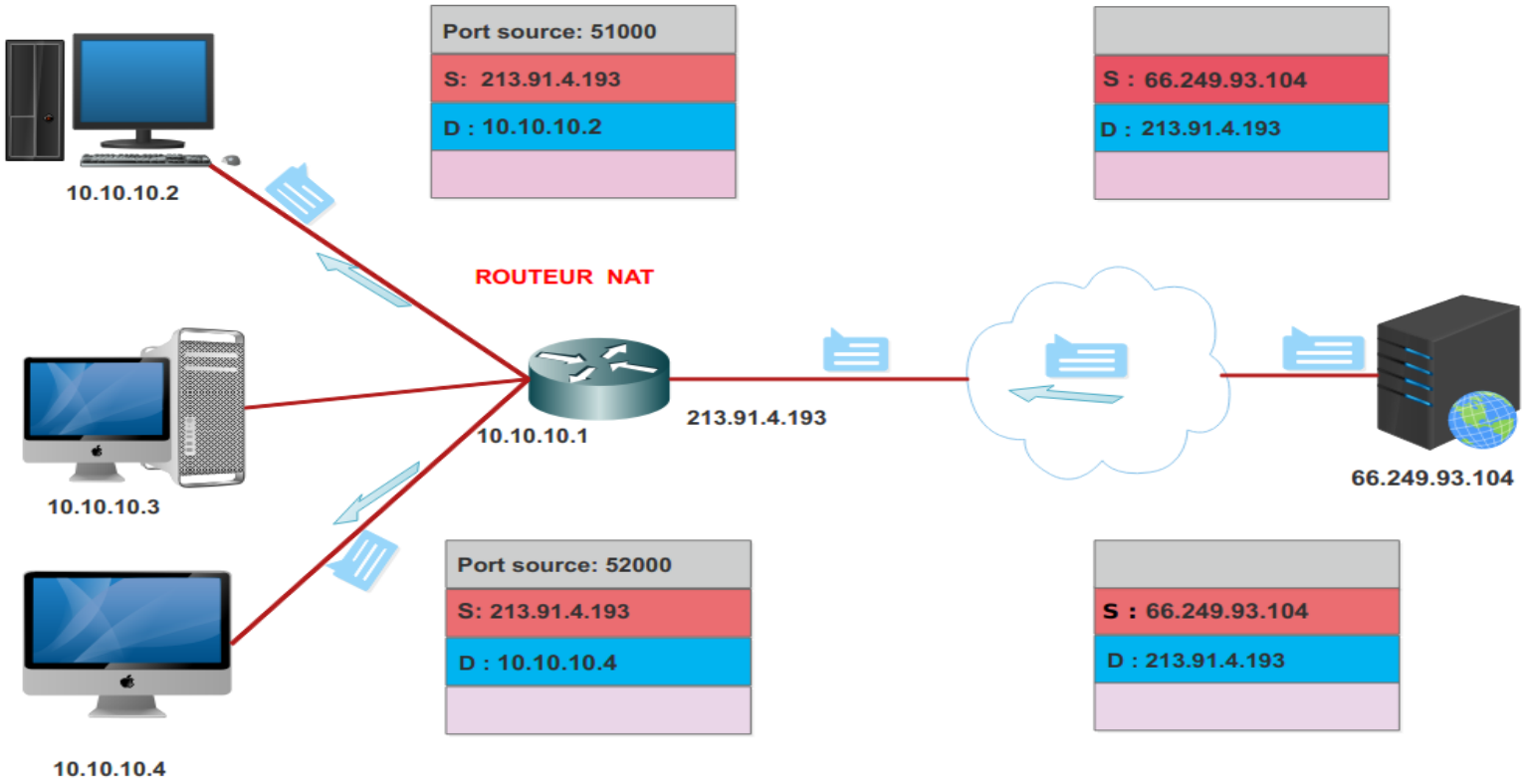


Figure 7.31 Transmission des paquets venant d'internet à destination du réseau privé

En fait il existe trois types de translation d’adresses :

- **NAT statique** : à une adresse IP source dans le privé on associe une adresse et une seule adresse IP publique.
- **NAT dynamique :** on définit une plage d’adresses IP publique qu’on attribue à des machines qui initient des connexions vers l’extérieur
- **PAT** : une seule adresse IP publique est utilisé pour toutes les connexions sortantes quelque soit la machine dans le réseau privé mais à chaque requête on associe un numéro de port.

NB: Ces différentes types de translation seront détaillées an TP

Chapitre 8 : Les protocoles applicatifs (HTTP, SMTP, DNS...)

Objectifs spécifiques du chapitre 8 : Les protocoles d'applications

1. Comprendre le fonctionnement de la couche application du modèle TCP/IP
2. Comprendre et mettre en œuvre le service DHCP en environnement CISCO et Linux
3. Comprendre et mettre en œuvre le service DNS en environnement CISCO et Linux
4. Comprendre et utiliser un service de messagerie (SMTP, POP, IMAP, MIME)
5. Comprendre et utiliser un service de transfert de fichier FTP (FileZilla client, FileZilla serveur, vsFTP, Proftpd)
6. Comprendre le fonctionnement du protocole HTTP à travers l'utilisation du serveur Apache et wireshark.
7. Comprendre l'encapsulation des messages applicatifs

Sommaire

8.1 Présentation du protocole DHCP

Problématique

- Dans un réseau, chaque station doit avoir ses propriétés TCP/IP configurées correctement. Il faut renseigner :
  - L'adresse IP,
  - Le masque réseau,
  - La passerelle par défaut,
  - L’adresse du serveur DNS, etc…
- Si l’administrateur du réseau doit effectuer ce paramétrage sur un grand nombre d'ordinateurs, cela peut induire des erreurs (ex: double utilisation d'une même adresse IP), et faire perdre du temps.

- Une solution: le protocole DHCP (Dynamic Host Configuration Protocol) permet de rendre automatique ces configurations.

Fonctionnement du protocole DHCP

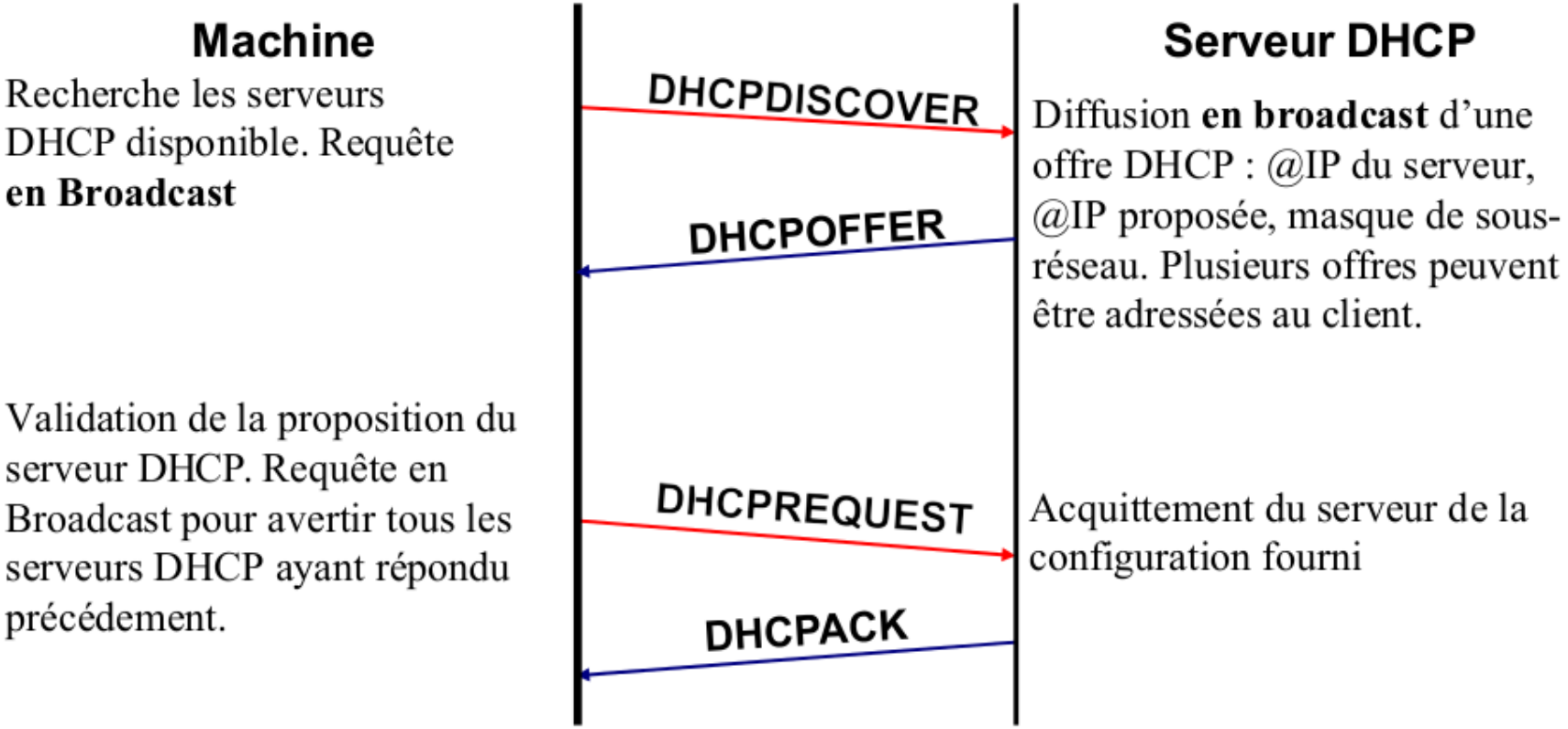


figure 8.1 Fonctionnement du protocole DHCP

Remarque

Il faut indiquer la plage d'adresses que le serveur DHCP est autorisé à distribuer.

- Les adresses IP distribuées ont **une date de début** et **une date de fin de validité**. C'est ce qu'on appelle **un « bail ».**

- **D’autre commande DHCP existe, par exemple :**

    DHCPRELEASE : Arrête le bail en cours. Ex : ipconfig /release

8.2 Présentation du protocole HTTP (Hyper Text Transfert Protocol)

Le but du protocole HTTP est de permettre un transfert de données (HTML) depuis un endroit du serveur jusqu'au client.

Comme tous les protocoles, la communication se fait en deux temps :

- Requête HTTP émise par le client (à l'aide d'un URL : nom du protocole, identifiant et mot de passe, nom du serveur, numéro de port, chemin d'accès à la ressource)

- Réponse HTTP envoyée par le serveur (après traitement de la requête)

Protocole	Mot de Passe (facultatif)	Nom du serveur	Port	Chemin de la ressource
http://	user:password@	www.ec2lt.sn	80	/ec2lt/reseau.html

figure 8.2 Les différents informations contenant dans une réponse envoyée par le serveur

8.3 Présentation du protocole DNS

Un site est toujours localisé sur Internet par l'adresse IP de la machine sur laquelle il se trouve. Mais on indique généralement au navigateur le nom de domaine du site que l'on souhaite visiter, et non pas son adresse IP.

Un navigateur web, pour se rendre sur un site, doit donc connaître l'adresse IP du nom de domaine correspondant. Il faut donc faire la correspondance entre un nom de domaine et son adresse IP.

Si les services de correspondance sont hors service, aucun navigateur ne pourra connaître l'adresse IP du site correspondant, et donc consulter ce site.

DNS est un serveur qui permet d'associer un nom de domaine à l'adresse IP d'une machine. Les noms de machines sont composés de plusieurs parties et se lisent de droite à gauche, par exemple : www.google.fr ou ftp.ec2lt.sn

Problématique

- En partant de droite le . Représente la racine.

- Le **.fr** est le domaine de premier niveau (TLD : Top Level Domain). Les abréviations des pays(.sn, .fr, .td, etc) s'appellent des **ccTDD** (country code TLD), tandis que les autres TLD(.com, .org, .net, .edu etc) s'appellent gTLD (generic TLD).

- Le nom du domaine est : .**google** ou .**ec2lt**

- Le **www** ou **ftp** est le nom de la machine qui offre le service. Ce nom dépend de l'administrateur qui choisit dans son domaine d'appeler une machine par tel ou tel nom.

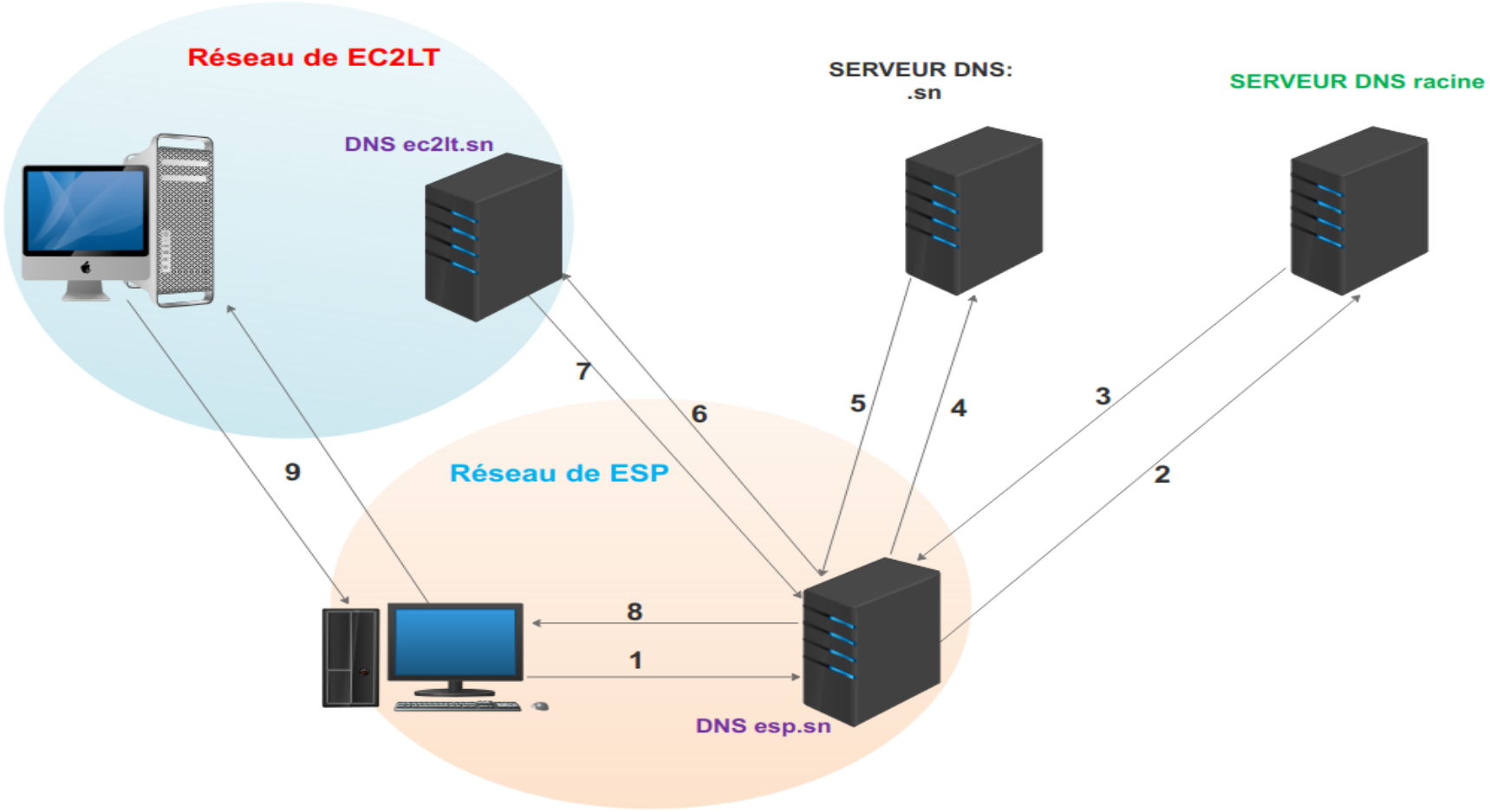


Figure 8.3 Principe de résolution de nom

(1) Quelle est l'adresse IP de ec2lt.sn ?

(2) Quel serveur DNS fait autorité sur le domaine .sn ?

(3) Réponse

(4) Quel serveur DNS fait autorité sur le domaine ec2lt.sn ?

(5) Réponse

(6) Quelle est l'adresse IP de www.ec2lt.sn?

(7) Réponse

(8) Voici l'adresse IP de www.ec2lt.sn!

(9) Connexion au serveur www.ec2lt.sn

Mise en place d'un serveur DNS avec des fonctionnalités minimales

Un domaine de nom internet peut être vu comme un ensemble de machines sous une administration unique qui s'occupe de donner des noms à ses machines.

Le service DNS intégre les règles suivantes à suivre :

1. un domaine a un nom qui ne doit pas comporter le caractére **.**
2. Un domaine peut comporter de sous-domaines et le nom complet d'un sous domaine est de la forme : **nomsousdomaine.nomdomaine**

exemple ec2lt.sn est le nom complet du sous domaine ec2lt du domaine sn

3. une machine dans un domaine a un nom d'hôte qui ne doit pas comporter de caractére . Et le nom complet d'une machine dans un domaine appelé son FQDN est de la forme : **nomhote.nomcompletdomaine**

exemple le FQDN **pc.ec2lt.sn** correspond au nom complet d'une machine dans le domaines ec2lt.sn

dont le d'hôte est **pc.**

Les informations stockées sur un serveur DNS sont appelés des enregistrements.

Type d'enregistrements possibles :

- **SOA:** permet de préciser le nom du domaine sur lequel le serveur DNS a autorité, le nom complet du serveur DNS ainsi que l'email de l'administrateur du domaine dont @ qui y figure doit être remplacé par .
- **A :** Hôte local. Utilisé pour lier un nom de domaine DNS avec une adresse IP.
- **PTR :** Pointeur(PTR). Utilisé pour lier une adresse IP à un nom domaine.
- **NS :** Serveur de nom : Utilisé pour lier un nom de domaine DNS avec le nom d'un ordinateur qui fait office d'un serveur DNS.
- **CNAME** : Nom canonique. Utilisé pour lier un nom de domaine DNS canonique avec un autre nom principal ou canonique.
- **MX**: Serveur de Messagerie. Utilisé pour lier un nom de domaine DNS avec le nom d'un ordinateur qui échange ou transmet du courrier pour un domaine.
- **SRV:** Utilisé pour déclarer les services
- **Naptr :** Utilisé pour déclarer des serveurs de téléphonie

Pour mettre en place d'un serveur DNS avec des fonctionnalités minimales on doit suivre les étapes suivantes:

1) faire un enregistrement de type SOA sur la machine serveur
2) faire un enregistrement de type NS sur le serveurs
3) faire au moins un enregistrement de type A pour faire le lien entre l'adresse IP du serveur DNS et son nom déclaré dans le type NS.
4) Faire des enregistrements de type A pour baptiser les machines du domaine

Installation

Tout simplement les commandes suivantes :

- **apt-get update** (mettre à jour la liste des sites sur lesquels on peut télécharger les paquets)
- **apt-get install bind9** (intaller la paquet et ses dépendances)

Configuration de bind

Dans notre cas, nous aurons un certain nombre de fichiers à configurer. L fichier principalement pour la configuration de Bind s'appelle **named.conf (/etc/bind/named.conf).**

named.conf

Ce fichier inclut le fichier est composé :

- Les options principales
- Les zones de recherche directe et de recherche inversée que l'on veut déclarer.

Fichiers de zone :

```
zone "ec2lt" {
        type master;
        file "/etc/bind/ec2lt.sn";
};


zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/ec2lt.inv";
};
```

- Recherche directe : **/etc/bind/ec2lt.sn**

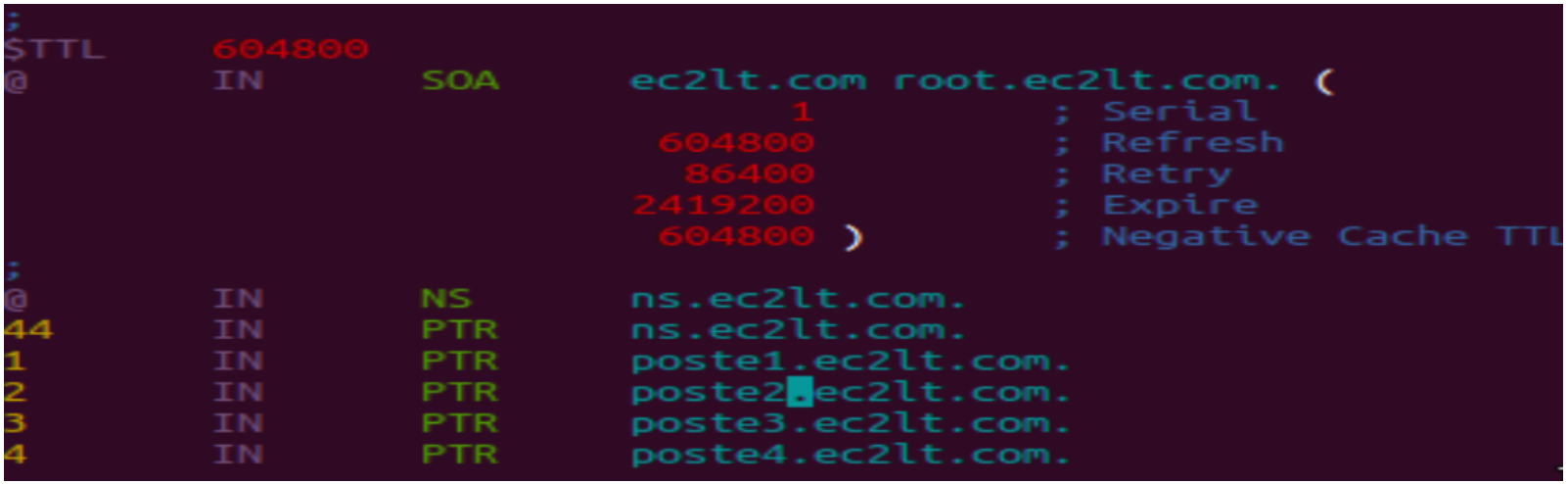
Ce fichier contient les noms de machines dans le domaine, mappée avce leur @ IP.

```
;
$TTL    604800
@       IN      SOA     ec2lt.com. root.ec2lt.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
; déclaration d'un serveur de nom et d'un serveur de Mail
@               IN      NS      ns.ec2lt.com.
@               IN      MX 10   mail.ec2lt.com.

;déclaration de type A
ns.ec2lt.com.   IN      A       192.168.1.44
mail.ec2lt.com. IN      A       192.168.1.54

;exemple d'enregistrement d'alias
www.ec2lt.com.  IN      CNAME   ns.ec2lt.com.
```

Ce fichier contient les @IP du réseau mappées avec leur nom dans le domaine.



Explications :

En-tête :

- $TTL : Durée de vie de la zone exprimée en secondes par défaut.
- @ IN SOA : désigne l'enregistrement de "début d'autorité"(Start Of authority)

  il est suivi du serveur DNS primaire et de l'adresse du responsable
- (XXXXXXXX; N° de version. Par habitude AAAAMMJJ(ex :20171127)

  XXXXXXXX; Refresh temps d'attente pour le rafraîchissement du DNS secondaire

  XXXXXXXX; Retry : d'attente du serveur secondaire pour refaire une requête. XXXXXXXX; Expire temps pendant lequel le DNS secondaire doit garder les zones

  XXXXXXXXX;) TTL(Time To Live) temps de vie par défaut de tous les enregistrements

Vérification de la configuration :

On peut utiliser les commandes suivantes pour vérifier d'une part la configuration et d'autre part les zones de recherche directes et inversées **:**

- **named-checkconf** /etc/bind/named.conf
- **named-checkzone** ec2lt.sn /etc/bind/ec2lt.sn
- **named-checkzone** ec2lt.sn /etc/bind/ec2lt.inv

Tant que vous n'avez pas Ok comme réponse après ces commandes, vous devez corriger les erreurs qui se trouvent dans vos fichiers de configuration ou de zones.

Le fichier **/etc/resolv.conf :**

Vous changez le contenu du fichier /etc/resolv.conf en lui indiquant le domaine auquel vous appartenez et l'adresse IP du serveur DNS à tester(127.0.0.1 pour votre machine).

```
nameserver 192.168.1.44
nameserver 127.0.0.1
```

La commande **host :**

Cette commande permet de tester le serveur DNS. Pour voir les détails de la commande **host,** voyez la page de manuel (**man host**).

Test de la recherche direct :

- **host poste4** …. devrait vous donner 192.168.1.4

Test de la recherche inversée :

- **host 192.168.1.44** ...devrait vous ns.ec2lt.sn

Test de noms canoniques (CNAME) :

- **host www** ...devrait vous donner 192.168.1.44

Test des forwarders (DNS publics qui prennent le relai) :

- **host** www.google.fr ...devrait vous donner une adresse IP publique

La commande **nslookup :**

La commande permet aussi de tester le DNS. Tapez **nslookup** :

>help

>set type=NS pour pouvoir lister les entr2es de type NS (sinon remplacez par autre chose ANY pour tous)

>ec2lt.sn pour tester sur le domaine ec2lt.sn

Présentation des protocoles (SMTP, POP, IMAP, MIME)

Introduction

Proposé en 1982, SMTP (Simple Mail Transfer Protocol) est un protocole largement déployé dans les réseaux Internet.

SMTP transporte les messages sur les différents réseaux (TCP/IP ou autres réseaux) SMTP gère principalement le serveur de courrier électronique (email). Le client utilise SMTP pour émettre les messages au serveur de courrier électronique. Quant à la réception de messages électronique, le client utilise les autres protocoles tels que POP (Post Office Protocol), IMAP (Internet Message Access Protocol) et les systèmes propriétaires (Microsoft Exchange et Lotus Note) ou libres (evolution,thunderbird).

Modèle SMTP

- Le modèle principal SMTP consiste en cinq parties suivantes :
  - MUA (Mail User Agent) est un client de messagerie ; Agent utilisateur de messagerie soumet le courrier électronique au serveur MSA.
  - MSA (Mail Soumission Agent) est un serveur de messagerie et relai qui transfère le courrier au MTA.
  - MTA (Mail Transfer Agent) est un serveur de messagerie et commutateur de courriers ;
  - MSA et MTA sont souvent intégrés dans un seul serveur.

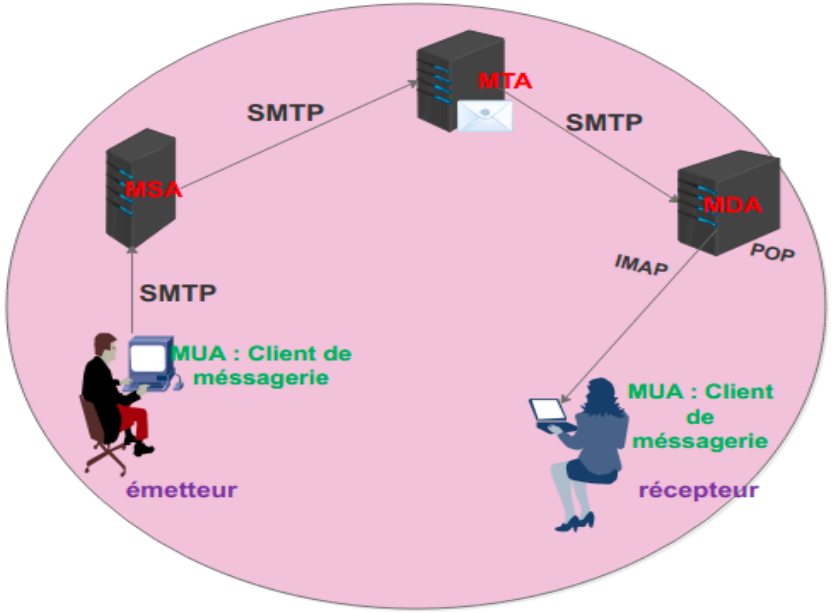


figure 8.4 Architecture générale d'un service de messagerie

Le modèle principal SMTP consiste en cinq parties suivantes :

- MX (Mail Exchanger) est un serveur de messagerie du destinataire qui accepte le courrier et le transfère au MDA.
- MDA (Mail Delivery Agent) est un serveur de messagerie du destinataire qui délivre des messages locaux.
- MX et MDA sont souvent intégrés dans un seul serveur.

Fonctionnement principal

- MUA (Mail User Agent) soumet le courrier électronique au serveur MSA en passant par SMTP/TCP port 587 (ou port 25 avec l'ancienne version)
- MSA (Mail Soumission Agent) transfère le courrier au MTA
- MTA (Mail Transfer Agent) recherche d'abord la localisation du destinataire du courrier par la technique DNS (type MX).
- A l'aide du retour de RR (Registre Record) – nom de hôte, MTA recherche l'adresse IP (type A). Ensuite, MTA connecte au serveur du destinataire étant un client SMTP
- MX (Mail Exchanger) accepte le courrier et le transmet à MDA
- MDA (Mail Delivery Agent) est le responsable de courriers locaux
- MDA enregistre les courriers en format mailbox ou maildir
- MDA stocke les courriers ou les transmets au réseau local par le protocole LMTP (Local Mail Transfer Protocol)

- Une fois qu'un courrier est arrivé au serveur de messagerie du destinataire (serveur local), on utilise un client de messagerie qui se base sur le protocole POP (Post Office Protocol) ou IMAP(Internet Message Access Protocol).
- Le client (MUA) doit s'authentifier pour retirer ses courriers stockés dans le serveur local.

Exemple de MUA libres : les logiciels thunderbird kmail et evolution sont des clients libres de messagerie qu'on peut télécharger et installer gratuitement

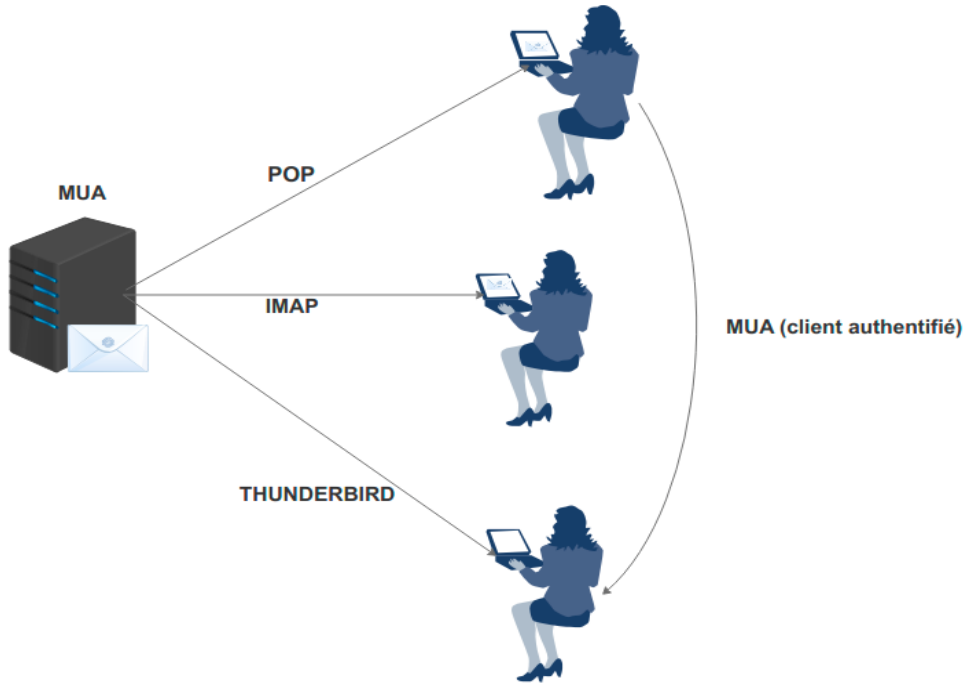


Figure 8.5 Utilisation d'un client de messagerie pour l'envoi et réception de mails

Protocole SMTP

Structure client-serveur

Un utilisateur souhaite émettre un courrier. Le client SMTP se charge de trouver le destinataire en échangeant les commandes/réponses avec le serveur SMTP. Le serveur SMTP local se charge de transférer le courrier.

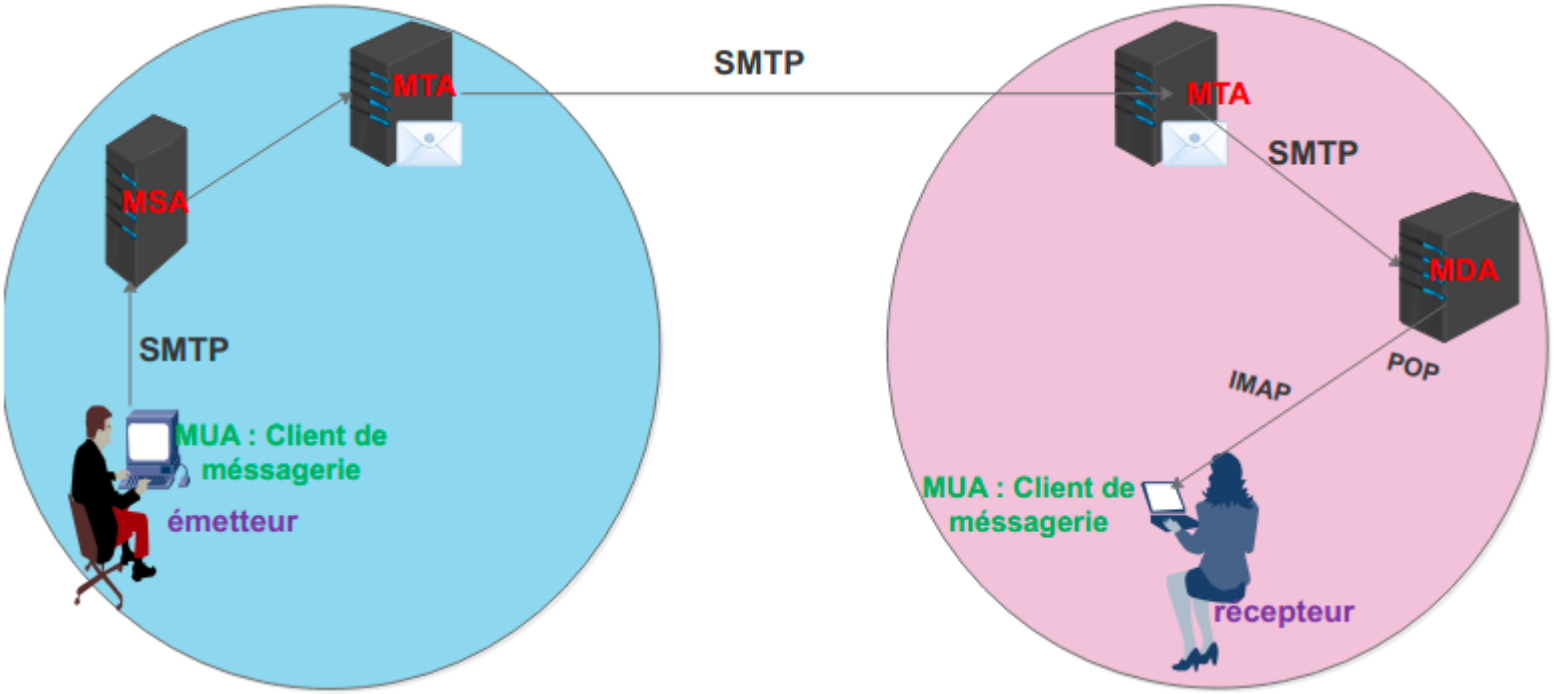


Figure 8.6 Envoi et réception de mails entre plusieurs domaines

Adresses de courrier

Les adresses globales de courrier sont définies par les standards RFC 5321 et 5322 une adresse se compose en deux parties : le nom de boîte à lettre et le nom de domaine DNS

exemple:

aly.tirera@ec2lt.sn

- La taille du nom de boîte à lettre ne doit pas dépasser 64 octets ; le point « . » est autorisé
- Le système de nom de domaines (DNS) permet de déterminer le serveur de courrier (RR de type MX).

Format des messages

Le format des messages/courriers est constitué de deux parties distinctes.

Figure 8.7 Format des messages

- L'en-tête (return-path + received) : la partie return-path concerne les informations ajoutées à travers le routage ; la partie received concerne la traçabilité du chemin parcouru.

- Le corps est séparé de l'en-tête par au moins une ligne vide. Il est défini par MIME (Multipurpose Internet Mail Extensions) MIME définit le format d'en-têtes non-US ASCII.

Format des messages – entête

Il y a au moins trois lignes obligatoires dans l'en-tête mais d'autres lignes sont aussi prévues dans les différentes normes:

- From: adresse émetteur
- To: adresse destinataire
- Date: date de création du message

On a la possibilité de créer des en-têtes propriétaires à condition de les faire précéder de X-

Exemple d'un entête

Delivered-To: fatu.diabakhate@ec2lt.com
Date: Thu, 21 Mar 2002 15:15:39+0100
From: kachallah <kachou@rtn.sn>
Organization: EDF-DER
X-Accept-Language: fr
MIME-Version: 1.0
To: th-rntl-accord@rd.francetelecom.com
Cc: latyr <latyr@rtn.sn>
Subject: Un premier cours sur la messageries

Corps du courrier électronique constitué de lignes (de longue au plus égale à 1000 caractères).

Objectifs de MIME

Le protocole MIME permet d'attacher les fichiers multmédia à des courriers

Types principaux de données MIME

Cinq types de données 'discrets' (avec sous types) par RFC 2046

- Type texte : données lisibles : text/plain [RFC2646] ; text/html [RFC2854]
- Type image : différents codages image : image/jpeg ; image/gif
- Type son : différents codages 'audio' : audio/basic (MIC mu 8000 Hz 8 bits)
- Type vidéo : images animées : video/mpeg
- Type application : les données qui restent.

Commandes de requêtes de client SMTP

- Chaque requête (un message du protocole SMTP) correspond à une ligne de texte terminée par CRLF (' carriage return ' code 13 et ' line feed code '10).

- HELO <SP> <domaine> <CRLF> : L'ouverture de session entre le client et le serveur (le message contient le nom de domaine FQDN du client).

- MAIL <SP> FROM: <route-retour> <CRLF> : Définit l'adresse mail de l'émetteur (utilisé pour le retour éventuel d'erreurs).

- RCPT <SP> TO: <route-aller> <CRLF> : Définit l'adresse d'un destinataire (le routage du courrier est possible en donnant une liste de MTA à visiter : routage par la source @Hote_1,@ Hote_2,usager@ Hote$_3$)

- DATA <CRLF>: Définit l'enveloppe (l'entête) et le corps (le texte) du message.

- QUIT <CRLF>: Termine un courrier.
- SEND <SP> FROM: <route-retour> <CRLF> : L'envoie d'un message sur un terminal.
  * SEND or MAIL (ou « SEND and MAIL ») : l'envoie sur un terminal ou une boîte à lettre.

- Quelques commandes de requêtes annexes.

- RSET : Commande pour abandonner le courrier en cours de transmission et restaurer la connexion.

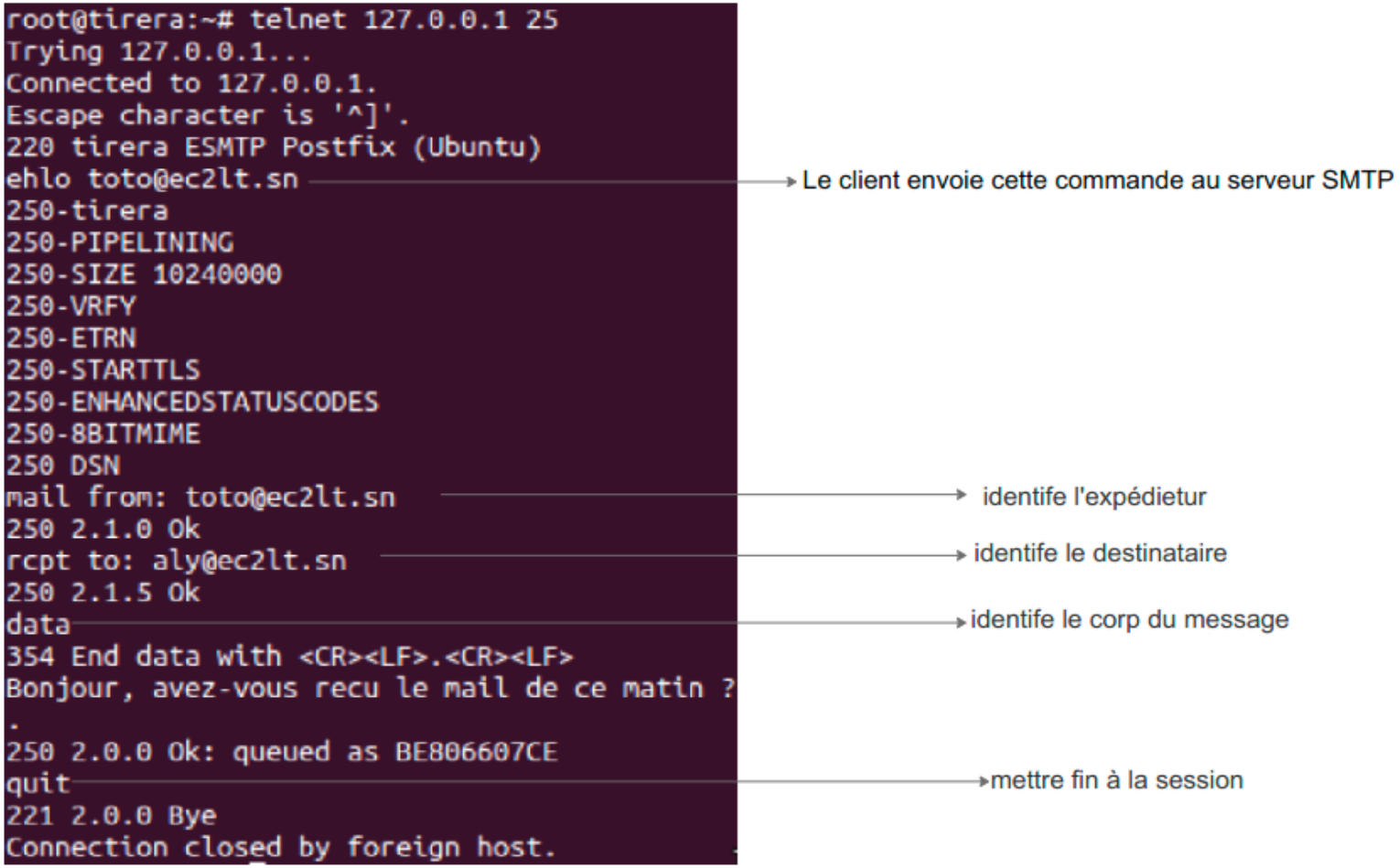
- VRFY : Commande pour vérifier une adresse de destinataire sans lui transmettre de courrier (utilisable pour déterminer la cause d'un problème).

- NOOP : Commande vide qui oblige simplement le serveur à répondre 200 OK.

- EXPN : Expansion d'une liste de diffusion ('mailing list').

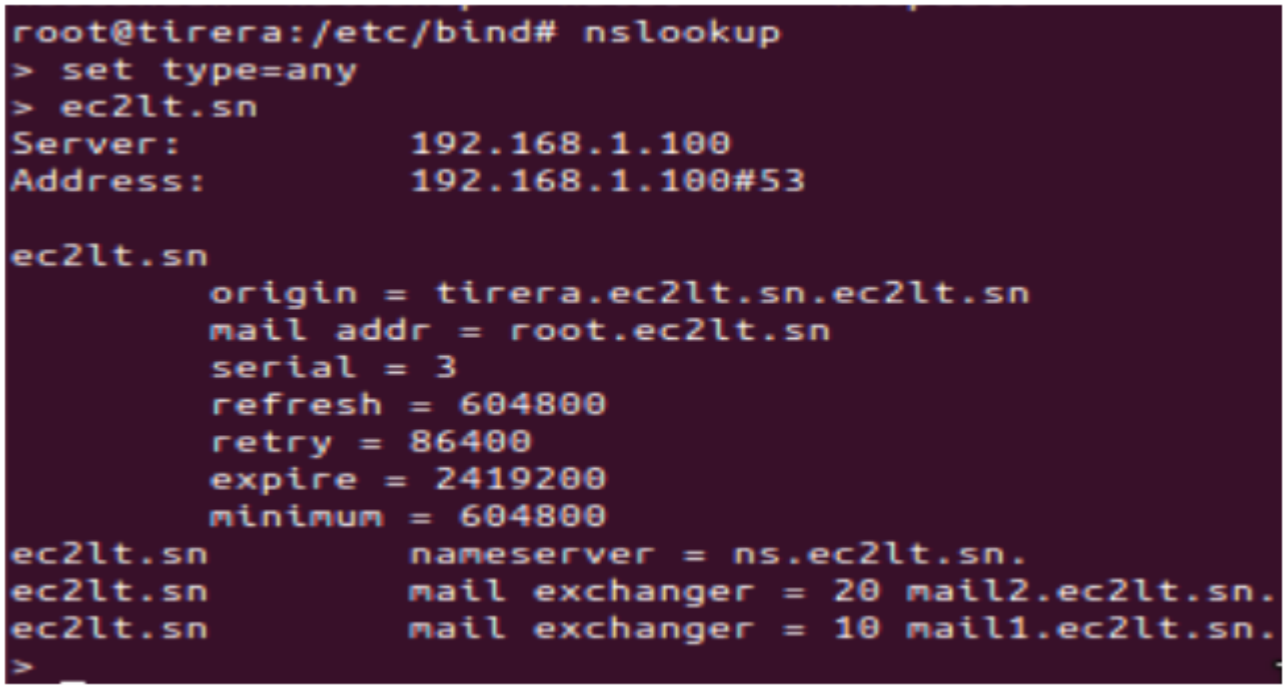
- TURN : Inversion des rôles client et serveur pour envoyer du courrier dans l'autre sens sans ouvrir une nouvelle connexion TCP.

Exemple type de transmission d'un courrier



Interactions avec DNS

- Le protocole SMTP est dépendant du DNS comme tout protocole utilisant des noms de sites
  - un exemple:



Tous les mails adressés à ec2lt.sn vont être enregistrés dans le serveur mail1.ec2lt.sn (en priorité). Si mail1.ec2lt.sn est en panne, mail2.ec2lt.sn reçoivent les mails.

Protocoles POP et IMAP

Introduction

- Les protocoles POP (Post Office Protocol) et IMAP (Internet Message Access Protocol) sont les protocoles dérivés du protocole SMTP.
- Ils sont utilisés pour relever du courrier dans une boite aux lettres
- Fonctions de transfert de courrier d'un serveur de messagerie vers un client de messagerie.
- Fonctions de gestion des archives de courrier (liste de messages en attente, destruction de message...)

POP (Post Office Protocol)

- Le protocole POP3 (Post Office Protocol version 3), le standard RFC 1939, est largement implémenté. C'est un protocole le plus simple.
- Le principe : le client connecte au serveur pour relever définitivement les messages en attente via TCP (port 110) ; les messages sont transmis vers la boîte aux lettres du client.
- Afin d'éviter tous problèmes de sécurité, le client doit s'authentifier. Les mesures de TLS (Transport Layer Security) et SSL (Secure Socket Layer) pourront être utilisées.

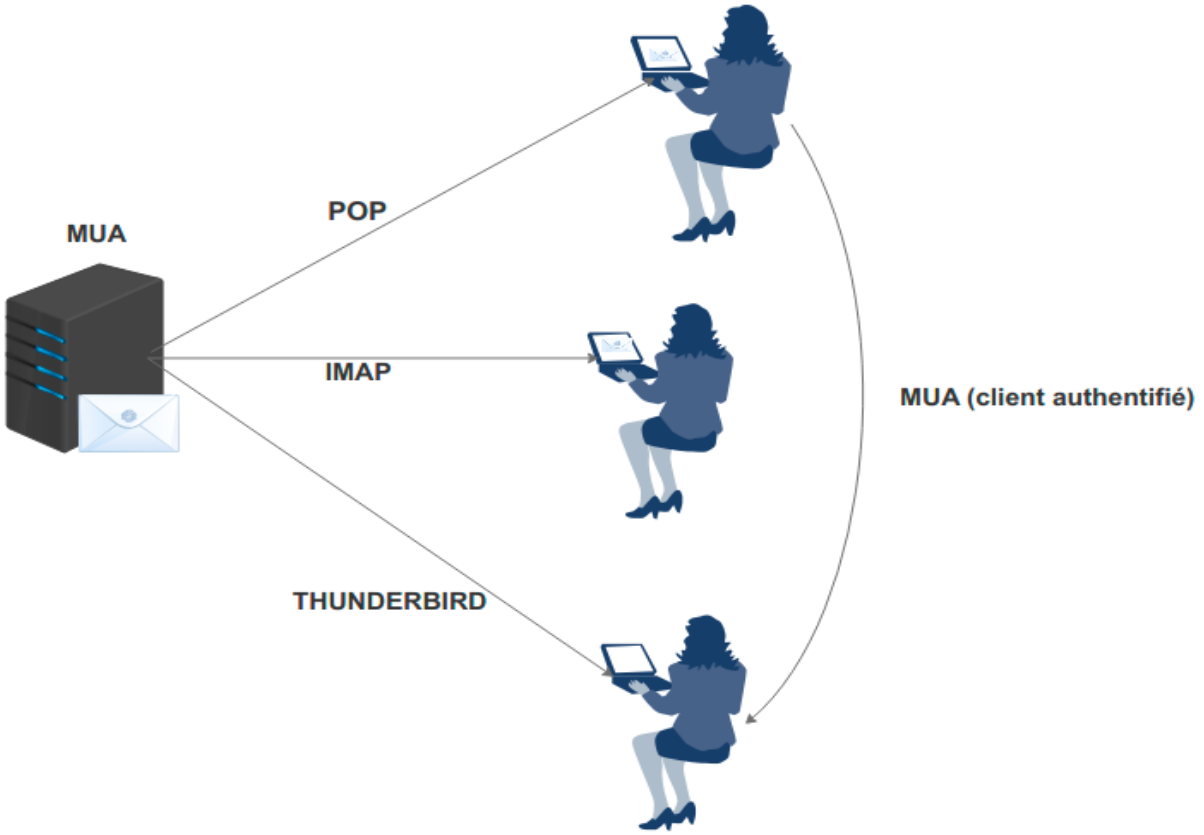


Figure 8.7 Utilisation des protocoles POP et IMAP pour la réception des mails

Commandes POP3 principales

- USER Fourniture du nom de la boite aux lettres
- PASS Fourniture du mot de passe en clair
- APOP Fourniture cryptée du mot de passe
- STAT Nombre de messages dans la boite

- LIST Liste des messages présents
- RETR Transfert du message n
- DELE Marquage message pour la suppression
- LAST Numéro du dernier message consulté
- RSET Annulation des actions d'une session
- QUIT Fin de session.

On peut utiliser telnet pour interagir avec un serveur de POP comme le montre la figure ci-après

figure 8.8 Utilisation de telnet sur un serveur de messagerie POP

IMAP4 (Internet Message Access Protocol)

- Le protocole IMAP4 (Internet Message Access Protocol), le standard RFC 3501 qui remplace le RFC 2060, est largement implémenté par client. C'est un protocole le plus complet.
- Le principe :
  - deux modes (connected and disconnected modes) permettent à un client de connecter (ou déconnecter, les courriers récupérés durant une connexion sont dans un cache) au serveur pour relever les messages en attente via TCP (port 143)
  - les messages sont souvent restés dans la boîte aux lettres du serveur de messagerie; ce mode permet à un utilisateur de consulter ses courriers via plusieurs machines différentes.
  - La boîte aux lettres du serveur peut être consultée par plusieurs clients.

Commandes IMAP4 principales

- AUTHENTICATE : Mécanisme d'authentification choisi.
- LOGIN : Usager mot de passe.
- LOGOUT : Fin de session IMAP.
- CREATE/DELETE/RENAME : Nom de boite aux lettres.
- SELECT/EXAMINE : Nom de boite aux lettres.
- LIST/LSUB/STATUS : Etat de boite aux lettres.
- EXPUNGE/CLOSE : Détruit les messages marqués (et ferme).
- SEARCH : Recherche de message sur différents critères.
- FETCH : Récupération des données concernant un courrier.
- COPY : Recopie d'un message d 'une boite aux lettres dans une autre.

- CAPABILITY : Liste des fonctions implantées d’un serveur.
- NOOP : Opération vide.

Implantations

Serveurs de messagerie libre(MTA)

Les serveurs MTA libres peuvent être implantés:

- sendmail,
- **Postfix est le MTA le plus utilisé dans le monde libre aujourd’hui**
- Qmail ,
- Exim,

Serveurs de messagerie propriétaire (MTA)

- Les messageries d’entreprise sont souvent intégrés dans des sites bureautiques ou serveurs WEB.
- Microsoft Exchange / Internet Information Service
- Lotus Notes/Domino (IBM)
- IMAIL

Serveurs de délivrance de messages (MDA)

Le MDA permet le stockage (formats mbox ou maildir) et le filtrage des messages et l'envoi de messages de réponse automatique.

Voici quelques implémentations :

- **procmail**, **maildrop**, **deliver** et **mailfilter**.

Les MDA incorporent généralement des outils de protection contre les virus et le SPAM (très grande variété de produits)

Client de messagerie (MUA)

Il existe deux types de clients de messagerie :

- Clients lourds : qui nécessitent l'installation d'une application particulière sur le poste client par exemple Outlook Express, Mozilla Thunderbird, Eudora, foxmail …
- Clients légers ou webmail permettant de consulter son mail à travers le web par exemple **SquirrelMail** et **roundcube**