TCP/IP - Protocoles de base

CentralWeb 56, Boulevard Pereire - 75017 PARIS
Tel : +33 01.46.31.44.88 Fax: +33 01.46.31.49.99
info@centralweb.fr / www.centralweb.fr

Plan

- INTRODUCTION
- CONCEPTS DE L'INTERCONNEXION
- L'ADRESSAGE INTERNET
- ARP : PROTOCOLE DE RESOLUTION D'ADRESSE
- RARP : PROTOCOLE DE RESOLUTION D'ADRESSE INVERSE
- LE PROTOCOLE INTERNET
- ROUTAGE DES DATAGRAMMES
- LE SOUS-ADRESSAGE
- LE PROTOCOLE ICMP
- UDP : LE PROTOCOLE TRANSPORT DATAGRAM
- TCP : LE PROTOCOLE DE TRANSPORT FIABLE
- CONCLUSION

Internetworking with TCP/IP, Douglas E. Comer
TCP/IP network administration, Craig Hunt

Introduction

- TCP/IP : but = interconnexion de réseaux sur une base planétaire
- Technologie issue des années 1970, de projets DARPA
- Aujourd'hui : 100000 réseaux interconnectés, plusieurs millions de machines, plusieurs dizaines de millions d'utilisateurs de "l'Internet".
- Interconnecte divers réseaux : Ethernet, T.R., X25, FR, FDDI, etc.
- La technologie est constituée par des protocoles de base (suite TCP/IP) qui offrent les services de base du transfert des données :
- transport de datagrammes : service élémentaire de la commutation de paquets.
- transport de messages sécurisés : service orienté connexion permettant d'acheminer des données en garantissant leur intégrité
- adaptation de la technologie TCP / IP à la plupart des interfaces matérielles.
- Ces services de base sont indépendants du support de transmission; adaptables à toute sorte de media depuis les réseaux locaux jusqu'aux réseaux longue distance.

Introduction

- **Interconnexion universelle : les machines ont une adresse unique sur l'Internet. Deux machines reliées au réseau, communiquent grâce aux autres noeuds du réseau qui routent de manière coopérative sur la base de l'adresse destinataire.**
- **Interconnexion d'égal à égal (peer to peer systems) : il n'y a pas de machines prioritaires (en opposition à une structure hiérarchique).**
- **Dans le cadre du transport sécurisé, les acquittements sont effectués entre les systèmes finaux (source et destinataire) plutôt que continuellement entre chaque noeud relayant les messages.**
- **Applications standards bâties sur la technologie de base : courrier électronique, transfert de fichier, émulation terminal, etc.**
- **Technologie publique et largement diffusée au travers de RFC's.**
- **Indépendante des constructeurs et disponible sur tous types de matériel (micro, station, super-calculateur et équipements de réseaux)**
- **Largement validée depuis de nombreuses années dans un monde hétérogène.**

Concepts de l’interconnexion

- **Point de départ : les réseaux interconnectés sont de nature diverse**
- **Les différences entre tous ces réseaux ne doivent pas apparaître à l'utilisateur de l'interconnexion.**
- **Abstraction à chaque niveau de fonctionnalité (couches de protocoles) qui encapsule les fonctionnalités de niveau inférieur**
- **Affranchit l'utilisateur des détails relatifs aux couches inférieures et finalement au réseau lui-même (couche physique).**
- **Les premiers systèmes d'interconnexion ont traité le problème au niveau applicatif : messagerie relayant le message de noeud en noeud. Cette solution présente plusieurs inconvénients :**
- **si les applications interfacent elles-mêmes le réseau (aspects physiques), elles sont victimes de toute modification de celui-ci,**
- **plusieurs applications différentes sur une même machine dupliquent l'accès au réseau,**
- **lorsque le réseau devient important, il est impossible de mettre en oeuvre toutes les applications nécessaires à l'interconnexion sur tous les noeuds des réseaux.**

Concepts de l'interconnexion (suite)

- **Alternative à cette solution : mise en oeuvre de l'interconnexion au niveau des protocoles gérant la couche réseau de ces systèmes.**
- **Avantage considérable : les données sont routées par les noeuds intermédiaires sans que ces noeuds aient la moindre connaissance des applications responsables des ces données**
- **Autres avantages :**
  - **la commutation est effectuée sur la base de paquets de petite taille plutôt que sur la totalité de fichiers pouvant être de taille très importante,**
  - **le système est flexible puisqu'on peut facilement introduire de nouveaux interfaces physiques en adaptant la couche réseau alors que les applications demeurent inchangées,**
  - **les protocoles peuvent être modifiés sans que les applications soient affectées.**

Concepts de l’interconnexion (suite)

- **Le concept d'interconnexion ou d'*internet* repose sur la mise en oeuvre d'une couche réseau masquant les détails de la communication physique du réseau et détachant les applications des problèmes de routage.**

- **L'interconnexion : faire transiter des informations depuis un réseau vers un autre réseau par des noeuds spécialisés appelés passerelles (*gateway*) ou routeurs (*router*)**

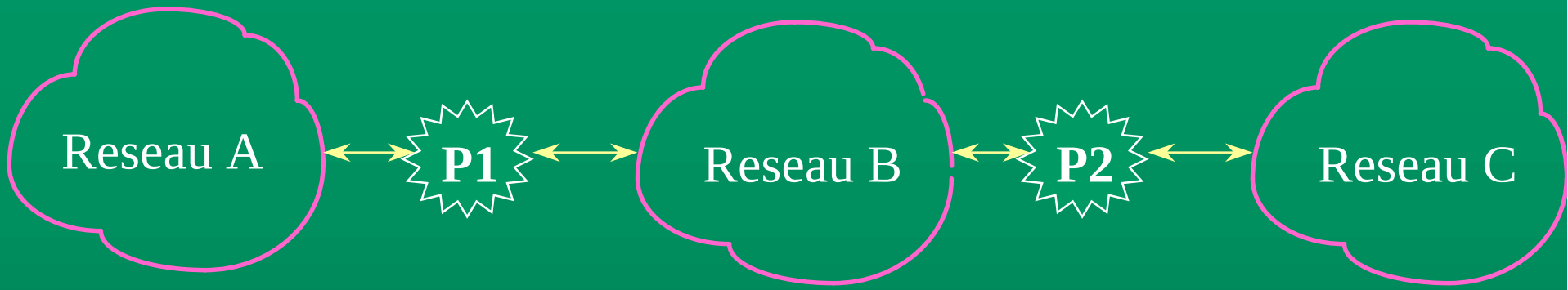
Concepts de l'interconnexion (suite)

- **Les routeurs possèdent une connexion sur chacun des réseaux:**

La passerelle P interconnecte les réseaux A et B.

- **Le rôle de la paserelle P est de transférer sur le réseau B, les paquets circulant sur le réseau A et destinés au réseau B et inversement.**

Concepts de l'interconnexion (suite)

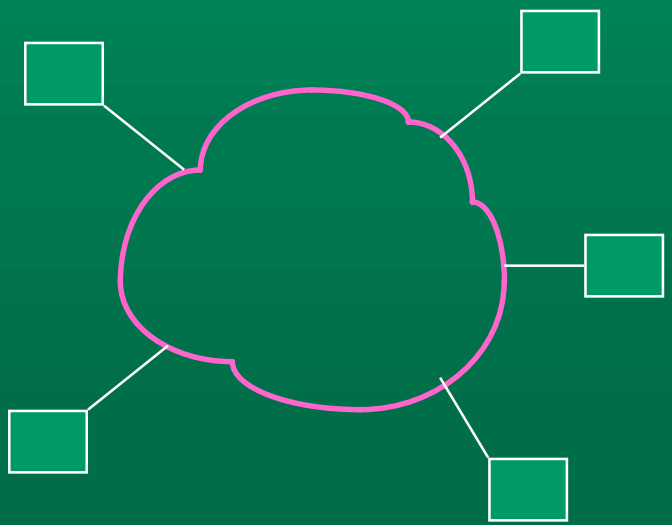


- **P1 transfère sur le réseau B, les paquets circulant sur le réseau A et destinés aux réseaux B et C**
- **P1 doit avoir connaissance de la topologie du réseau; à savoir que C est accessible depuis le réseau B.**
- **Le routage n'est pas effectué sur la base de la machine destinataire mais sur la base du réseau destinataire**

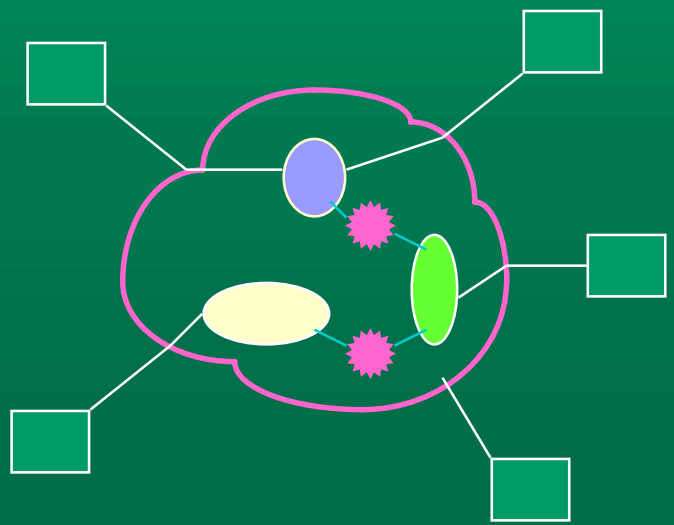
Concepts de l'interconnexion (suite)

- **A l'intérieur de chaque réseau, les noeuds utilisent la technologie spécifique de leur réseau (Ethernet, X25, etc)**

- **Le logiciel d'interconnexion (couche réseau) encapsule ces spécificités et offre un service commun à tous les applicatifs, faisant apparaître l'ensemble de ces réseaux disparates comme un seul et unique réseau.**



Vue utilisateur



Vue réelle du réseau

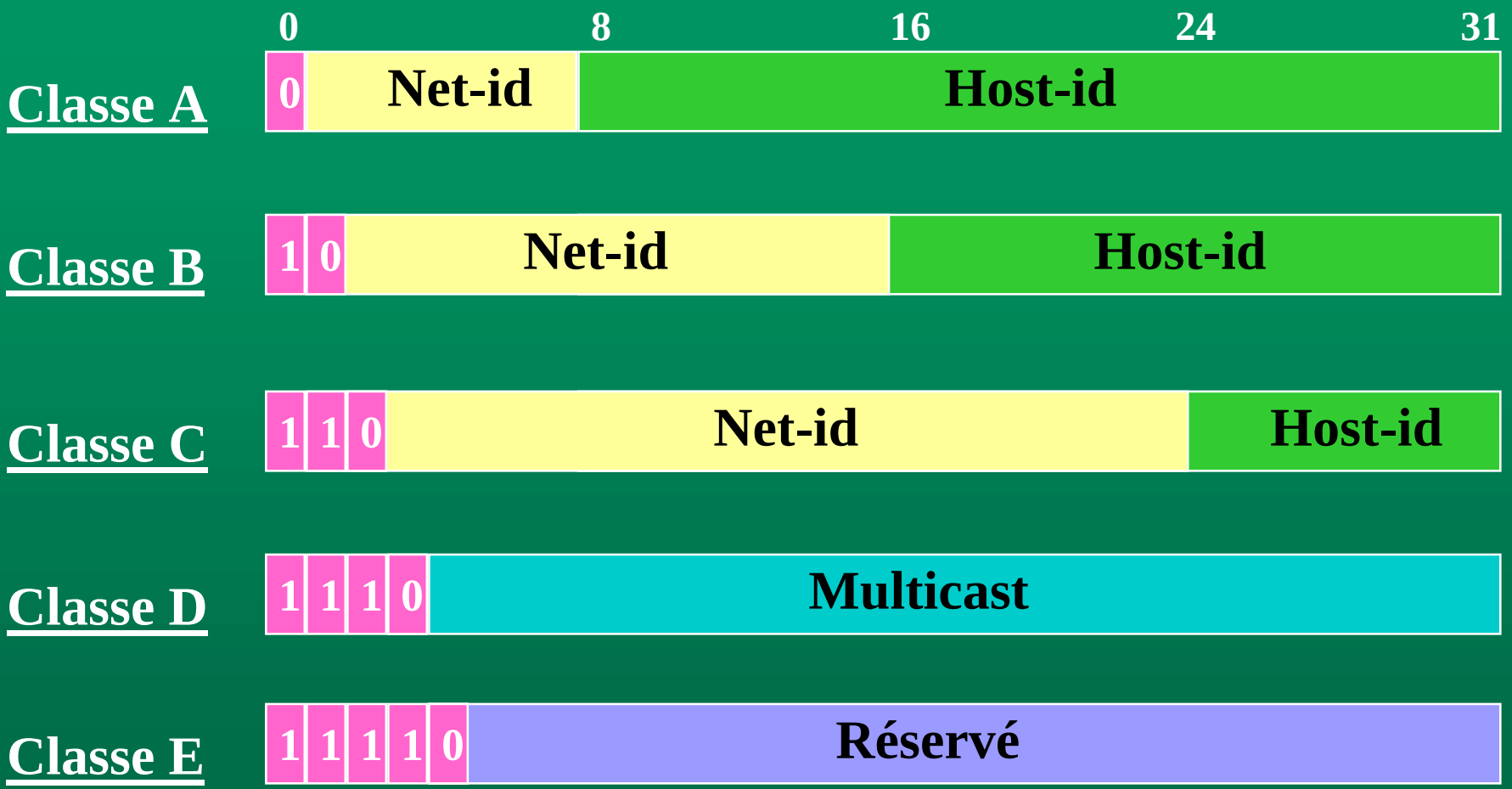
L’adressage Internet

- **But : fournir un service de communication universel permettant à toute machine de communiquer avec toute autre machine de l’interconnexion**
- **Une machine doit être accessible aussi bien par des humains que par d'autres machines**
- **Une machine doit pouvoir être identifiée par :**
  - **un nom (mnémotechnique pour les utilisateurs),**
  - **une adresse qui doit être un identificateur universel de la machine,**
  - **une route précisant comment la machine peut être atteinte.**

L’adressage Internet

- **Solution : adressage binaire compact assurant un routage efficace**
- **Adressage "à plat" par opposition à un adressage hiérarchisé permettant la mise en oeuvre de l'interconnexion d'égal à égal**
- **Utilisation de noms pour identifier des machines (réalisée à un autre niveau que les protocoles de base)**
- **Les classes d'adressage**
  - **Une adresse = 32 bits dite "internet address" ou "IP address" constituée d'une paire (netid, hostid) où netid identifie un réseau et hostid identifie une machine sur ce réseau.**
  - **Cette paire est structurée de manière à définir cinq classes d'adresse**

L’adressage Internet (suite)



L’adressage Internet (suite)

- **Notation décimale**

L'interface utilisateur concernant les adresses IP consiste en la notation de quatre entiers décimaux séparés par un point, chaque entier représentant un octet de l'adresse IP :

10000000 00001010 00000010 00011110 est écrit :

128.10.2.30

- **Adresses particulières**
  - **Adresses réseau : adresse IP dont la partie hostid ne comprend que des zéros; => la valeur zéro ne peut être attribuée à une machine réelle : 192.20.0.0 désigne le réseau de classe B 192.20.**
  - **Adresse machine locale : adresse IP dont le champ réseau (netid) ne contient que des zéros;**
  - **hostid = 0 (=> tout à zéro), l'adresse est utilisée au démarrage du système afin de connaître l'adresse IP (Cf RARP).**

L’adressage Internet (suite)

- **hostid != 0, hostid spécifie l'adresse physique de la machine (si la longueur le permet; c'est le cas pour T. R., ce n'est pas possible avec Ethernet). permet de ne pas utiliser RARP (ne franchit pas les ponts) n'est valide qu'au démarrage du système pour des stations ne connaissant pas leur adresse IP.**

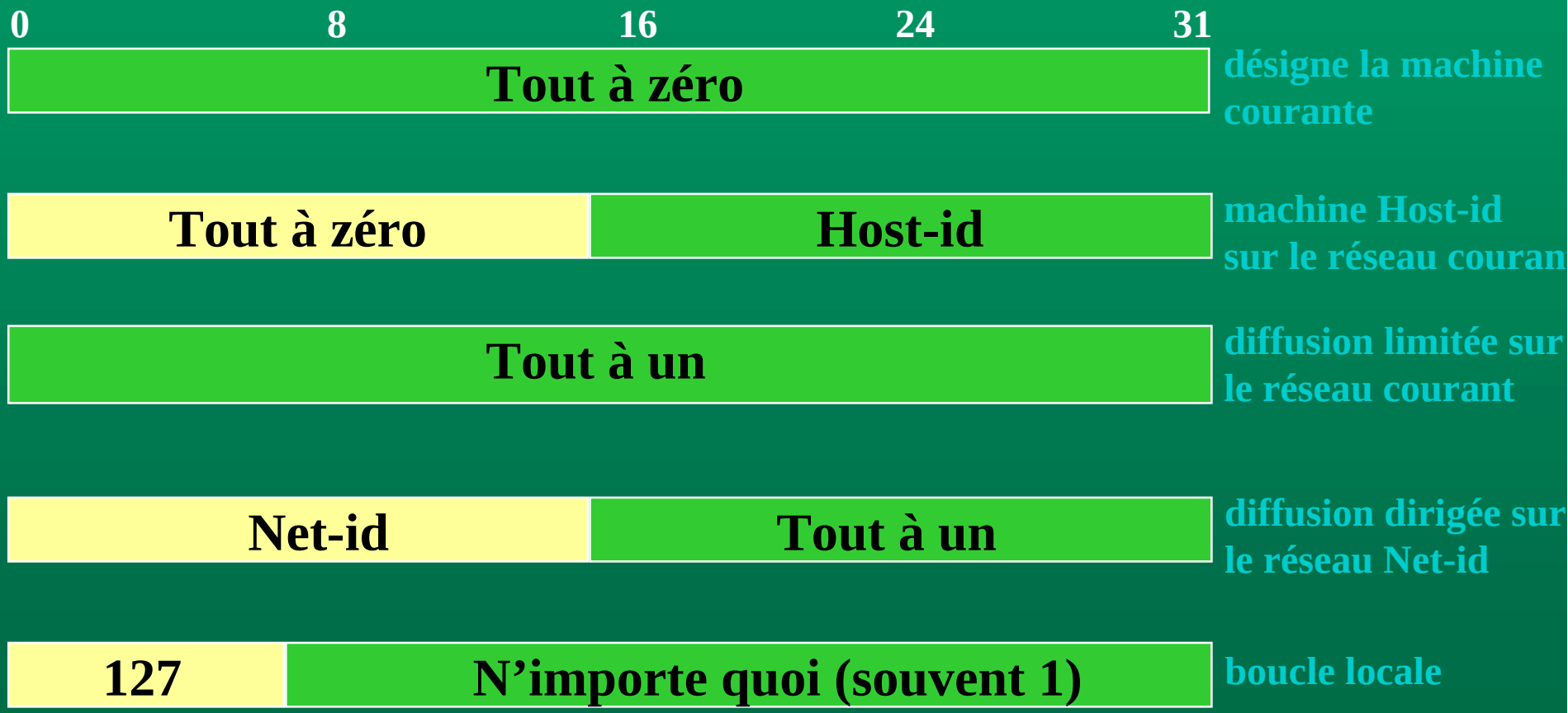
- **Adresses de diffusion : la partie hostid ne contient que des 1**
- **Adresse de diffusion limitée : netid ne contient que des 1 : l'adresse constituée concerne uniquement le réseau physique associé**
- **L'adresse de diffusion dirigée : netid est une adresse réseau spécifique => la diffusion concerne toutes les machines situées sur le réseau spécifié : 192.20.255.255 désigne toutes les machines du réseau 192.20.**
- **En conséquence, une adresse IP dont la valeur hostid ne comprend que des 1 ne peut être attribuée à une machine réelle.**

L’adressage Internet (suite)

- **Adresse de boucle locale : l'adresse réseau 127.0.0.0 est réservée pour la désignation de la machine locale, c'est à dire la communication intra-machine. Une adresse réseau 127 ne doit, en conséquence, jamais être véhiculée sur un réseau et un routeur ne doit jamais router un datagramme pour le réseau 127.**

L’adressage Internet (suite)

- **Résumé**



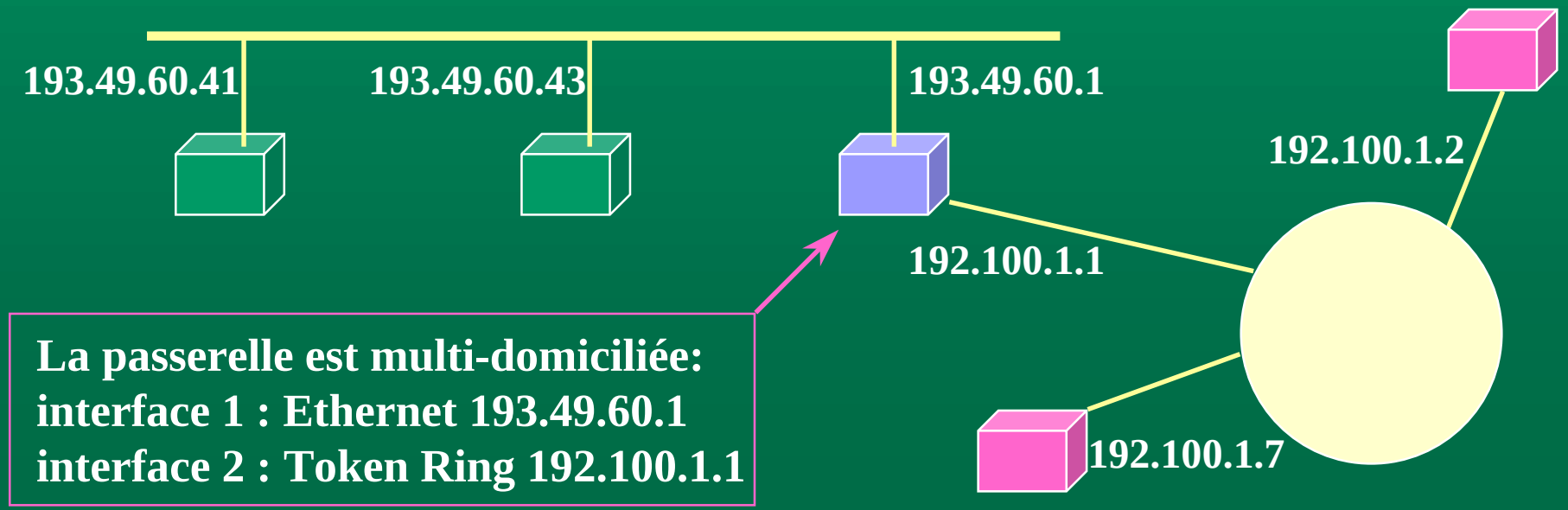
L’adressage Internet (suite)

☞ **Adresses et connexions**

Une adresse IP => une interface physique => une connexion réseau.

S'applique particulièrement aux routeurs qui possèdent par définition plusieurs connexions à des réseaux différents

A une machine, est associé un certain nombre N d'adresses IP. Si N > 0 la machine (ou passerelle) est multi-domiciliée.

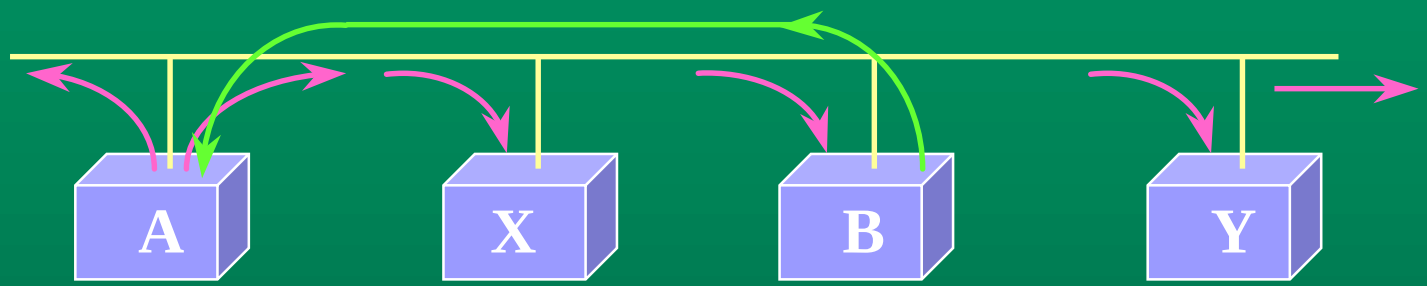


ARP: Address Resolution Protocol

- **Le besoin**
  - La communication entre machines ne peut s'effectuer qu'à travers l'interface physique
  - Les applicatifs ne connaissant que des adresses IP, comment établir le lien adresse IP / adresse physique?
- **La solution : ARP**
  - Mise en place dans TCP/IP d'un protocole de bas niveau appelé Adress Resolution Protocol (ARP)
  - Rôle de ARP : fournir à une machine donnée l'adresse physique d'une autre machine située sur le même réseau à partir de l'adresse IP de la machine destinatrice
- **LA technique :**
  - Diffusion d'adresse sur le réseau physique
  - La machine d'adresse IP émet un message contenant son adresse physique
  - Les machines non concernées ne répondent pas
  - Gestion cache pour ne pas effectuer de requête ARP à chaque émission

ARP: Address Resolution Protocol

- **L'association adresse physique - adresse IP de l'émetteur est incluse dans la requête ARP de manière à ce que les récepteurs enregistrent l'association dans leur propre mémoire cache**



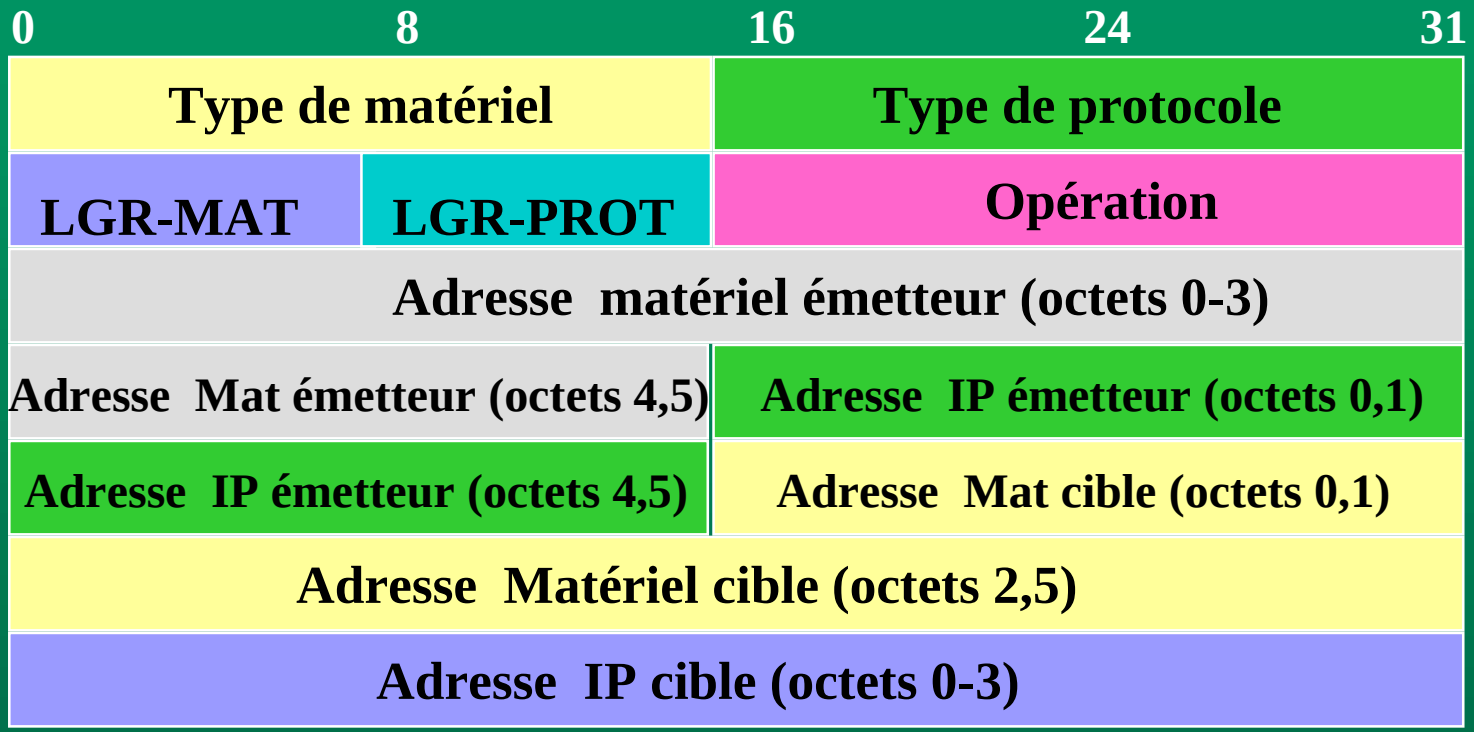
- **Pour connaître l'adresse physique de B, PB, à partir de son adresse IP IB, la machine A diffuse une requête ARP qui contient l'adresse IB vers toutes les machines; la machine B répond avec un message ARP qui contient la paire (IB, PB).**

ARP: Address Resolution Protocol

- **Format du message ARP**

- **La requête ARP est véhiculée dans un message protocolaire lui-même encapsulé dans la trame de liaison de données.**
- **Lorsque la trame arrive à destination, la couche liaison de données détermine l'entité responsable du message encapsulé; Ex: champ type de la trame Ethernet: 0806 pour ARP**
- **La structure du message ARP/RARP gère une association adresse de protocole / adresse physique indépendamment de l'interface physique et du protocole utilisé :**

ARP: Address Resolution Protocol



Autre technique : proxy Arp

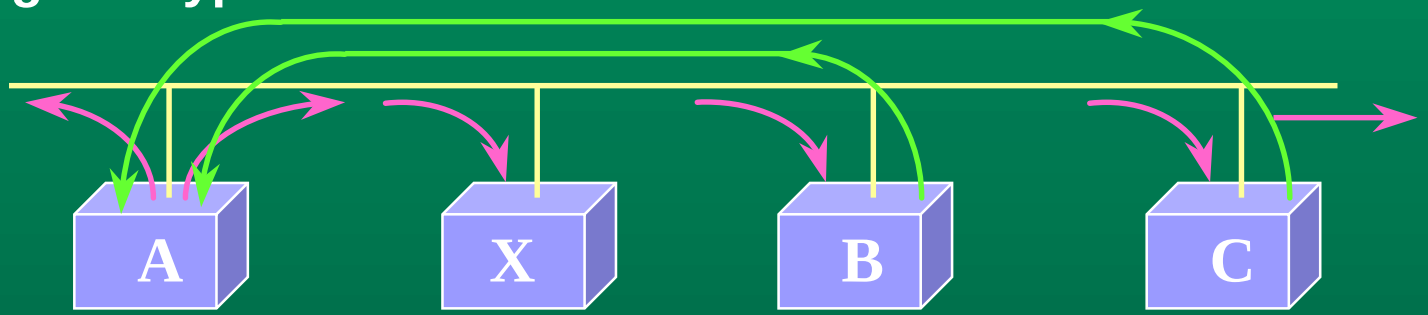
RARP: ReverseAddress Resolution Protocol

- **Le besoin**
  - **L'adresse IP d'une machine est configurable (elle dépend du réseau sur lequel elle se trouve) et est souvent enregistrée sur la mémoire secondaire où le système d'exploitation l'accède au démarrage.**
  - **Ce fonctionnement usuel n'est plus possible dès lors que la machine est une station sans mémoire secondaire.**
- **Problème : déterminer un mécanisme permettant à la station d'obtenir son adresse IP depuis le réseau.**
- **La solution**
  - **Protocole de bas niveau appelé Reverse Adress Resolution Protocol**
  - **Permet d'obtenir son adresse IP à partir de l'adresse physique qui lui est associée.**
- **Fonctionnement**

Serveur RARP sur le réseau physique; son rôle: fournir les adresses IP associées aux adresses physiques des stations du réseau;

RARP: ReverseAddress Resolution Protocol

- Le serveur possède une base de données contenant les couples adresse physique/adresse IP,
- les stations émettent une requête RARP sur le réseau, consistant à demander l'adresse IP qui est associée à leur adresse physique,
- Les requêtes RARP sont propagées vers le ou les serveur(s) RARP par mécanisme de diffusion. Le(s) serveur(s) RARP réponde(nt) par un message de type RARP.

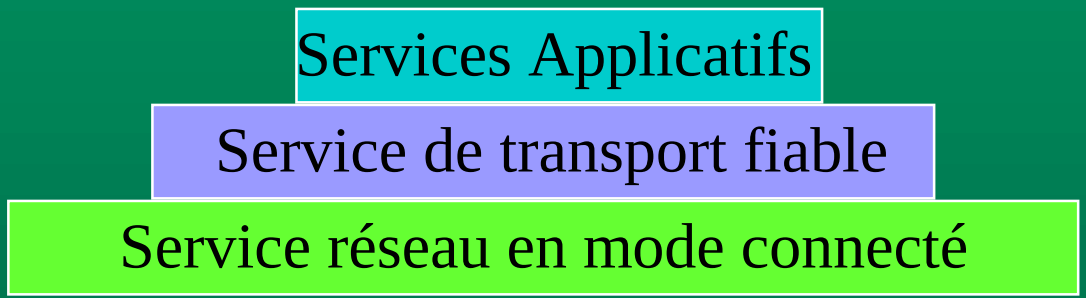


Pour conaître son adresse IP, A diffuse sur le réseau, une requête RARP qui la désigne comme destinataire

Les Serveurs RARP (B et C) répondent à la requête.

IP : Internet Protocol

- **Le protocole Internet (Internet Protocol ou IP) :**
  - **réalise les fonctionnalités de la couche réseau selon le modèle OSI**
  - **se situe au coeur de l'architecture TCP/IP qui met en oeuvre un mode de transport fiable (TCP) sur un service réseau en mode non connecté :**



- **Le service offert par le protocole IP est dit non fiable :**
  - **remise de paquets non garantie,**
  - **sans connexion (paquets traités indépendamment les uns des autres),**
  - **pour le mieux (*best effort*, les paquets ne sont pas éliminés sans raison).**

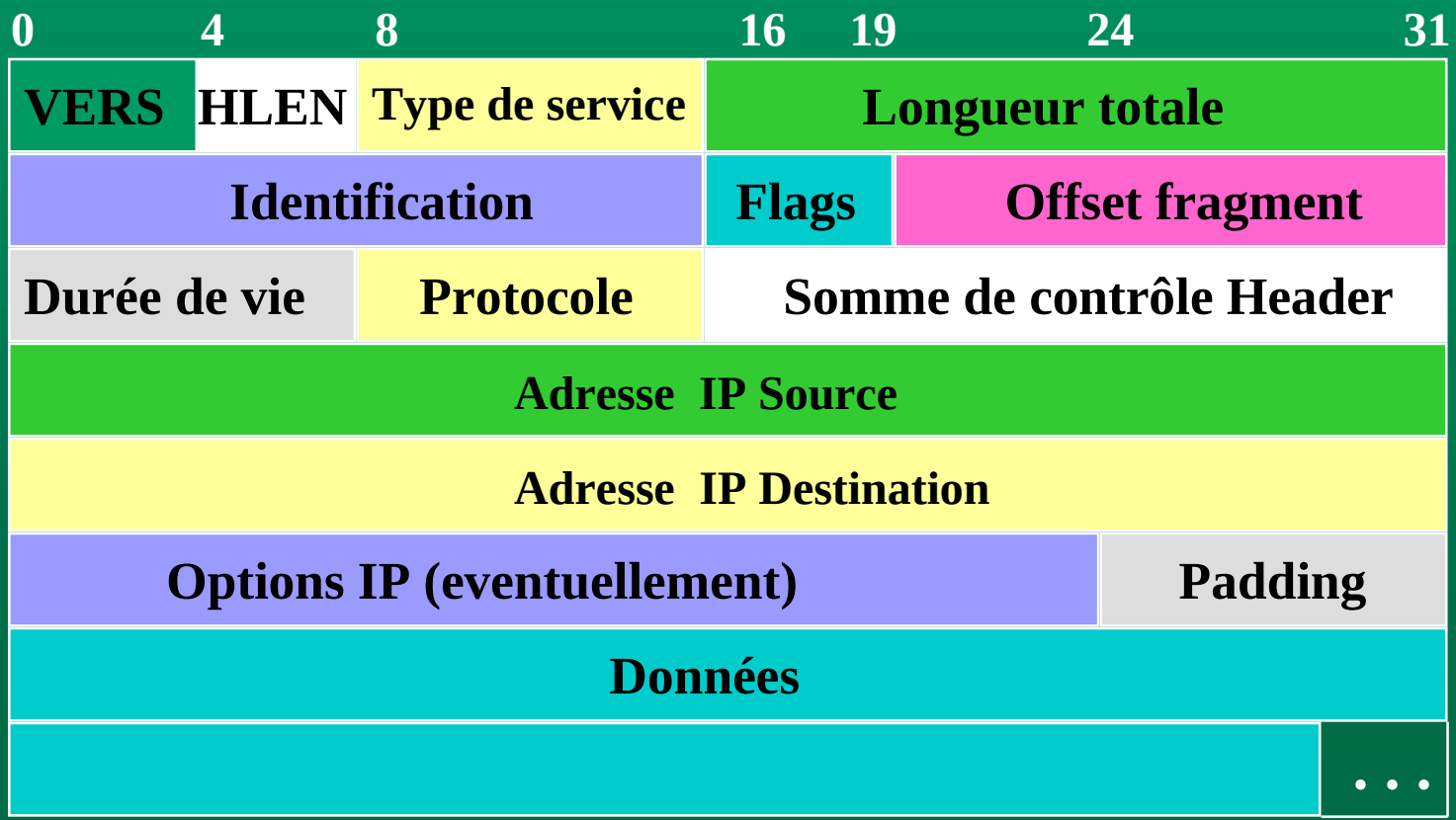
IP : Internet Protocol (suite)

- **Le protocole IP définit :**
  - **l'unité de donnée transférée dans les interconnexions (datagramme),**
  - **la fonction de routage,**
  - **les règles qui mettent en oeuvre la remise de paquets en mode non connecté**

IP : Internet Protocol (le datagramme)

☞ **Le datagramme IP**

L'unité de transfert de base dans un réseau internet est le datagramme qui est constituée d'un en-tête et d'un champ de données:



IP : Internet Protocol (le datagramme)

Signification des champs du datagramme IP :

- **VERS : numéro de version de protocole IP, actuellement version 4,**
- **HLEN : longueur de l'en-tête en mots de 32 bits, généralement égal à 5 (pas d'option),**
- **Longueur totale : longueur totale du datagramme (en-tête + données)**
- **Type de service : indique comment le datagramme doit être géré :**

- **PRECEDENCE (3 bits) : définit la priorité du datagramme; en général ignoré par les machines et passerelles (pb de congestion).**
- **Bits D, T, R : indiquent le type d'acheminement désiré du datagramme, permettant à une passerelle de choisir entre plusieurs routes (si elles existent) : D signifie délai court, T signifie débit élevé et R signifie grande fiabilité.**

IP : Internet Protocol (le datagramme)

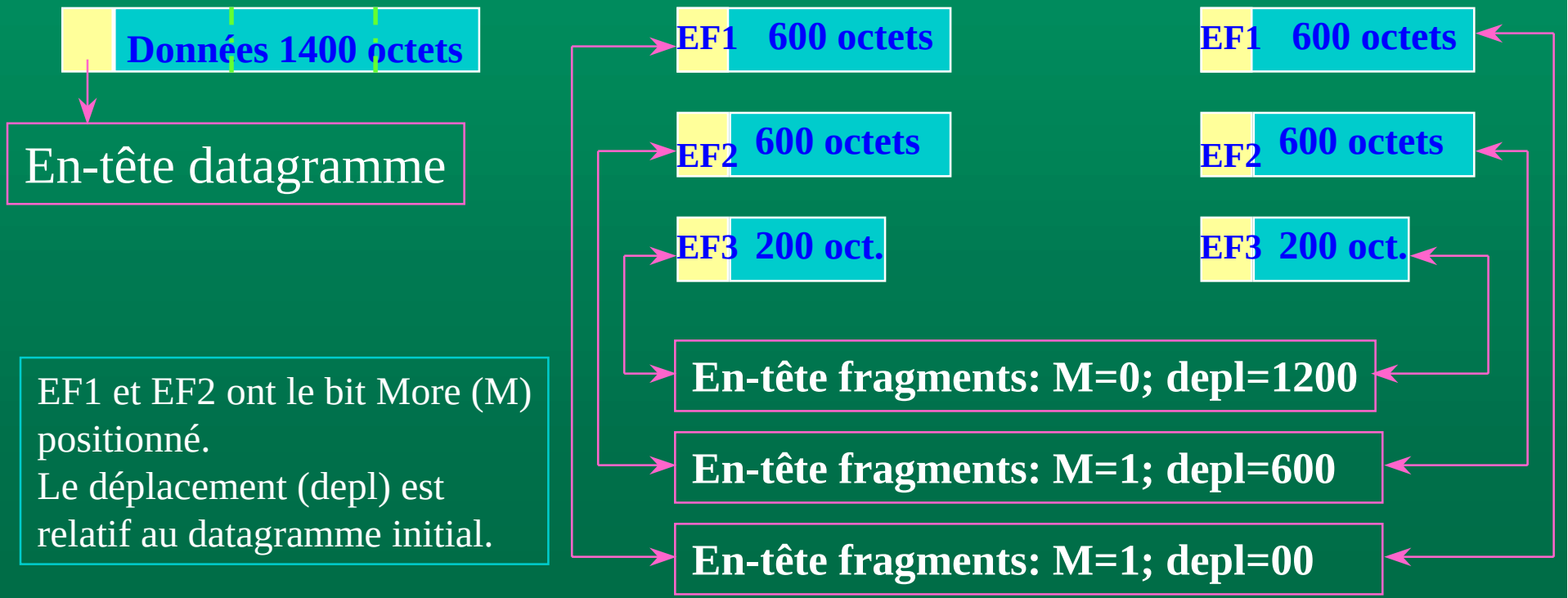
- **FRAGMENT OFFSET, FLAGS, IDENTIFICATION : les champs de la fragmentation.**
  - **Sur toute machine ou passerelle mettant en oeuvre TCP/IP une unité maximale de transfert (*Maximum Transfert Unit* ou MTU) définit la taille maximale d'un datagramme véhiculé sur le réseau physique correspondant**
  - **lorsque le datagramme est routé vers un réseau physique dont le MTU est plus petit que le MTU courant, la passerelle fragmente le datagramme en un certain nombre de fragments, véhiculés par autant de trames sur le réseau physique correspondant,**
  - **lorsque le datagramme est routé vers un réseau physique dont le MTU est supérieur au MTU courant, la passerelle route les fragments tels quels (rappel : les datagrammes peuvent emprunter des chemins différents),**
  - **le destinataire final reconstitue le datagramme initial à partir de l'ensemble des fragments reçus; la taille de ces fragments correspond au plus petit MTU emprunté sur le réseau. Si un seul des fragments est perdu, le datagramme initial est considéré comme perdu : la probabilité de perte d'un datagramme augmente avec la fragmentation.**

IP : Internet Protocol (le datagramme)

- **FRAGMENT OFFSET : indique le déplacement des données contenues dans le fragment par rapport au datagramme initial. C'est un multiple de 8 octets; la taille du fragment est donc également un multiple de 8 octets.**

- **chaque fragment a une structure identique à celle du datagramme initial, seul les champs FLAGS et FRAGMENT OFFSET sont spécifiques.**

IP : Internet Protocol (le datagramme)



EF1 et EF2 ont le bit More (M) positionné.
Le déplacement (depl) est relatif au datagramme initial.

IP : Internet Protocol (le datagramme)

- **Longueur totale : taille du fragment et non pas celle du datagramme initial, à partir du dernier fragment (TOTAL LENGTH, FRAGMENT OFFSET et FLAGS) on peut déterminer la taille du datagramme initial.**
- **IDENTIFICATION : entier qui identifie le datagramme initial (utilisé pour la reconstitution à partir des fragments qui ont tous la même valeur).**
- **FLAGS contient un bit appelé "*do not fragment*" (01X)**
- **un autre bit appelé "*More fragments*" (FLAGS = 001 signifie d'autres fragments à suivre) permet au destinataire final de reconstituer le datagramme initial en identifiant les différents fragments (milieu ou fin du datagramme initial)**
- **les passerelles doivent accepter des datagrammes dont la taille maximale correspond à celle du MTU le plus grand, des réseaux auxquels elle est connectée.**
- **les passerelles doivent accepter sans les fragmenter, les datagrammes de longueur 576 octets.**

IP : Internet Protocol (le datagramme)

- **Durée de vie**
  - **Ce champ indique en secondes, la durée maximale de transit du datagramme sur l'internet. La machine qui émet le datagramme définit sa durée de vie.**
  - **Les passerelles qui traitent le datagramme doivent décrémenter sa durée de vie du nombre de secondes (1 au minimum) que le datagramme a passé pendant son séjour dans la passerelle; lorsque celle-ci expire le datagramme est détruit et un message d'erreur est renvoyé à l'émetteur.**
- **Protocole**

Ce champ identifie le protocole de niveau supérieur dont le message est véhiculé dans le champ données du datagramme :

- **6 : TCP,**
- **17 : UDP,**
- **1 : ICMP.**

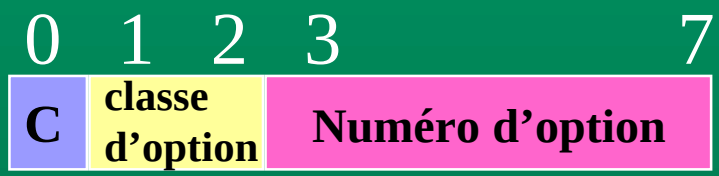
IP : Internet Protocol (le datagramme)

- **Somme de contrôle de l'en-tête**
  - **Ce champ permet de détecter les erreurs survenant dans l'en-tête du datagramme, et par conséquent l'intégrité du datagramme.**
  - **Le total de contrôle d'IP porte sur l'en-tête du datagramme et non sur les données véhiculées. Lors du calcul, le champ HEADER CHECKSUM est supposé contenir la valeur 0 :**

xxxx xxxx xxxx xxxx	**(VERS, HLEN, TYPE OF SERVICE)**
xxxx xxxx xxxx xxxx	**(TOTAL LENGTH)**
xxxx xxxx xxxx xxxx	**(ID. FLAGS, FRAGMENT OFFSET)**
xxxx xxxx xxxx xxxx	**(TIME TO LIVE, PROTOCOL)**
0000 0000 0000 0000	**(HEADER CHECKSUM)**
xxxx xxxx xxxx xxxx	**(IP SOURCE)**
xxxx xxxx xxxx xxxx	**(IP SOURCE)**
xxxx xxxx xxxx xxxx	**(IP DESTINATION)**
xxxx xxxx xxxx xxxx	**(IP DESTINATION)**
...	**(OPTIONS éventuelles + PADDING**

IP : Internet Protocol (le datagramme)

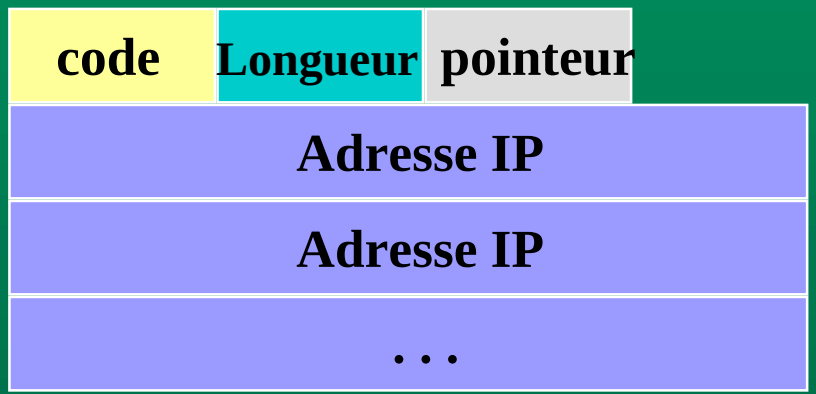
- **OPTIONS**
  - **Le champ OPTIONS est facultatif et de longueur variable. Les options concernent essentiellement des fonctionnalités de mise au point. Une option est définie par un champ octet :**



  - **copie (C) indique que l'option doit être recopiée dans tous les fragments (c=1) ou bien uniquement dans le premier fragment (c=0).**
  - **les bits classe d'option et numéro d'option indiquent le type de l'option et une option particulière de ce type :**

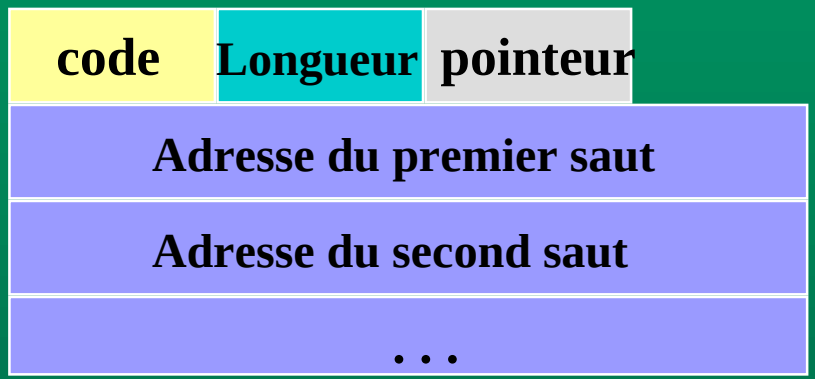
IP : Internet Protocol (le datagramme)

- **<u>Enregistrement de route</u> (classe = 0, option = 7) : permet à la source de créer une liste d'adresse IP vide et de demander à chaque passerelle d'ajouter son adresse dans la liste.**



IP : Internet Protocol (le datagramme)

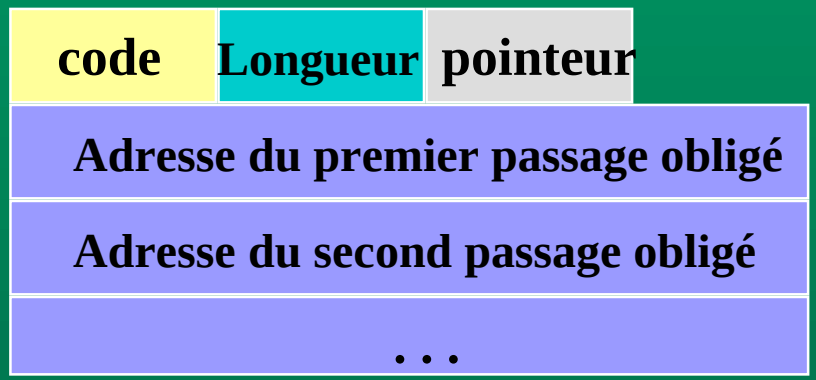
- **<u>Routage strict prédéfini par l'émetteur</u> (classe = 0, option = 9): prédéfinit le routage qui doit être utilisé dans l'interconnexion en indiquant la suite des adresses IP dans l'option :**



  - **Le chemin spécifié ne tolère aucun autre intermédiaire; une erreur est retournée à l'émetteur si une passerelle ne peut appliquer le routage spécifié.**
  - **Les passerelles enregistrent successivement leur adresse à l'emplacement indiqué par le champ *pointeur*.**

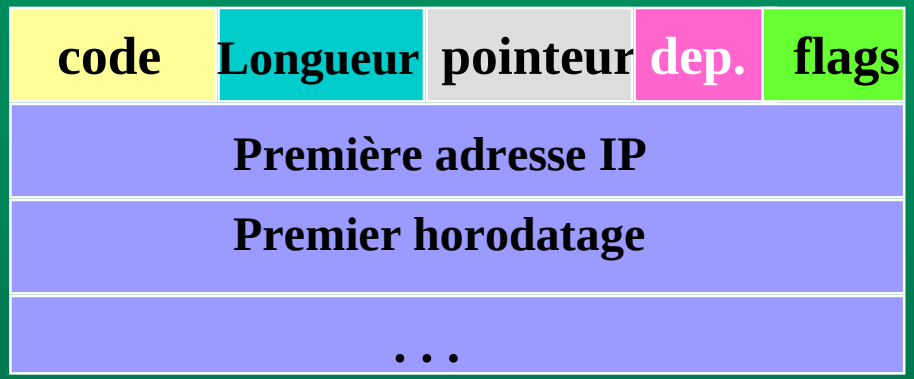
IP : Internet Protocol (le datagramme)

- **Routage lâche prédéfini par l'émetteur (classe = 0, option = 3): Cette option autorise, entre deux passages obligés, le transit par d'autres intermédiaires :**



IP : Internet Protocol (le datagramme)

- **Horodatage (classe = 2, option = 4) : cette option permet d'obtenir les temps de passage (*timestamp*) des datagrammes dans les passerelles. Exprimé en heure et date universelle.**



- **Une liste de couples (adresse IP - horodatage) est réservée par l'émetteur; les passerelles ont à charge de remplir un champ lors du passage du datagramme.**

IP : Internet Protocol (le datagramme)

- **Le champ dépassement de capacité (dep.) comptabilise les passerelles qui n'ont pu fournir les informations requises (liste initiale était trop petite).**

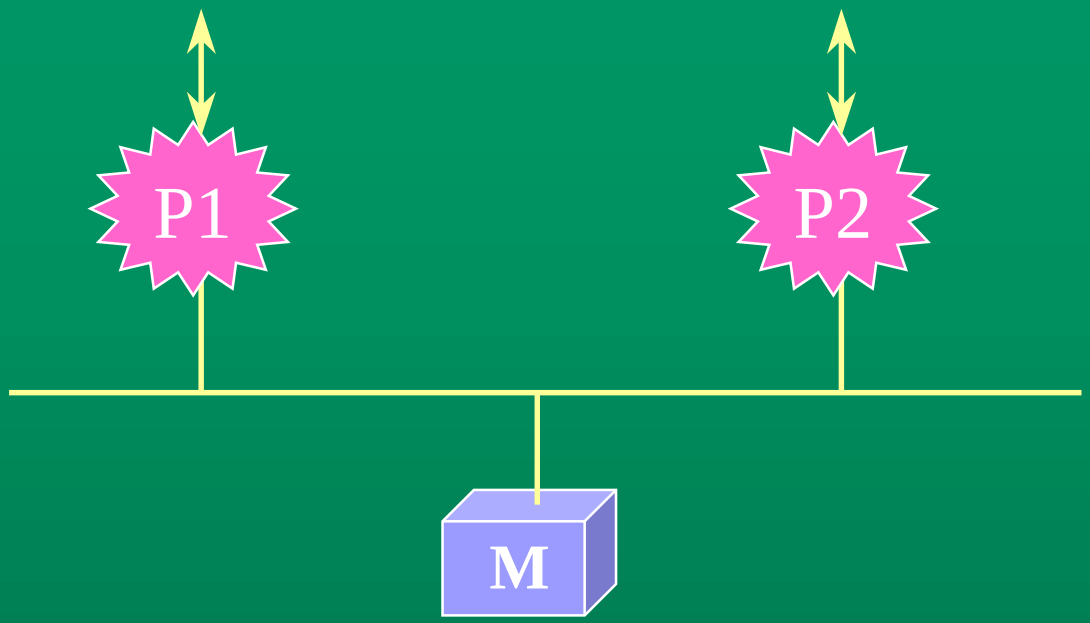
- **Le champ FLAGS indique si les passerelles doivent renseigner uniquement l'horodatage (FLAGS = 0), ou bien l'horodatage et l'adresse IP (FLAGS=1). Si les adresses IP sont prédéfinies par l'émetteur (FLAGS=3), les passerelles n'indiquent l'horodatage que si l'adresse IP pointée par le champ *pointeur* est identique à leur adresse IP.**

- **Les horodatages, bien qu'exprimés en temps universel, ne constituent qu'une estimation sur le temps de passage car les horloges des machines situées sur les réseaux ne sont pas synchronisées.**

Routage des datagrammes

- Le routage est le processus permettant à un datagramme d'être acheminé vers le destinataire lorsque celui-ci n'est pas sur le même réseau physique que l'émetteur.
- Le chemin parcouru est le résultat du processus de routage qui effectue les choix nécessaires afin d'acheminer le datagramme.
- Les routeurs forment une structure coopérative de telle manière qu'un datagramme transite de passerelle en passerelle jusqu'à ce que l'une d'entre elles le délivre à son destinataire. Un routeur possède deux ou plusieurs connexions réseaux tandis qu'une machine possède généralement qu'une seule connexion.
- Machines et routeurs participent au routage :
  - les machines doivent déterminer si le datagramme doit être délivré sur le réseau physique sur lequel elles sont connectées (routage direct) ou bien si le datagramme doit être acheminé vers une passerelle; dans ce cas (routage indirect), elle doit identifier la passerelle appropriée.
  - les passerelles effectuent le choix de routage vers d'autres passerelles afin d'acheminer le datagramme vers sa destination finale.

Routage des datagrammes (suite)



M est mono-domiciliée et doit acheminer les datagrammes vers une des passerelles P1 ou P2; elle effectue donc le premier routage. Dans cette situation, aucune solution n'offre un meilleur choix.

Le routage indirect repose sur une table de routage IP, présente sur toute machine et passerelle, indiquant la manière d'atteindre un ensemble de destinations.

Routage des datagrammes (suite)

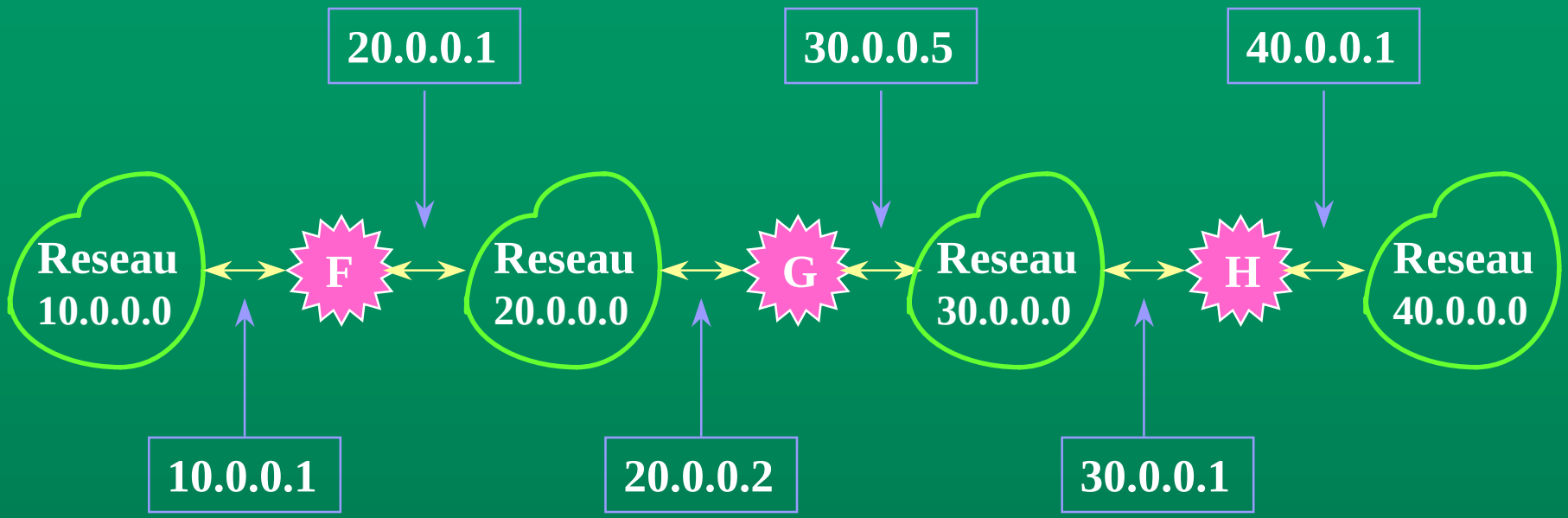
- **Les tables de routage IP, pour des raisons évidentes d'encombrement, renseignent seulement les adresses réseaux et non pas les adresses machines.**

- **Typiquement, une table de routage contient des couples (R, P) où R est l'adresse IP d'un réseau destination et P est l'adresse IP de la passerelle correspondant au prochain saut dans le cheminement vers le réseau destinataire.**

- **La passerelle ne connaît pas le chemin complet pour atteindre la destination.**

- **Pour une table de routage contenant des couples (R, P) et appartenant à la machine M, P et M sont connectés sur le même réseau physique dont l'adresse de niveau réseau (partie Netid de l'adresse IP) est R.**

Routage des datagrammes (suite)



Pour atteindre les machines du réseau	10.0.0.0	20.0.0.0	30.0.0.0	40.0.0.0
Router vers	20.0.0.1	direct	direct	30.0.0.1

Table de routage de G

Routage des datagrammes (suite)

Route_Datagramme_IP(datagramme, table_de_routage)

- **Extraire l'adresse IP destination, ID, du datagramme,**
- **Calculer l'adresse du réseau destination, IN.**
- **Si IN correspondant à une adresse de réseau directement accessible,**
  envoyer le datagramme vers sa destination, sur ce réseau.
- **sinon si dans la tablede routage, il existe une route vers ID**
  router le datagramme selon les informations contenues dans la table de routage.
- **sinon si IN apparaît dans la table de routage,**
  router le datagramme selon les informations contenues dans la table de routage.
- **sinon s'il existe une route par défaut**
  router le datagramme vers la passerelle par défaut.
- **sinon déclarer une erreur de routage.**

Routage des datagrammes (suite)

- **Après exécution de l'algorithme de routage, IP transmet le datagramme ainsi que l'adresse IP determinée, à l'interface réseau vers lequel le datagramme doit être acheminé.**

- **L'interface physique détermine alors l'adresse physique associée à l'adresse IP et achemine le datagramme sans l'avoir modifié (l'adresse IP du prochain saut n'est sauvegardée nulle part).**

- **Si le datagramme est acheminé vers une autre passerelle, il est à nouveau géré de la même manière, et ainsi de suite jusqu'à sa destination finale.**

Routage des datagrammes (suite)

- **Les datagrammes entrants sont traités différemment selon qu'il sont reçus par une machine ou une passerelle :**

- **machine : le logiciel IP examine l'adresse destination à l'intérieur du datagramme**

  - **si cette adresse IP est identique à celle de la machine, IP accepte le datagramme et transmet son contenu à la couche supérieure.**
  - **sinon, le datagramme est rejeté; une machine recevant un datagramme destiné à une autre machine ne doit pas router le datagramme.**

- **passerelle : IP détermine si le datagramme est arrivé à destination et dans ce cas le délivre à la couche supérieure. Si le datagramme n'a pas atteint sa destination finale, il est routé selon l'algorithme de routage précédemment décrit.**

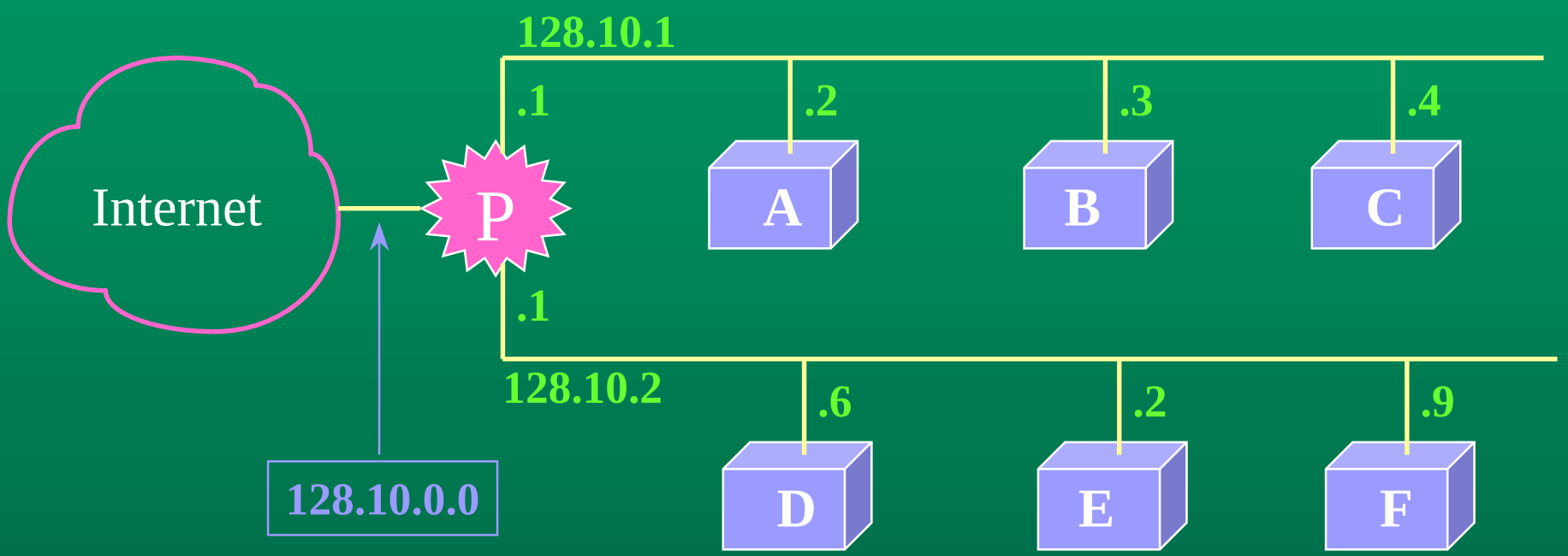
Le sous-adressage

- **Le sous-adressage est une extension du plan d'adressage initial**
- **Devant la croissance du nombre de réseaux de l'Internet, il a été introduit afin de limiter la consommation d'adresses IP qui permet également de diminuer :**
  - **la gestion administrative des adresses IP,**
  - **la taille des tables de routage des passerelles,**
  - **la taille des informations de routage,**
  - **le traitement effectué au niveau des passerelles.**

- **Principes**
  - **A l'intérieur d'une entité associée à une adresse IP de classe A, B ou C, plusieurs réseaux physiques partagent cette adresse IP.**
  - **On dit alors que ces réseaux physiques sont des sous-réseaux (*subnet*) du réseau d'adresse IP.**

Le sous-adressage (suite)

Les sous-réseaux 128.10.1.0 et 128.10.2.0 sont notés seulement avec le NetId, les machines seulement avec le Hostid ; exemple IP(F) = 128.10.2.9



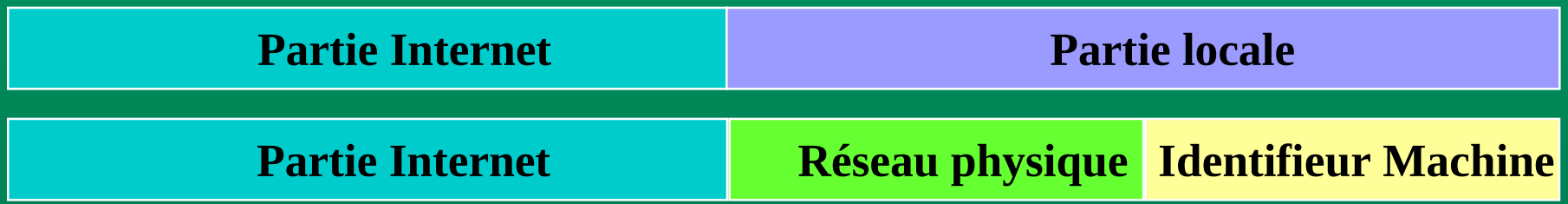
Un site avec deux réseaux physiques utilisant le sous-adressage de manière à ce que ses deux sous-réseaux soient couverts par une seule adresse IP de classe B.
La passerelle P accepte tout le trafic destiné au réseau 128.10.0.0 et sélectionne le sous-réseau en fonction du troisième octet de l'adresse destination.

Le sous-adressage (suite)

- **Le site utilise une seule adresse pour les deux réseaux physiques.**
- **A l'exception de P, toute passerelle de l'internet route comme s'il n'existait qu'un seul réseau.**
- **La passerelle doit router vers l'un ou l'autre des sous-réseaux ; le découpage du site en sous-réseaux a été effectué sur la base du troisième octet de l'adresse :**
  - **les adresses des machines du premier sous-réseau sont de la forme 128.10.1.X,**
  - **les adresses des machines du second sous-réseau sont de la forme 128.10.2.X.**
- **Pour sélectionner l'un ou l'autre des sous-réseaux, P examine le troisième octet de l'adresse destination : si la valeur est 1, le datagramme est routé vers réseau 128.10.1.0, si la valeur est 2, il est routé vers le réseau 128.10.2.0.**

Le sous-adressage (suite)

- **Conceptuellement, la partie locale dans le plan d'adressage initial est subdivisée en "partie réseau physique" + "identification de machine (hostid) sur ce sous-réseau" :**

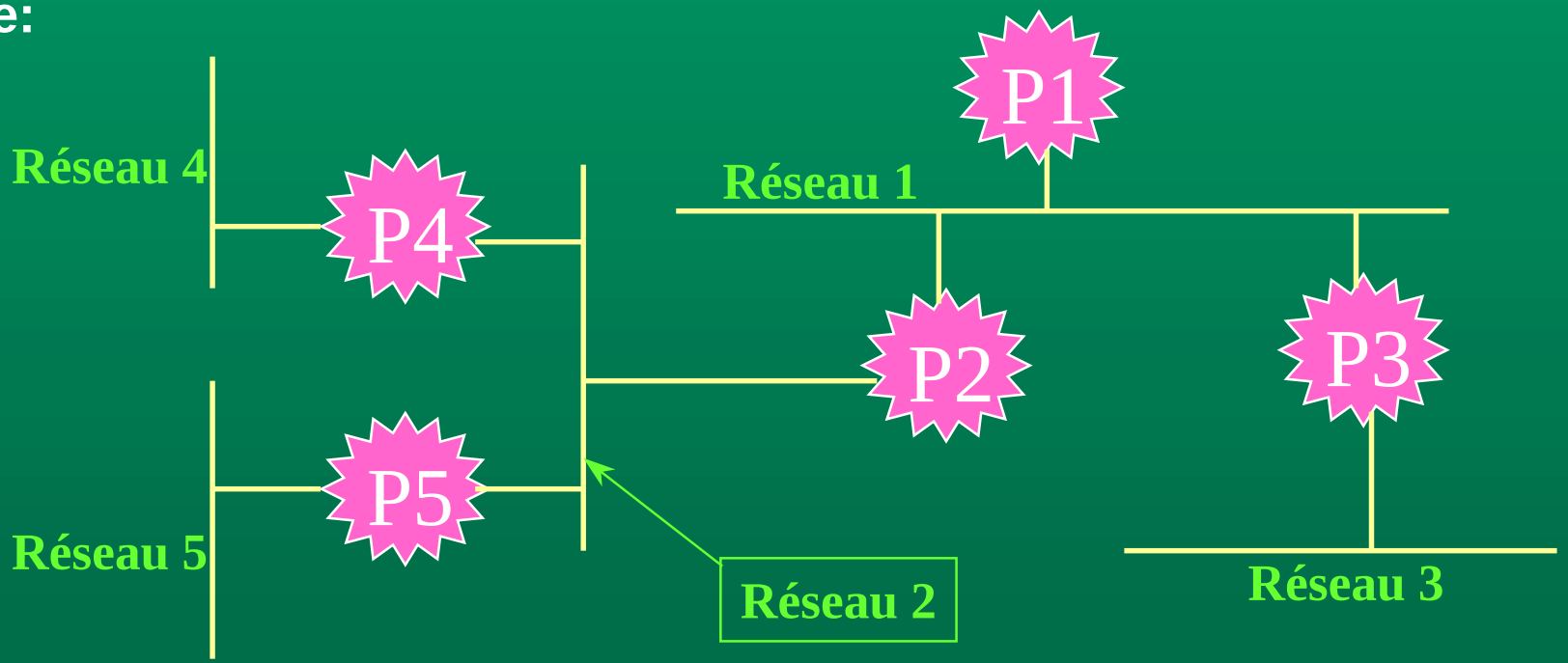


- **«Partie Internet» correspond au NetId (plan d'adressage initial)**
- **«Partie locale» correspond au hostid (plan d'adressage initial)**
- **les champs «Réseau physique» et «identifieur Machine» sont de taille variable; la longueur des 2 champs étant toujours égale à la longueur de la «Partie locale».**

Le sous-adressage (suite)

Structure du sous-adressage

- **Structuration souple : chaque site peut définir lui-même les longueurs des champs réseau physique et identificateur de machine.**
- **Flexibilité indispensable pour adapter la configuration réseau d'un site:**



Ce site a cinq réseaux physiques organisés en trois niveau : le découpage rudimentaire en réseau physique et adresse machine peut ne pas être optimal.

Le sous-adressage (suite)

- **Le choix du découpage dépend des perspectives d'évolution du site:**
  - **Exemple Classe B : 8 bits pour les parties réseau et machine donnent un potentiel de 256 sous-réseaux et 254 machines par sous-réseau, tandis que 3 bits pour la partie réseau et 13 bits pour le champ machine permettent 8 réseaux de 8190 machines chacun.**
  - **Exemple Classe C : 4 bits pour la partie réseau et 4 bits pour le champ machine permettent 16 réseaux de 14 machines chacun.**
- **Lorsque le sous-adressage est ainsi défini, toutes les machines du réseau doivent s'y conformer sous peine de dysfonctionnement du routage ==> configuration rigoureuse.**

Le sous-adressage (suite)

- **Utilisation de masques**
- **Le sous-adressage ==> masque de 32 bits associé au sous-réseau.**
- **Bits du masque de sous-réseau (*subnet mask*) :**
  - **positionnés à 1 : partie réseau,**
  - **positionnés à 0 : partie machine**
- **11111111 11111111 11111111 00000000**

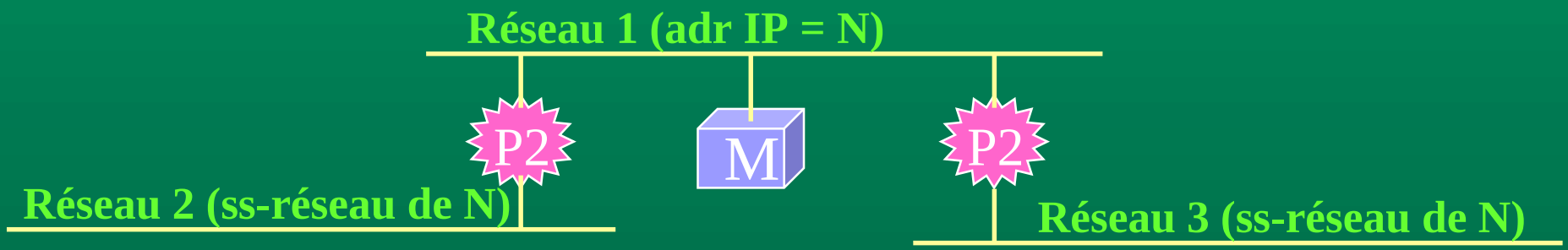
  ==> 3 octets pour le champ réseau, 1 octet pour le champ machine
- **Les bits du masque identifiant sous-réseau et machine peuvent ne pas être contigus : 11111111 11111111 00011000 01000000**
- **Les notations suivantes sont utilisées :**
  - **décimale pointée; exemple : 255.255.255.0**
  - **triplet : { <ident. réseau>, <ident. sous-réseau> <ident. machine> } ; cette notation renseigne les valeurs mais pas les champs de bits;**

    exemple { -1, -1, 0 } , { 128.10, 27, -1 }.
  - **adresse réseau/masque : 193.49.60.0/27 (27=# bits contigüs du masque)**

Le sous-adressage (suite)

Routage avec sous-réseaux

- **Le routage IP initial a été étendu à l'adressage en sous-réseaux;**
- **l'algorithme de routage obtenu doit être présent dans les machines ayant une adresse de sous-réseau, mais également dans les autres machines et passerelles du site qui doivent acheminer les datagrammes vers ces sous-réseaux.**



M doit utiliser le routage de sous-réseaux pour décider si elle route vers les passerelles P1 ou P2 bien qu'elle même soit connectée à un réseau (Réseau 1) n'ayant pas de sous-adressage

Le sous-adressage (suite)

Le routage unifié : Une entrée dans la table de routage =
(masque de sous-réseau, adresse sous-réseau, adresse de la passerelle)

Algorithme de routage unifié :

- Route_IP_Datagram(datagram, routing_table)
- Extraire l'adresse ID de destination du datagramme,
- Calculer l'adresse IN du réseau destination,
- Si IN correspond à une adresse réseau directement accessible
  envoyer le datagramme sur le réseau physique correspondant,
- Sinon
  - Pour chaque entrée dans la table de routage,
    - N = (ID & masque de sous-réseau de l'entrée)
    - Si N est égal au champ adresse réseau de l'entrée
      router le datagramme vers la passerelle correspondante,
  - Fin_Pour
- Si aucune entrée ne correspond, déclarer une erreur de routage.

Le sous-adressage (suite)

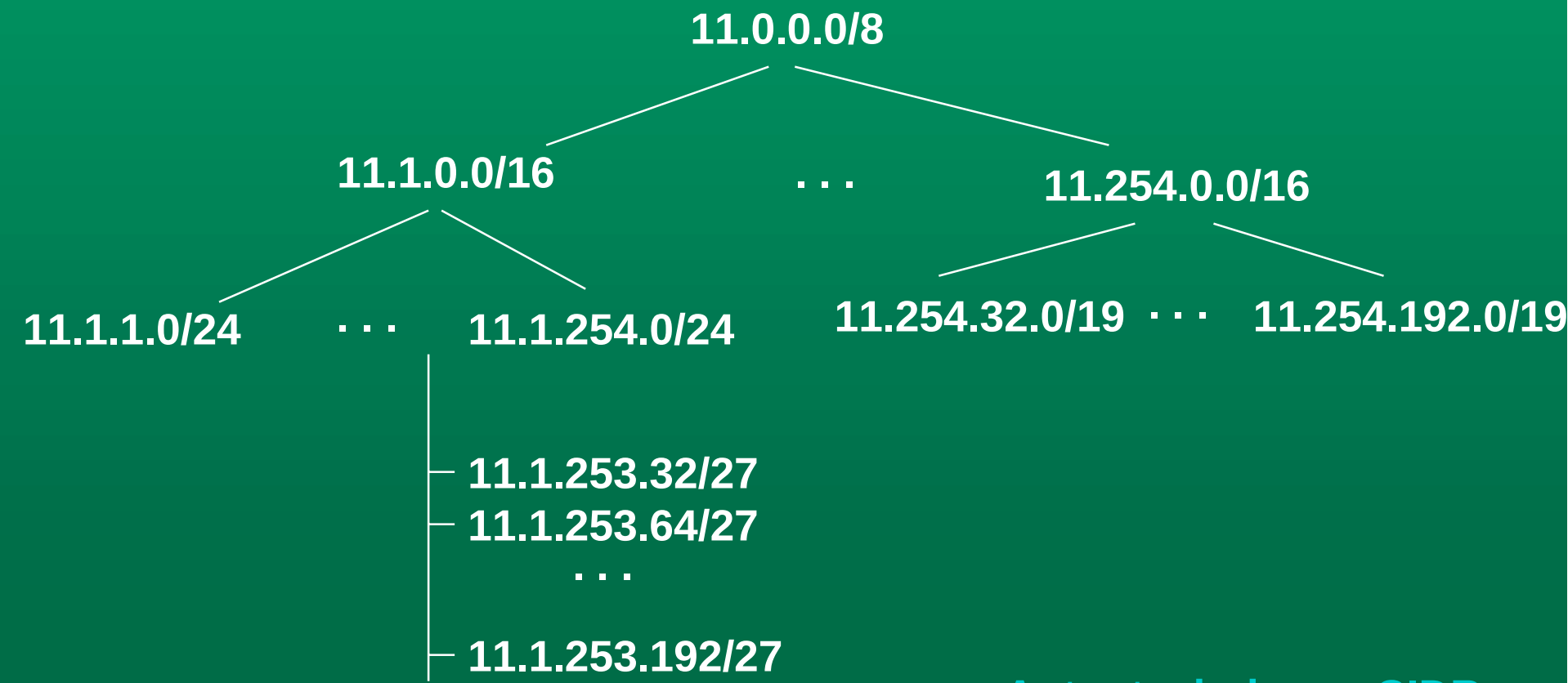
- **Diffusion sur les sous-réseaux**
- **Elle est plus complexe que dans le plan d'adressage initial.**
- **Dans le plan d'adressage Internet initial, Hostid = 11..1, ==> diffusion vers toutes les machines du réseau.**
- **D'un point de vue extérieur à un site doté de sous-réseaux, la diffusion n'a de sens que si la passerelle qui connaît les sous-réseaux propage la diffusion à tous ses réseaux physiques : { réseau, -1, -1 }.**
- **Depuis un ensemble de sous-réseau, il est possible d'émettre une diffusion sur un sous-réseau particulier : { réseau, sous-réseau, -1 }.**

Le sous-adressage variable (VLSM)

- **RFC 1009 : un réseau IP peut posséder plusieurs masques différents; ==> réseau de type VLSM (Variable Length Subnet Masks)**
- **Evite la rigidité du masque fixe qui impose :**
  - **le nombre de sous-réseaux**
  - **le nombre de machines par sous-réseau**
  - **Exemple : 130.5.0.0/22 ==> 64 sous-reseaux et 1022 machines / sous-réseau**
    - **inadapté pour des petits sous-réseaux de quelques machines; exemple 30 machines sur un sous-réseau ==> 992 adresses IP perdues**
- **Permet l'adaptation de l'adressage IP a la taille des sous-réseaux**
  - **Exemple précédent : cohabitation de grands et petits sous-réseaux**
    - **130.5.0.0/22 (64 sous-reseaux et 1022 machines / sous-réseau)**
    - **130.5.0.0/26 (1024 sous-réseaux de 62 machines / sous-réseau)**

VLSM : agregation de routes

- **Division d'un espace IP en sous-réseaux successifs**
- **Permet de masquer les informations de routage entre groupes de sous-reseaux**



Autre technique : CIDR

Le Protocole ICMP

Le besoin

- **Le protocole ICMP (Internet Control Message Protocol) permet d'envoyer des messages de contrôle ou d'erreur vers d'autres machines ou passerelles.**
- **ICMP rapporte les messages d'erreur à l'émetteur initial.**
- **Beaucoup d'erreurs sont causées par l'émetteur, mais d'autres sont dûes à des problèmes d'interconnexions rencontrées sur l'Internet :**
  - **machine destination déconnectée,**
  - **durée de vie du datagramme expirée,**
  - **congestion de passerelles intermédiaires.**
- **Si une passerelle détecte un problème sur un datagramme IP, elle le détruit et émet un message ICMP pour informer l'émetteur initial.**
- **Les messages ICMP sont véhiculés à l'intérieur de datagrammes IP et sont routés comme n'importe quel datagramme IP sur l'internet.**
- **Une erreur engendrée par un message ICMP ne peut donner naissance à un autre message ICMP (évite l'effet cummulatif).**

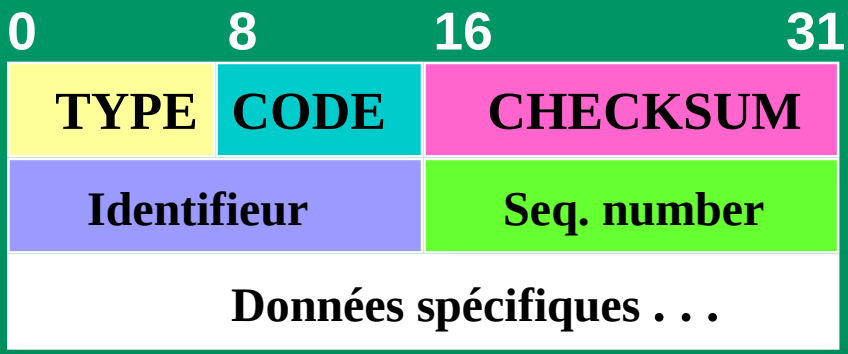
ICMP : format des messages

TYPE	**8 bits; type de message**
CODE	**8 bits; informations complémentaires**
CHECKSUM	**16 bits; champ de contrôle**
HEAD-DATA	**en-tête datagramme + 64 premiers bits des données.**

TYPE	Message ICMP
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect (change a route)
8	Echo Request
11	Time Exceeded (TTL)
12	Parameter Problem with a Datagram

TYPE	Message ICMP
13	Timestamp Request
14	Timestamp Reply
15	Information Request (obsolete)
16	Information Reply (obsolète)
17	Address Mask Reques
18	Address Mask Reply

ICMP : format des commandes



IDENTIFIER et SEQUENCE NUMBER sont utilisés par l'émetteur pour contrôler les réponses aux requêtes, (CODE = 0).

Demande d'écho et réponse *Request, Echo Reply*) d'écho (*Echo*

- **Permettent à une machine ou passerelle de déterminer la validité d'un chemin sur le réseau.**
- **Le champ de données spécifiques est composé de données optionnelles de longueur variable émises par la requête d'écho et devant être renvoyées par le destinataire si présentes.**
- **Utilisé par les outils applicatifs tels ping et traceroute.**

ICMP : les commandes

Synchronisation des Horloges et temps de transit

- **Les horloges de deux machines qui diffèrent de manière importante peuvent poser des problèmes pour des logiciels distribués.**
- **Une machine peut émettre une demande d'horodatage (*timestamp request*) à une autre machine susceptible de lui répondre (*timestamp reply*) en donnant l'heure d'arrivée de la demande et l'heure de départ de la réponse.**
- **L'émetteur peut alors estimer le temps de transit ainsi que la différence entre les horloges locale et distante.**
- **Le champ de données spécifiques comprend l'heure originale (*originate timestamp*) émis par le demandeur, l'heure de réception (*receive timestamp*) du destinataire, et l'heure de départ (*transmit timestamp*) de la réponse.**

ICMP : les commandes

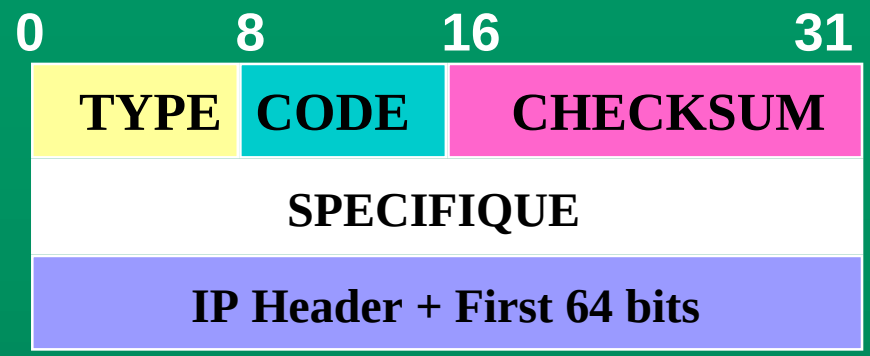
Demande et réponse d'information (*Information Request* + *Reply*)

- **Ces messages étaient initialement utilisés pour permettre aux machines de connaître leur adresse IP au démarrage du système.**
- **Ces commandes sont aujourd'hui remplacées par les protocoles RARP et BOOTP.**

Obtention de masque de sous-réseau

- **Une machine peut émettre une demande de masque de sous-réseau (*Subnet Mask Request*) vers une passerelle gérant le sous-réseau en question.**
- **La passerelle transmet par une "*Subnet Mask Reply*", l'adresse de masque de sous-réseau (de longueur 32 bits) dans le champ de donnée spécifique.**

ICMP : les messages d'erreur



Format des messages d'erreur ICMP

- **CODE indique le codage de l'erreur rapportée et est spécifique à chaque type d'erreur,**
- **SPECIFIQUE est un champ de données spécifique au type d'erreur,**
- **IP HEADER + FIRST 64 bits contient l'en-tête IP + les premiers 64 bits de données du datagramme pour lequel le message est émis.**
- **Compte rendu de destination inaccessible**

ICMP : les messages d'erreur

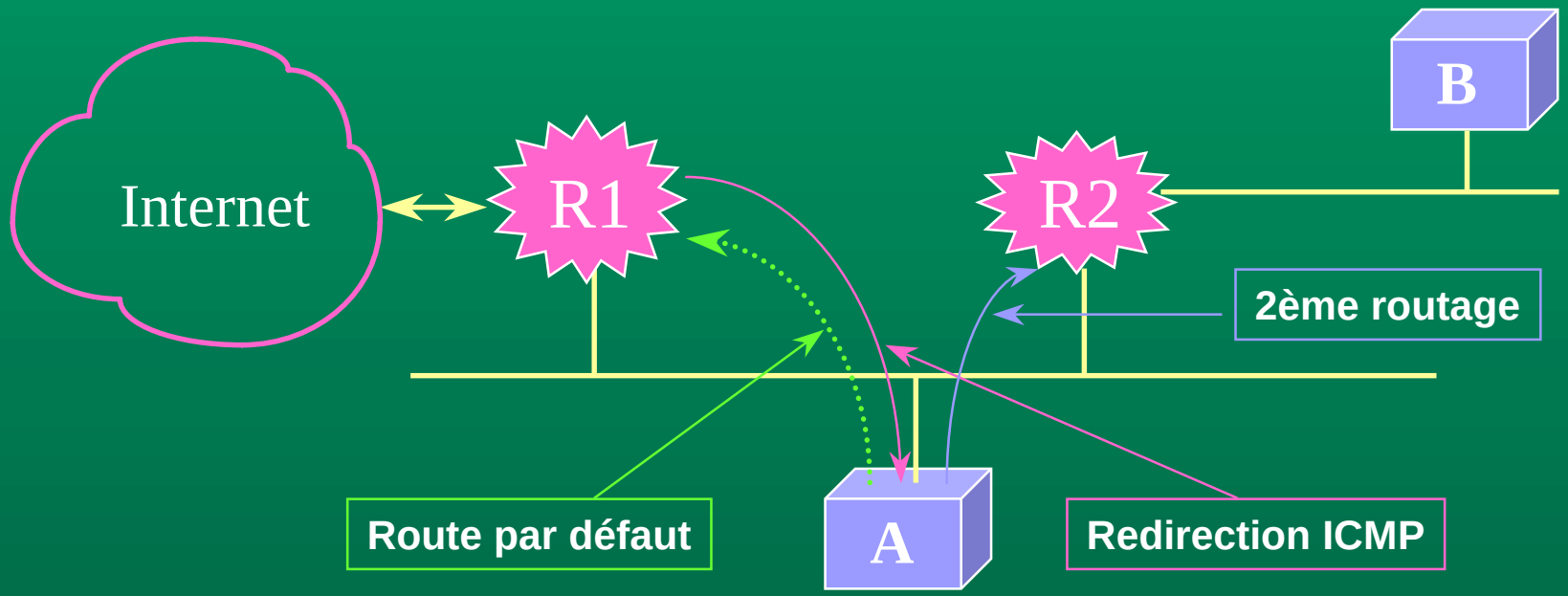
- **Lorsqu'une passerelle émet un message ICMP de type destination inaccessible, le champ code décrit la nature de l'erreur :**
  - **0 Network Unreachable**
  - **1 Host Unreachable**
  - **2 Protocol Unreachable**
  - **3 Port Unreachable**
  - **4 Fragmentation Needed and DF set**
  - **5 Source Route Failed**
  - **6 Destination Network Unknown**
  - **7 Destination Host Unknown**
  - **8 Source Host Isolated**
  - **9 Communication with desination network administratively prohibited**
  - **10 Communication with desination host administratively prohibited**
  - **11 Network Unreachable for type of Service**
  - **12 Host Unreachable for type of Service**

ICMP : contrôle de congestion

- **Le protocole IP étant un protocole en mode non connecté :**
  - **=> les passerelles ne peuvent réserver à l'avance la quantité de mémoire nécessaire au routage des datagrammes.**
  - **=> des datagrammes sont alors détruits.**
- **Cette situation de congestion se produit :**
  - **lorsqu'une passerelle est connectée à deux réseaux aux débits différents (elle ne peut écouler au rythme imposé par le réseau le plus rapide),**
  - **lorsque de nombreuses machines émettent simultanément des datagrammes vers une passerelle.**
- **Pour palier ce problème, la machine peut émettre un message ICMP de limitation de débit de la source (*Source Quench*) vers l'émetteur.**
- **Il n'existe pas de message d'annulation de limitation de débit. La source diminue le débit, puis l'augmente progressivement tant qu'elle ne reçoit pas de nouvelle demande de limitation.**

ICMP : modification de route

Un message ICMP de redirection de route peut être transmis par une passerelle vers une machine reliée au même réseau pour lui signaler que la route n'est pas optimale.



Une fois la redirection effectuée, les datagrammes seront acheminés vers la passerelle appropriée.

ICMP : modification de route

- **Dans le bloc de commande, le champ SPECIFIQUE indique l'adresse de la passerelle que la machine doit utiliser pour router le datagramme; CODE spécifie la redirection :**

CODE	SIGNIFICATION
0	Redirect datagrams for the Network
1	Redirect datagrams for the Host
2	Redirect datagrams for the Type of Service and Network
3	Redirect datagrams for the Type of Service and Host

Detection de routes circulaires ou excessivement longues

- **Une passerelle détruit les datagrammes dont le champ durée de vie est à zéro et émet un message ICMP de délai dépassé.**

CODE	SIGNIFICATION
0	time to live exceeded in transit
1	fragment reassembly time exceeded

ICMP : autres compte-rendus

- **Lorsqu'une passerelle ou une machine détecte un problème avec un datagramme (en-tête incorrecte) non couvert par les messages ICMP prédéfinis, elle émet un message "*Parameter Problem on a Datagram*" vers l'émetteur du datagramme.**
- **Le problème rencontré consiste soit en une option manquante (dans le datagramme), soit en une donnée erronée.**
- **Dans le bloc de commande, le champ CODE indique la nature du pb:**

CODE	SIGNIFICATION
0	erreonous data
1	missing option

- **Le champ spécifique comprend un pointeur (codé sur les 8 premiers bits, les 24 restants étant à 0) servant à identifier l'octet erroné dans le datagramme; il est non significatif lorsque CODE = 1.**

UDP : User Datagram Protocol

- **UDP : protocole de transport sans connexion de service applicatif :**
  - **émission de messages applicatifs : sans établissement de connexion au préalable**
  - **l'arrivée des messages ainsi que l'ordonnancement ne sont pas garantis.**
- **Identification du service : les ports**
  - **les adresses IP désignent les machines entre lesquelles les communications sont établies. Lorsqu'un processus désire entrer en communication avec un autre processus, il doit adresser le processus s'exécutant cette machine.**
  - **L'adressage de ce processus est effectué selon un concept abstrait indépendant du système d'exploitation des machines car :**
    - **les processus sont créés et détruits dynamiquement sur les machines,**
    - **il faut pouvoir remplacer un processus par un autre (exemple reboot) sans que l'application distante ne s'en aperçoive,**
    - **il faut identifier les destinations selon les services offerts, sans connaître les processus qui les mettent en oeuvre,**
    - **un processus doit pouvoir assurer plusieurs services.**

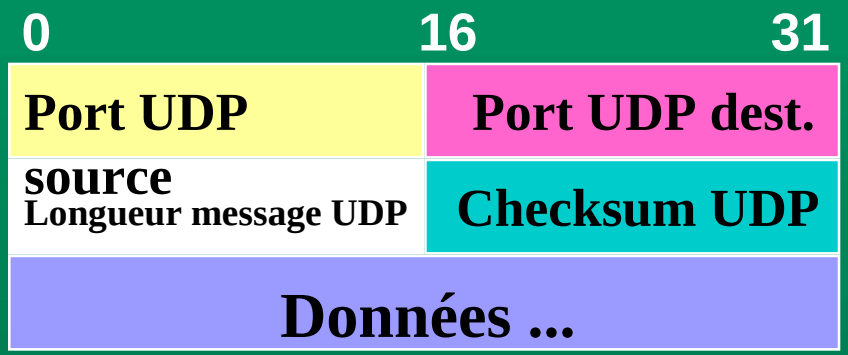
UDP : les ports

- **Ces destinations abstraites permettant d'adresser un service applicatif s'appellent des ports de protocole.**
- **L'émission d'un message se fait sur la base d'un port source et un port destinataire.**
- **Les processus disposent d'une interface système leur permettant de spécifier un port ou d'y accéder (socket, TLI, ...).**
- **Les accès aux ports sont généralement synchrones, les opérations sur les ports sont tamponnés (files d'attente).**

UDP : format des messages

Les messages UDP sont également appelés des datagrammes UDP.

Ils contiennent deux parties : un en-tête UDP et les données UDP.



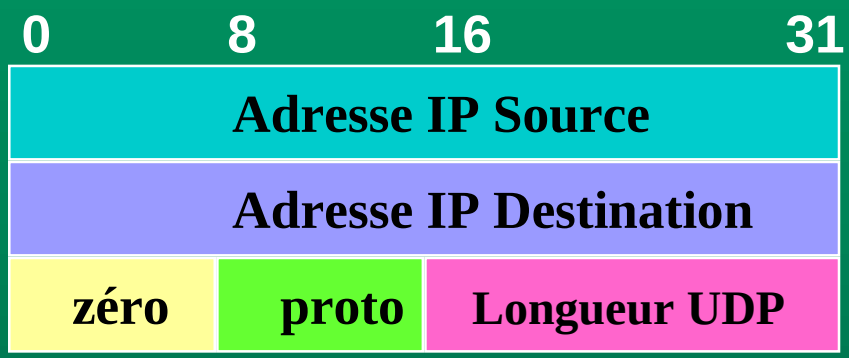
Format des messages UDP

Les ports source et destination contiennent les numéros de port utilisés par UDP pour démultiplexer les datagrammes destinés aux processus en attente de les recevoir. Le port source est facultatif (égal à zéro si non utilisé).

La longueur du message est exprimée en octets (8 au minimum) (en-tête + données), le champ de contrôle est optionnel (0 si non utilisé).

UDP : pseudo en-tête

- **Lorsqu'il est utilisé, le champ de contrôle couvre plus d'informations que celles contenue dans le datagramme UDP; En effet, le checksum est calculé avec un pseudo-en-tête non transmis dans le datagramme:**



Format du pseudo en-tête

Le champ PROTO indique l'identificateur de protocole pour IP (17= UDP)

Le champ LONGUEUR UPD spécifie la longueur du datagramme UPD sans le pseudo-en-tête.

UDP : Multiplexage

- **UDP multiplexe et démultiplexe les datagrammes en sélectionnant les numéros de ports :**
  - **une application obtient un numéro de port de la machine locale; dès lors que l'application émet un message via ce port, le champ PORT SOURCE du datagramme UDP contient ce numéro de port,**
  - **une application connaît (ou obtient) un numéro de port distant afin de communiquer avec le service désiré.**
- **Lorsque UDP reçoit un datagramme, il vérifie que celui-ci est un des ports actuellement actifs (associé à une application) et le délivre à l'application responsable (mise en queue)**
- **si ce n'est pas le cas, il émet un message ICMP *port unreachable*, et détruit le datagramme.**

UDP : les ports standards

- Certains ports sont réservés (*well-kown port assignements*) :

No port	Mot-clé	Description
7	ECHO	Echo
11	USERS	Active Users
13	DAYTIME	Daytime
37	TIME	Time
42	NAMESERVER	Host Name Server
53	DOMAIN	Domain Name Server
67	BOOTPS	Boot protocol server
68	BOOTPC	Boot protocol client
69	TFTP	Trivial File transfert protocol
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management prot.

- D'autres numéros de port (non réservés) peuvent être assignés dynamiquement aux applications.

TCP : Transmission Control Protocol

- **transport fiable de la technologie TCP/IP.**
  - **fiabilité = illusion assurée par le service**
  - **transferts tamponés : découpage en segments**
  - **connexions bidirectionnelles et simultanées**
- **service en mode connecté**
- **garantie de non perte de messages ainsi que de l'ordonnancement**

TCP : La connexion

- une connexion de type circuit virtuel est établie avant que les données ne soient échangées : appel + négociation + transferts

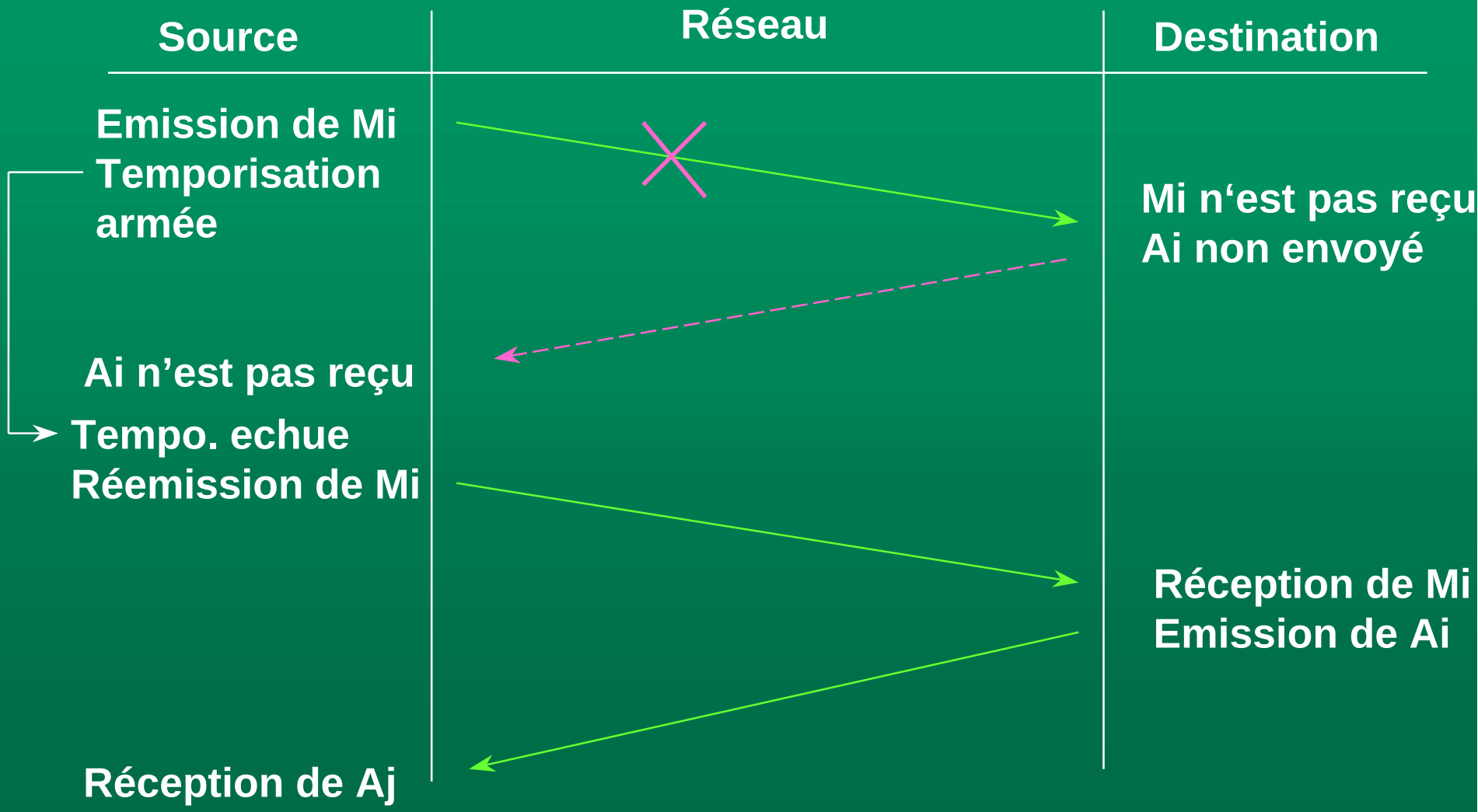
- Une connexion = une paire d'extrémités de connexion
- Une extrémité de connexion = couple (adresse IP, port)
- Exemple de connexion : ((124.32.12.1, 1034), (19.24.67.2, 21))
- Une extrémité de connexion peut être partagée par plusieurs autres extrémités de connexions (multi-instanciation)
- La mise en oeuvre de la connexion se fait en deux étapes :
  - une application (extrémité) effectue une ouverture passive en indiquant qu'elle accepte une connexion entrante,
  - une autre application (extrémité) effectue une ouverture active pour demander l'établissement de la connexion.

TCP : Segmentation

- **Segmentation, contrôle de flux**
  - **Les données transmises à TCP constituent un flot d'octets de longueur variable.**
  - **TCP divise ce flot de données en segments en utilisant un mécanisme de fenêtrage.**
  - **Un segment est émis dans un datagramme IP.**

- **Acquittement de messages**
  - **Contrairement à UDP, TCP garantit l'arrivée des messages, c'est à dire qu'en cas de perte, les deux extrémités sont prévenues.**
  - **Ce concept repose sur les techniques d'acquittement de message : lorsqu'une source S émet un message Mi vers une destination D, S attend un acquittement Ai de D avant d'émettre le message suivant Mi+1.**
  - **Si l'acquittement Ai ne parvient pas à S, S considère au bout d'un certain temps que le message est perdu et reémet Mi :**

TCP : Acquittements

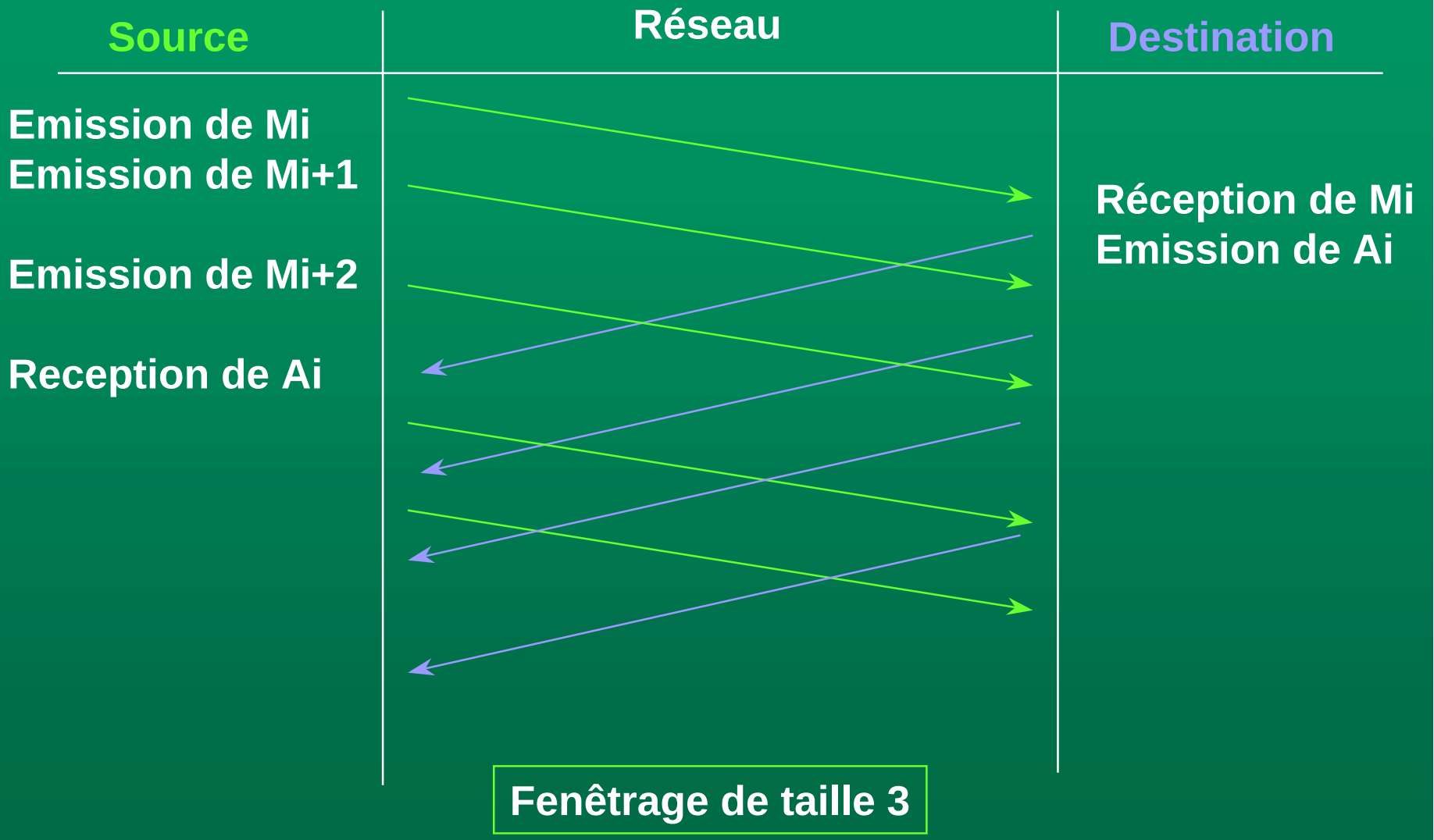


TCP : le fenêtrage

- **La technique acquittement simple pénalise les performances puisqu'il faut attendre un acquittement avant d'émettre un nouveau message. Le fenêtrage améliore le rendement des réseaux.**

- **La technique du fenêtrage : une fenêtre de taille T, permet l'émission d'au plus T messages "*non acquittés*" avant de ne plus pouvoir émettre :**

TCP : le Fenêtrage



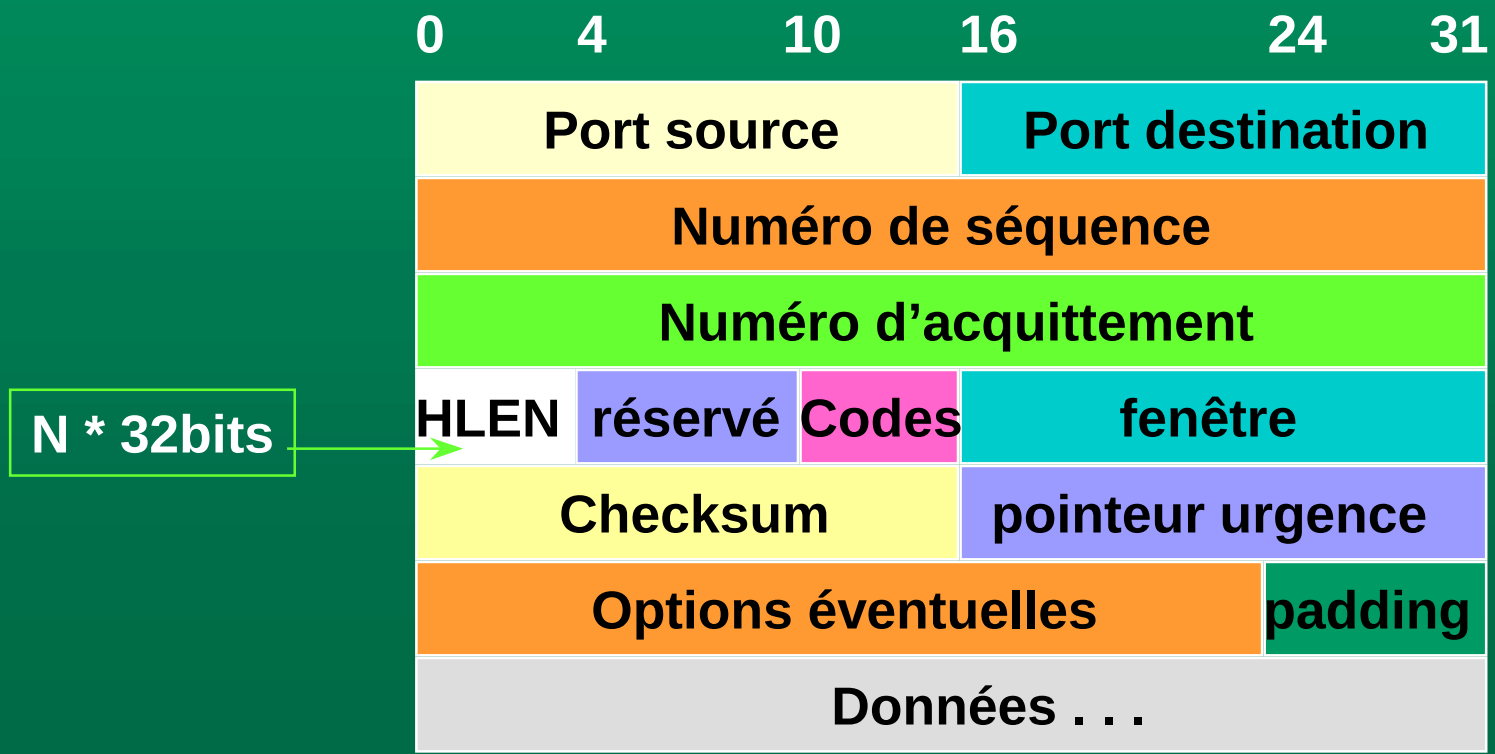
TCP : Technique de fenêtrage

- **fenêtrage glissante permettant d'optimiser la bande passante**
- **permet également au destinatairede faire diminuer le débit de l'émetteur donc de gérer le contrôle de flux.**
- **Le mécanisme de fenêtrage mis en oeuvre dans TCP opère au niveau de l'octet et non pas au niveau du segment; il repose sur :**

  - **la numérotation séquentielle des octets de données,**
  - **la gestion de trois pointeurs par fenêtrage :**

TCP : Segments

- **Segment : unité de transfert du protocole TCP.**
  - **échangés pour établir les connexions,**
  - **transférer les données,**
  - **émettre des acquittements,**
  - **fermer les connexions;**



TCP : format du segment

- **Numéro de séquence : le numéro de séquence du premier octet (NSP) de ce segment. Généralement à la suite d'octets O1, O2, ..., On (données du message) est associée la suite de numéro de séquence NSP, NSP+1, ..., NSP+n.**

  Il existe deux exceptions à cette règle :

  - **lorsque le bit SYN (voir CODE BITS) est positionné, le NSP représente cette donnée de contrôle et par conséquent la suite NSP, NSP+1, NSP+2, ..., NSP+n+1, associe la suite de données SYN, O1, O2, ..., On.**
  - **lorsque le bit FIN (voir CODE BITS) est positionné, le NSP+n représente cette donnée de contrôle et par conséquent la suite NSP, NSP+1, NSP+2, ..., NSP+n, associe la suite de données O1, O2, ..., On, FIN.**

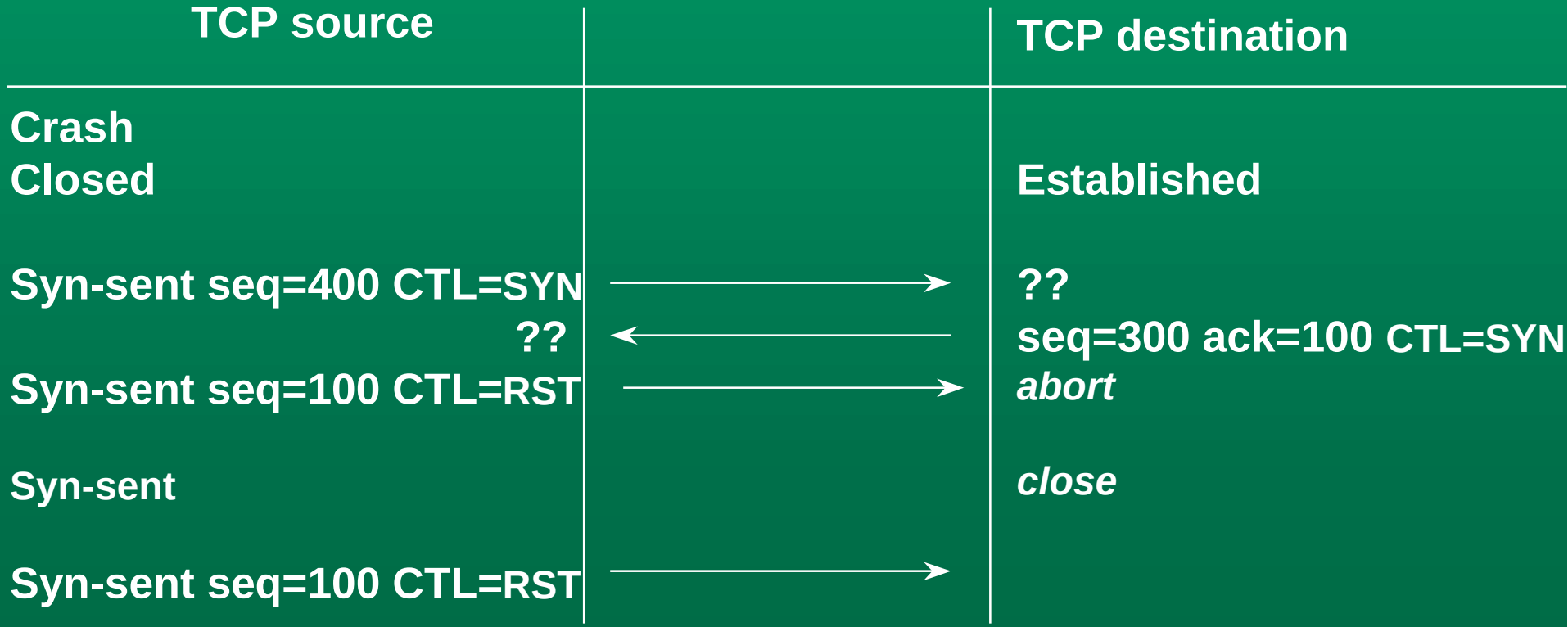
- **Numéro d'acquittement : le prochain numéro de séquence NS attendu par l'émetteur de cet acquittement. Acquitte implicitement les octets NS-1, NS-2, etc.**
- **Fenêtre: la quantité de données que l'émetteur de ce segment est capable de recevoir; ceci est mentionné dans chaque segment (données ou acquittement).**

TCP : Format du segment

- **CODE BITS : indique la nature du segment :**
  - **URG : le pointeur de données urgentes est valide (exemple : interrupt en remote login), les données sont émises sans délai, les données reçues sont remises sans délai.**
  - **SYN : utilisé à l'initialisation de la connexion pour indiquer où la numérotation séquentielle commence. Syn occupe lui-même un numéro de séquence bien que ne figurant pas dans le champ de données. Le Numéro de séquence inscrit dans le datagramme (correspondant à SYN) est alors un *Initial Sequence Number* (ISN) produit par un générateur garantissant l'unicité de l'ISN sur le réseau (indispensable pour identifier les duplications).**
  - **FIN : utilisé lors de la libération de la connexion;**
  - **PSH : fonction push. Normalement, en émission, TCP reçoit les données depuis l'applicatif , les transforme en segments à sa guise puis transfère les segments sur le réseau; un récepteur TCP décodant le bit PSH, transmet à l'application réceptrice, les données correspondantes sans attendre plus de données de l'émetteur. Exemple : émulation terminal, pour envoyer chaque caractère entré au clavier (mode caractère asynchrone).**

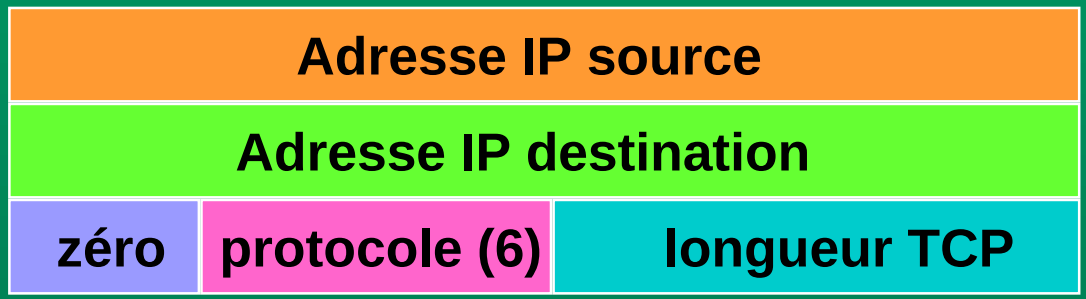
TCP : format du segment

- **RST : utilisé par une extrémité pour indiquer à l'autre extrémité qu'elle doit réinitialiser la connexion. Ceci est utilisé lorsque les extrémités sont désynchronisées. Exemple :**



TCP format du segment

- **<u>CHECKSUM</u> : calcul du champ de contrôle : utilise un pseudo-en-tête et s'applique à la totalité du segment obtenu (PROTO =6) :**



TCP : format du header

OPTIONS

- **Permet de négocier la taille maximale des segments échangés. Cette option n'est présente que dans les segments d'initialisation de connexion (avec bit SYN).**
- **TCP calcule une taille maximale de segment de manière à ce que le datagramme IP résultant corresponde au MTU du réseau. La recommandation est de 536 octets.**
- **La taille optimale du segment correspond au cas où le datagramme IP n'est pas fragmenté mais :**
  - **il n'existe pas de mécanisme pour connaître le MTU,**
  - **le routage peut entraîner des variations de MTU,**
  - **la taille optimale dépend de la taille des en-têtes (options).**

TCP : acquittements

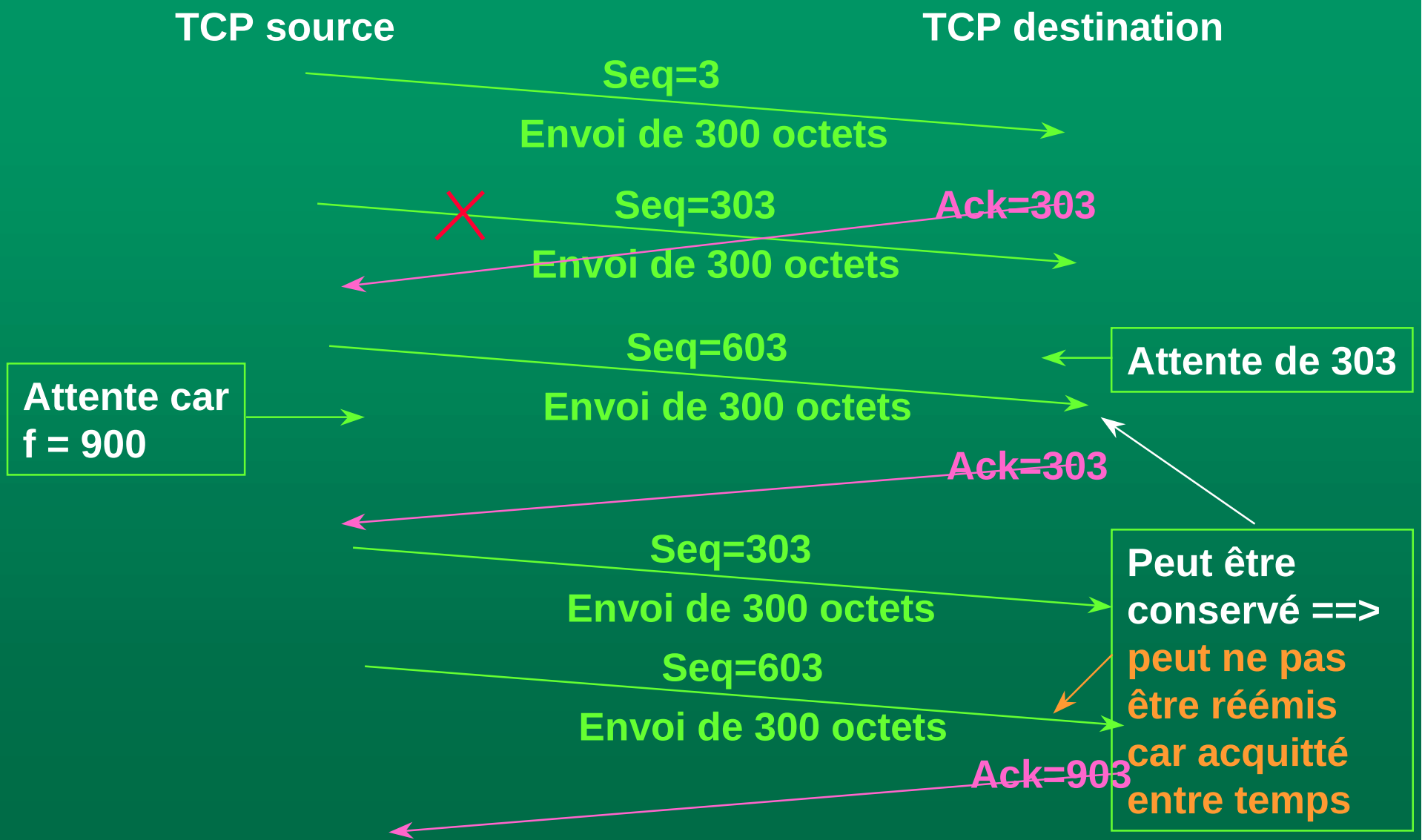
Acquittements et retransmissions

- **Le mécanisme d'acquittement de TCP est cumulatif :**
  - **il indique le numéro de séquence du prochain octet attendu : tous les octets précédents cumulés sont implicitement acquittés**
  - **Si un segment a un numéro de séquence supérieur au numéro de séquence attendu (bien que dans la fenêtre), le segment est conservé mais l'acquittement référence toujours le numéro de séquence attendu(-->).**
- **Pour tout segment émis, TCP s'attend à recevoir un acquittement**
  - **Si le segment n'est pas'acquitté, le segment est considéré comme perdu et TCP le retransmet.**
  - **Or un réseau d'interconnexion offre des temps de transit variables nécessitant le réglage des temporisations;**
  - **TCP gère des temporisations variables pour chaque connexion en utilisant un algorithme de retransmission adaptative**

TCP : Acquittements

Fenêtre=900

Segment=300



TCP : retransmissions

algorithme de retransmission adaptative

- **enregistre la date d'émission d'un segment,**
- **enregistre la date de réception de l'acquittement correspondant,**
- **calcule l'échantillon de temps de boucle A/R écoulé,**
- **détermine le temps de boucle moyen RTT (Round Trip Time) :**

  RTT = (a * anc_RTT) + ((1-a) * NOU_RTT))

  avec 0<= a < 1

  a proche de 1 : RTT insensible aux variations brèves,

  a proche de 0 : RTT très sensible aux variations rapides,
- **calcule la valeur du temporisateur en fonction de RTT.**
- **Les premières implémentations de TCP ont choisi un coefficient constant B pour déterminer cette valeur : Temporisation = B * RTT avec B >1 (généralement B=2).**
- **Aujourd'hui de nouvelles techniques sont appliquées pour affiner la mesure du RTT : l'algorithme de Karn.**

TCP : retransmissions

L'algorithme de Karn repose sur les constatations suivantes :

- **en cas de retransmission d'un segment, l'émetteur ne peut savoir si l'acquittement s'adresse au segment initial ou retransmis (ambiguïté des acquittements), =>l'échantillon RTT ne peut donc être calculé correctement,**
- **=> TCP ne doit pas mettre à jour le RTT pour les segments retransmis.**

- **L'algorithme de Karn combine les retransmissions avec l'augmentation des temporisations associées (*timer backoff*):**
  - **une valeur initiale de temporisation est calculée**
  - **si une retransmission est effectuée, la temporisation est augmentée (généralement le double de la précédente, jusqu'à une valeur plafond).**
- **Cet algorithme fonctionne bien même avec des réseaux qui perdent des paquets.**

TCP : la congestion

Gestion de la congestion

- **TCP gère le contrôle de flux de bout en bout mais également les problèmes de congestion liés à l'interconnexion.**

- **La congestion correspond à la saturation de noeud(s) dans le réseau provoquant des délais d'acheminement de datagrammes jusqu'a leur pertes éventuelles.**

- **Les extrémité ignorent tout de la congestion sauf les délais. Habituellement, les protocoles retransmettent les segments ce qui agrave encore le phénomène.**

- **Dans la technologie TCP/IP, les passerelles (niveau IP) utilisent la réduction du débit de la source mais TCP participe également à la gestion de la congestion en diminuant le débit lorsque les délais s'allongent :**

TCP : la congestion

- TCP maintient une fenêtre virtuelle de congestion
- TCP applique la fenêtre d'émission suivante:
  - fenêtre_autorisée = min (fenêtre_récepteur, fenêtre_congestion).

- Dans une situation de non congestion:
  - fenêtre_récepteur = fenêtre_congestion.

- En cas de congestion, TCP applique une diminution dichotomique :
  - à chaque segment perdu, la fenêtre de congestion est diminuée par 2 (minimum 1 segment)
  - la temporisation de retransmission est augmentée exponentiellement.

TCP retransmissions

- **Si la congestion disparaît, TCP définit une fenêtre de congestion égale à 1 segment et l'incrémente de 1 chaque fois qu'un acquittement est reçu; ce mécanisme permet un démarrage lent et progressif :**

  Fenêtre_congestion = 1,
  émission du 1er segment,
  attente acquittement,
  réception acquittement,

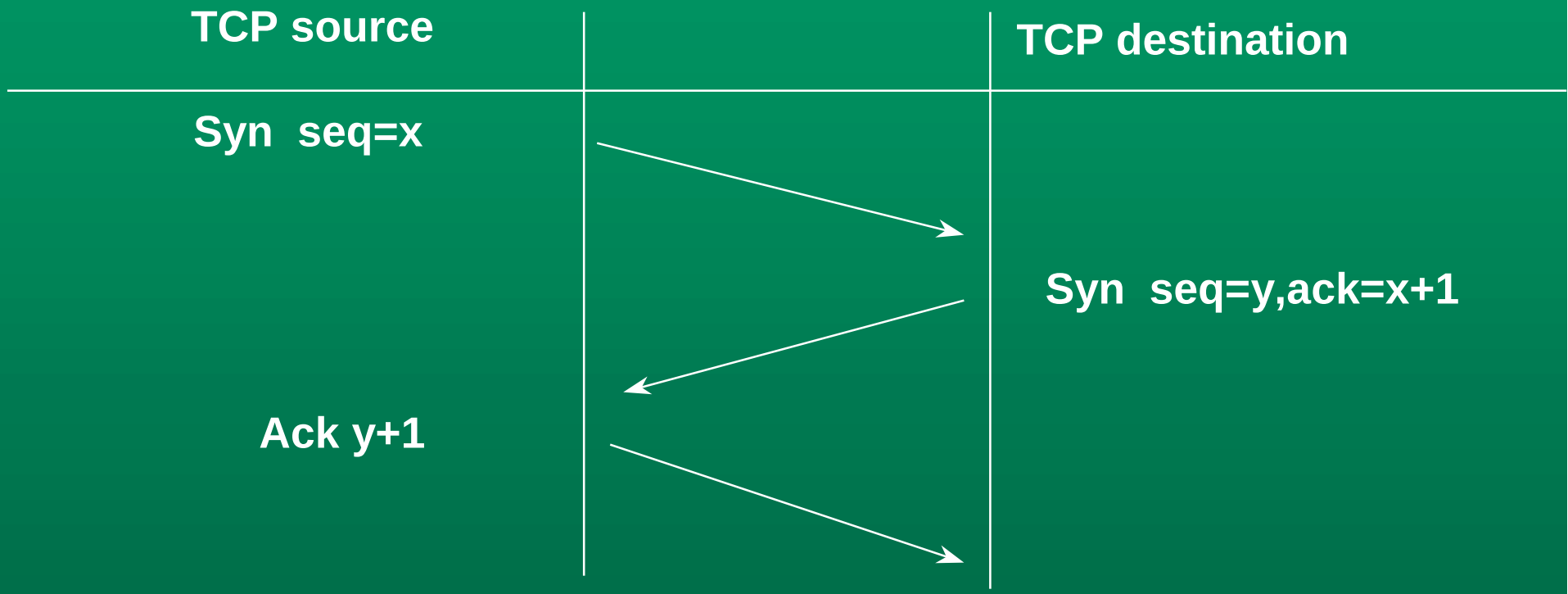
  Fenêtre_congestion = 2,
  émission des 2 segments,
  attente des acquittements,
  réception des 2 acquittements,

  Fenêtre_congestion = 4,
  émission des 4 segments, ...

Log2 N itérations pour envoyer N segments. Lorsque la fenêtre atteint une fois et demie sa taille initiale, l'incrément est limité à 1 pour tous les segments acquittés de la fenêtre.

TCP : connexion

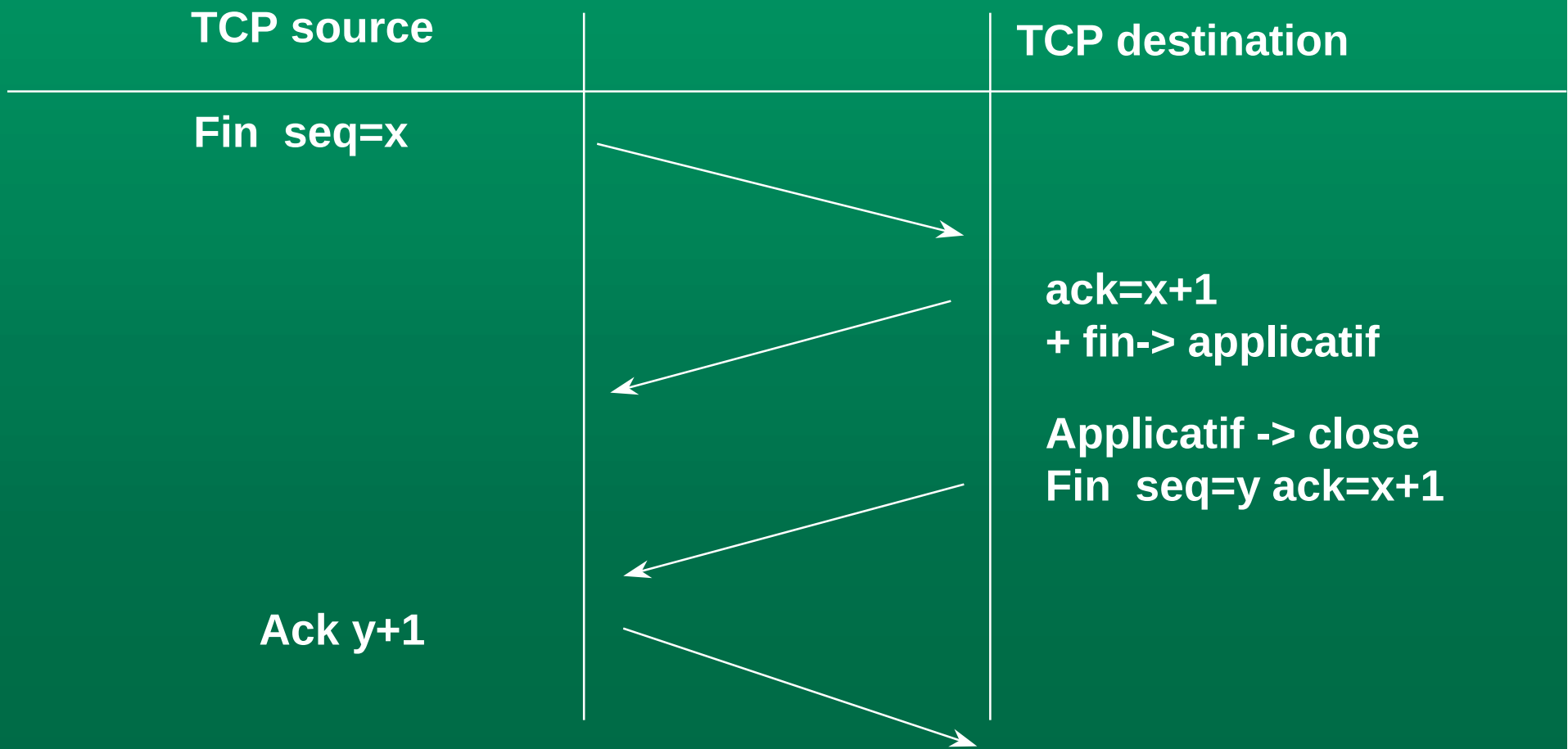
Une connexion TCP est établie en trois temps de manière à assurer la synchronisation nécessaire entre les extrémités :



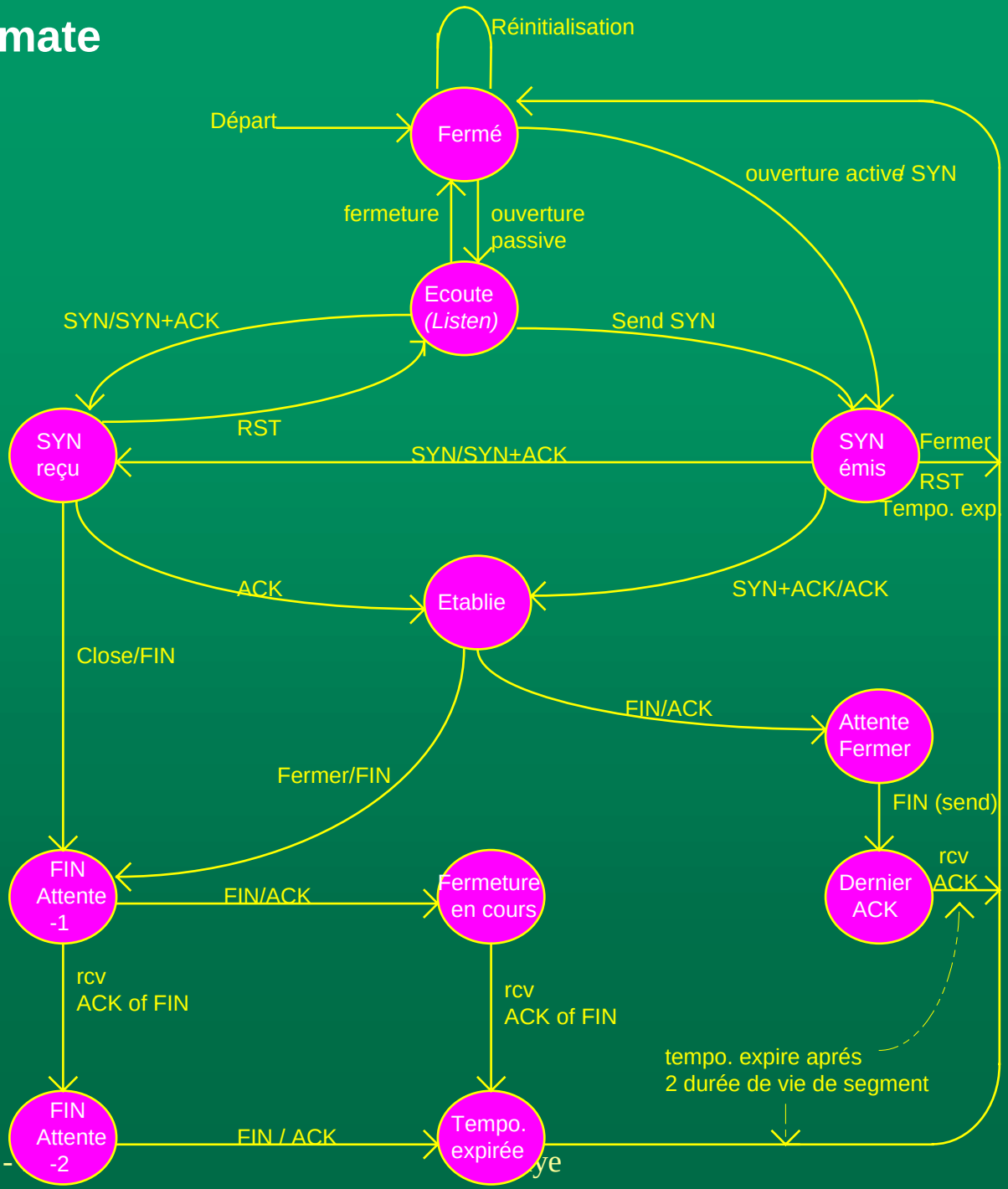
Ce schéma fonctionne lorsque les deux extrémités effectuent une demande d'établissement simultanément. TCP ignore toute demande de connexion, si cette connexion est déjà établie.

TCP : déconnexion

- **Une connexion TCP est libérée en un processus dit "trois temps modifié":**



TCP : L'automate



TCP : ports standards

No port	Mot-clé	Description
20	FTP-DATA	File Transfer [Default Data]
21	FTP	File Transfer [Control]
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer
37	TIME	Time
42	NAMESERVER	Host Name Server
43	NICNAME	Who Is
53	DOMAIN	Domain Name Server
79	FINGER	Finger
80	HTTP	WWW
110	POP3	Post Office Protocol - Version 3
111	SUNRPC	SUN Remote Procedure Call