Cours Réseau

Les protocoles TCP/IP

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'lnternet".
- 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)

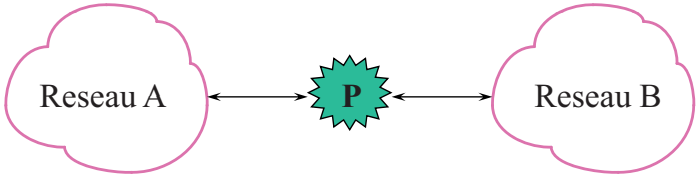
- Alternative à cette solution : mise en oeuvre de l'interconnexion au niveau des protocoles gérant la couche réseau de ces systèmes.
- Avantage considérable : les données sont routées par les noeuds intermédiaires sans que ces noeuds aient la moindre connaissance des applications responsables des ces 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.

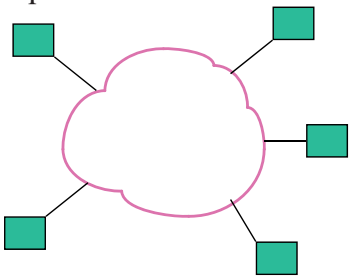
- Le rôle de la paserelle P est de transférer sur le réseau B, les paquets circulant sur le réseau A et destinés au réseau B et inversement.

Concepts de l'interconnexion (suite)

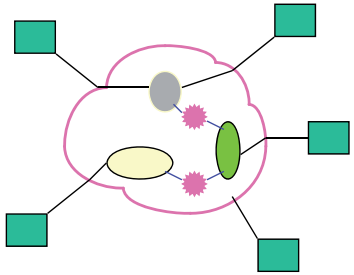
- P1 transfère sur le réseau B, les paquets circulant sur le réseau A et destinés aux réseaux B et C
- P1 doit avoir connaissance de la topologie du réseau; à savoir que C est accessible depuis le réseau B.
- Le routage n'est pas effectué sur la base de la machine destinataire mais sur la base du réseau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

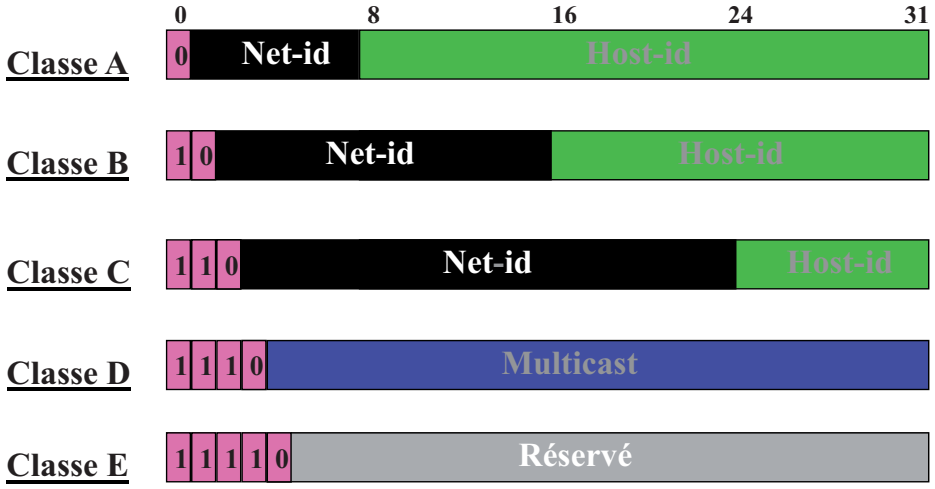
L'adressage Internet

- 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).

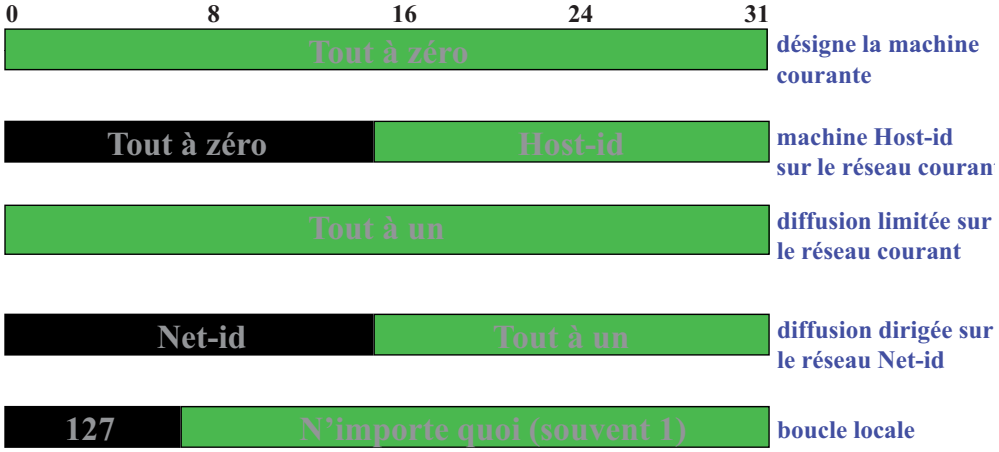
L'adressage Internet (suite)

  - 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)



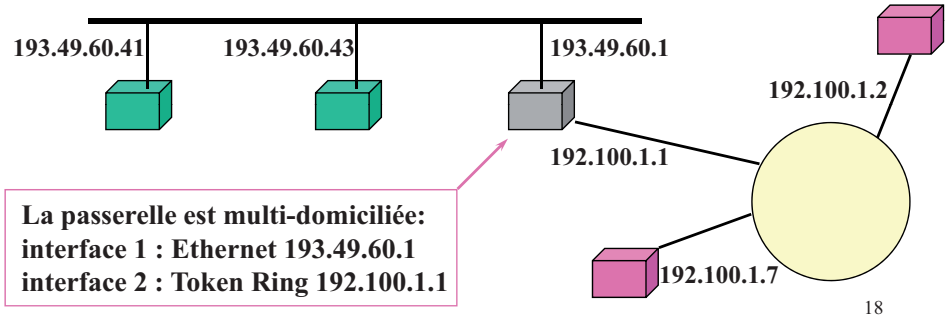
L'adressage Internet (suite)

•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.



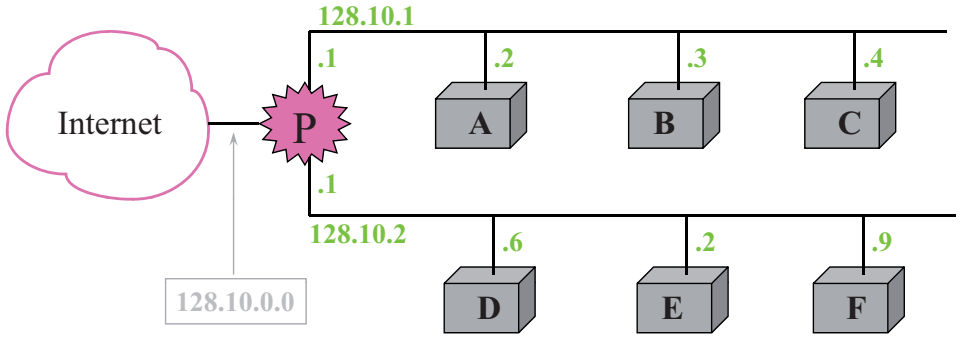
Le sous-adressage

- 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



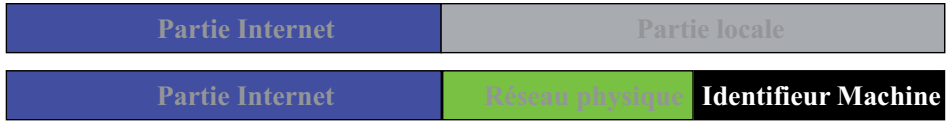
Un site avec deux réseaux physiques utilisant le sous-adressage de manière à ce que ses deux sous-réseaux soient couverts par une seule adresse IP de classe B.
La passerelle P accepte tout le trafic destiné au réseau 128.10.0.0 et sélectionne le sous-réseau en fonction du troisième octet de l'adresse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

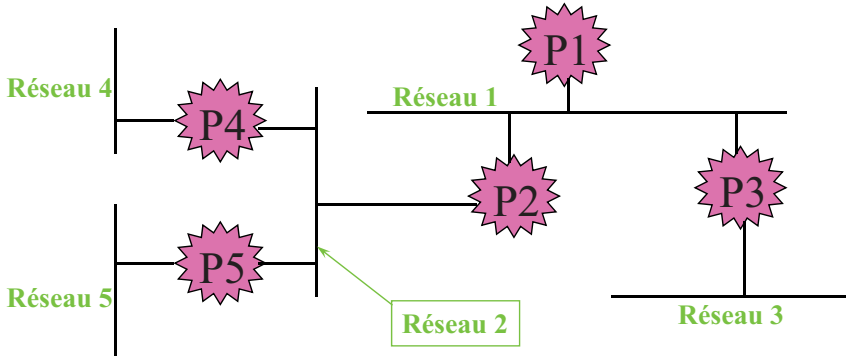


- **«Partie Internet» correspond au NetId (plan d'adressage initial)**
- **«Partie locale» correspond au hostid (plan d'adressage initial)**
- **les champs «Réseau physique» et «identifieur Machine» sont de taille variable; la longueur des 2 champs étant toujours égale à la longueur de la «Partie locale».**

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.

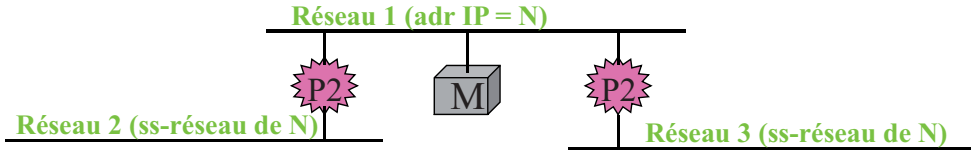
Le sous-adressage (suite)

- 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 }.

Protocoles de la couche réseau

ARP

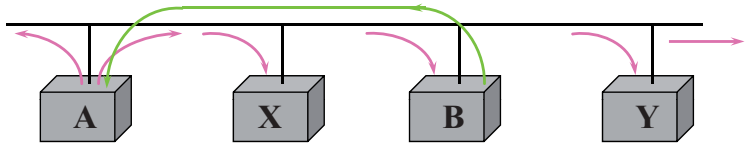
Address Resolution Protocol

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



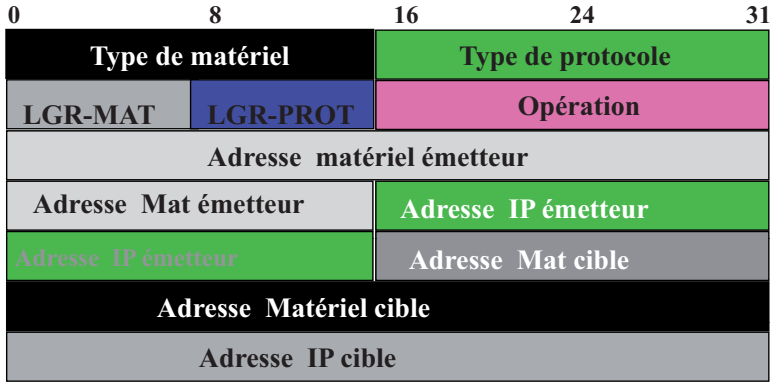
- Pour connaître l'adresse physique de B, PB, à partir de son adresse IP IB, la machine A diffuse une requête ARP qui contient l'adresse IB vers toutes les machines; la machine B répond avec un message ARP qui contient la paire (IB, PB).

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



ICMP

Internet Control Message Protocol

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).

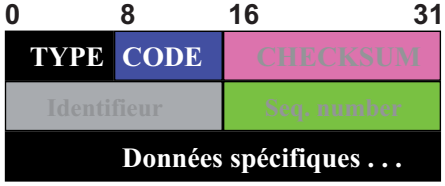
ICMP : format des messages

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

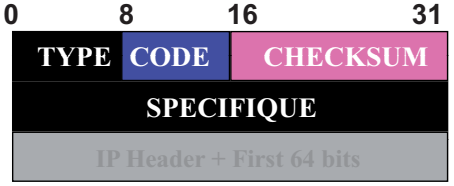
Demande et réponse d'information (*Information Request + Reply*)

- 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

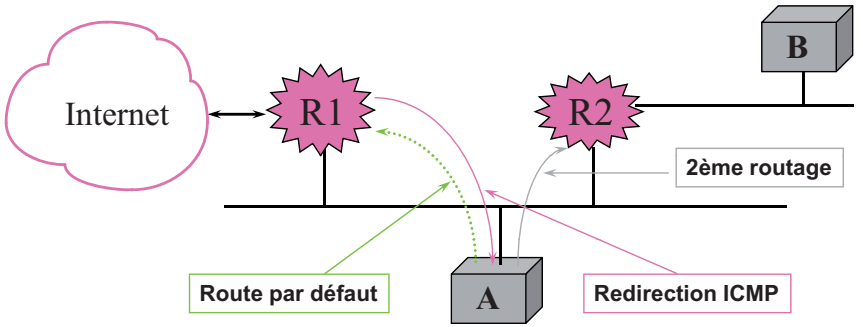
- Lorsqu'une passerelle émet un message ICMP de type destination inaccessible, le champ code décrit la nature de l'erreur :
  - 0 Network Unreachable
  - 1 Host Unreachable
  - 2 Protocol Unreachable
  - 3 Port Unreachable
  - 4 Fragmentation Needed and DF set
  - 5 Source Route Failed
  - 6 Destination 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.



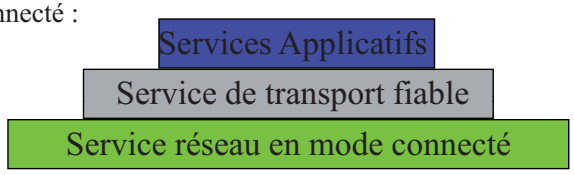
Une fois la redirection effectuée, les datagrammes seront acheminés vers la passerelle appropriée.

IP

Internet Protocol

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).**

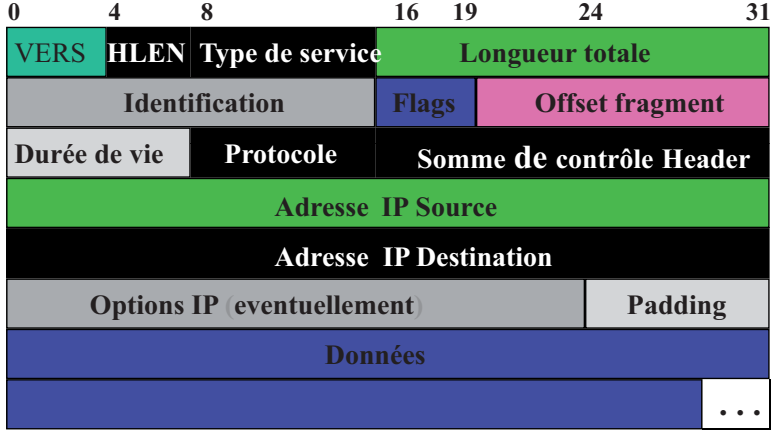
IP : Internet Protocol (suite)

- 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)

- **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)

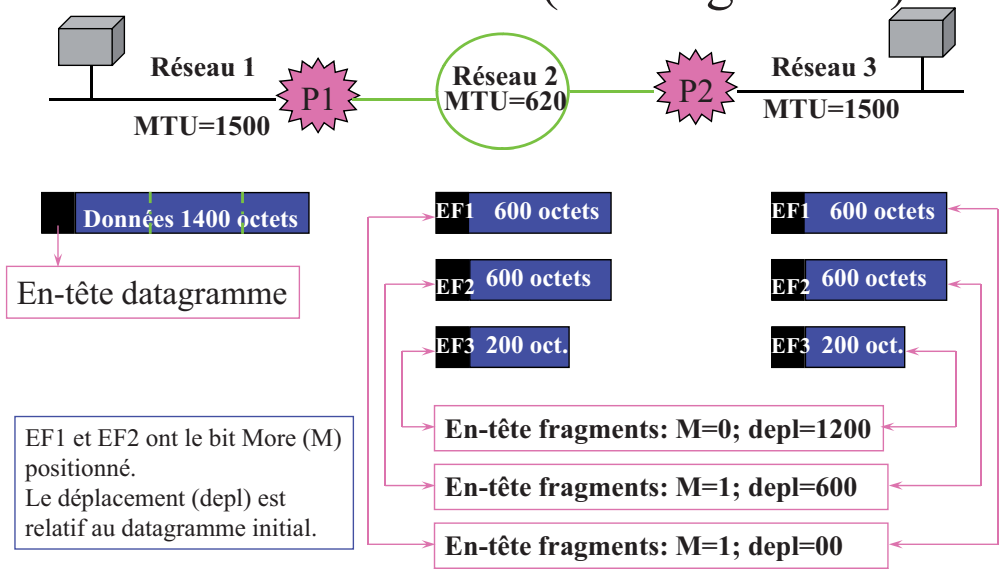
- FRAGMENT OFFSET, FLAGS, IDENTIFICATION : les champs de la fragmentation.
  - Sur toute machine ou passerelle mettant en oeuvre TCP/IP une unité maximale de transfert (*Maximum Transfert Unit* ou MTU) définit la taille maximale d'un datagramme véhiculé sur le réseau physique correspondant
  - lorsque 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.

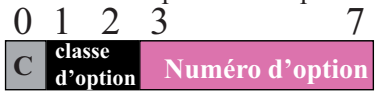
- chaque fragment a une structure identique à celle du datagramme initial, seul les champs FLAGS et FRAGMENT OFFSET sont spécifiques.

IP : Internet Protocol (le datagramme)



IP : Internet Protocol (le datagramme)

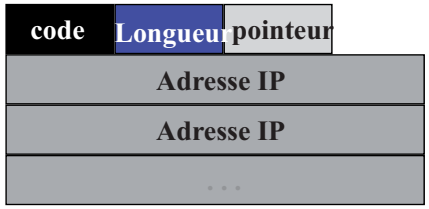
- OPTIONS
  - Le champ OPTIONS est facultatif et de longueur variable. Les options concernent essentiellement des fonctionnalités de mise au point. Une option est définie par un champ octet :



- **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 :**

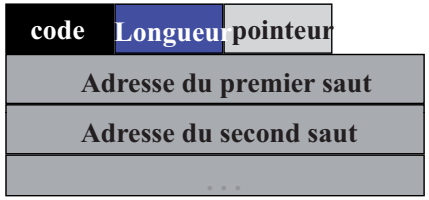
IP : Internet Protocol (le datagramme)

– Enregistrement de route (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)

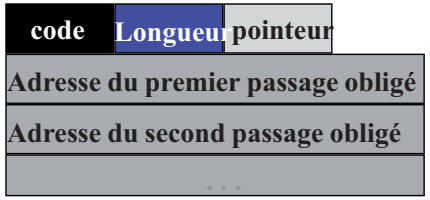
– Routage strict prédéfini par l'émetteur (classe = 0, option = 9): prédéfinit le routage qui doit être utilisé dans l'interconnexion en indiquant la suite des adresses IP dans l'option :



- **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*.**

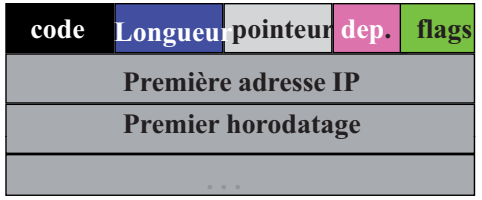
IP : Internet Protocol (le datagramme)

– 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.

Principe de routage

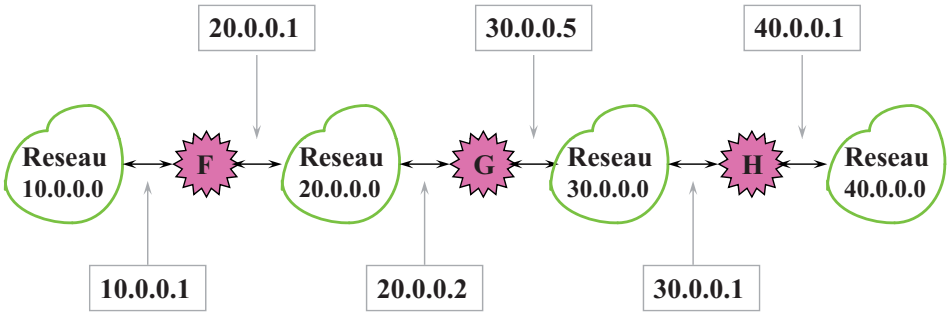
Routage des datagrammes

- 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)

- 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.

Protocoles de la couche transport

UDP

User Datagram Protocol

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.

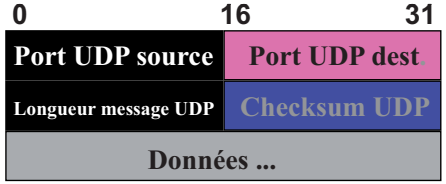
UDP : les ports

- 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.



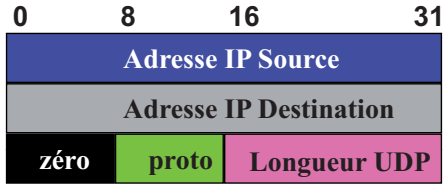
Format des messages UDP

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.

TCP

Transmission Control Protocol

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

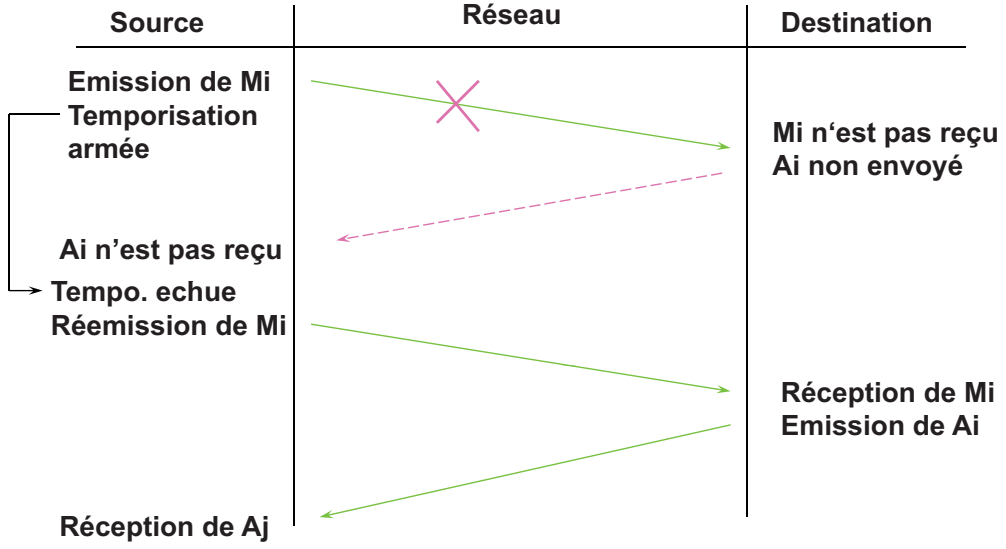
- Une connexion = une paire d'extrémités de connexion
- Une extrémité de connexion = couple (adresse IP, port)
- Exemple de connexion : ((124.32.12.1, 1034), (19.24.67.2, 21))
- Une extrémité de connexion peut être partagée par plusieurs autres extrémités de connexions (multi-instanciation)
- La mise 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

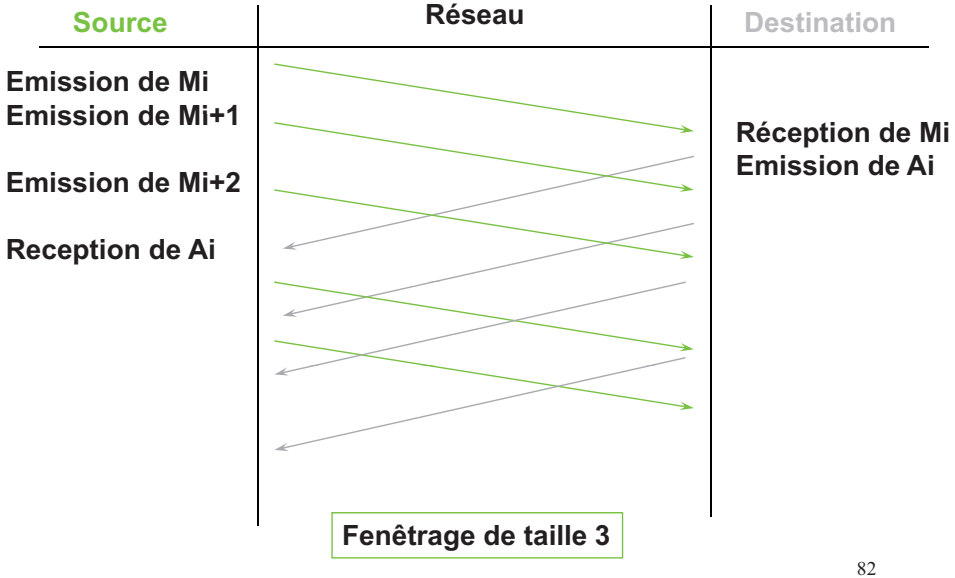


TCP : le fenêtrage

- 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

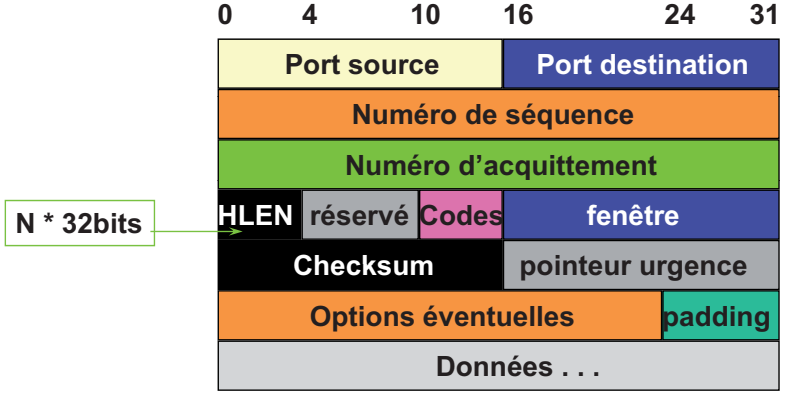


TCP : Technique de fenêtrage

- 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.

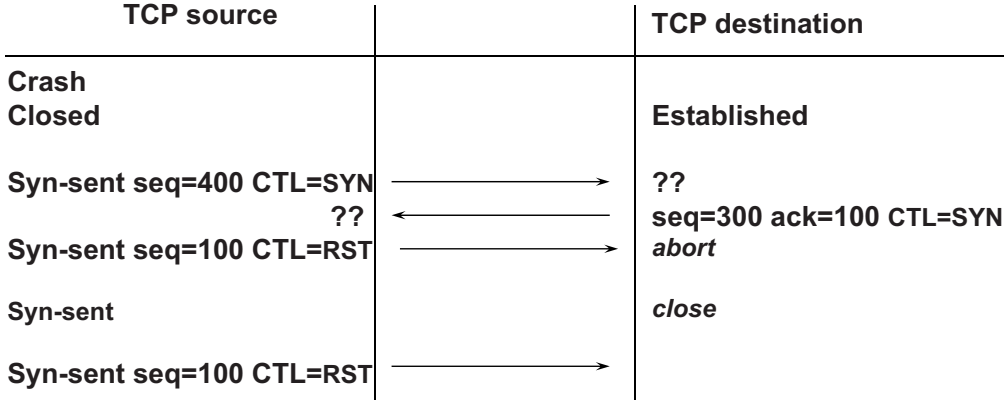
  – lorsque le bit FIN (voir CODE BITS) est positionné, le NSP+n représente cette donnée de contrôle et par conséquent la suite NSP, NSP+1, NSP+2, ..., NSP+n, associe la suite de données O1, O2, ..., On, FIN.
- Numéro d'acquittement : le prochain numéro de séquence NS attendu par l'émetteur de cet acquittement. 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).

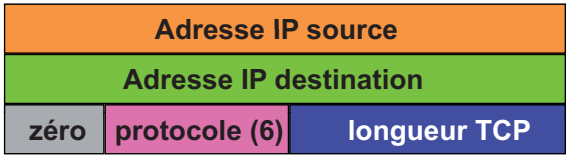
TCP : format du segment

- 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

- CHECKSUM : 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

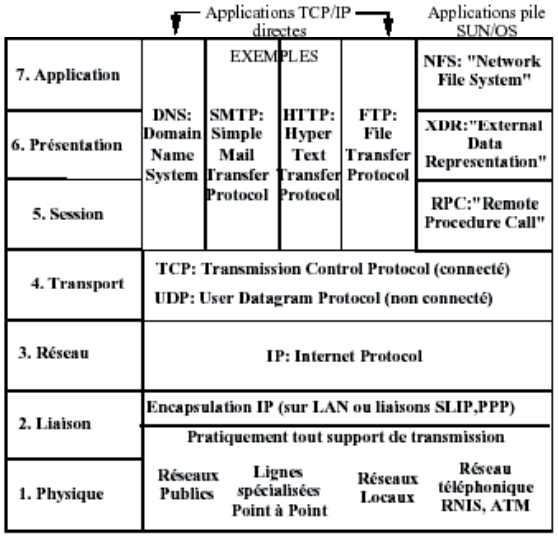
Acquittements et retransmissions

- 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/IP

Couche haute

Introduction



DNS

Domain Name System

DNS, le besoin

- L'Internet est constitué de réseaux (dizaines de milliers)
- Les réseaux sont constitués de sous-réseaux
- Les sous-réseaux sont constitués de machines,
- La technologie de base (TCP/IP) permet l'accès aux machines par leur adresse IP,
- Il est pratiquement devenu impossible aux humains de connaître les adresses (IP) des machines auxquelles ils veulent accéder.
- Le système DNS permet d'identifier une machine par un (des) nom(s) représentatif(s) de la machine et du (des) réseau(x) sur le(les)quel(s) elle se trouve ; exemple :

  www.algorithmics.fr identifie la machine www sur le réseau algorithmics.fr
- Le système est mis en œuvre par une base de données distribuée au niveau mondial
- Les noms sont gérés par un organisme mondial : l'interNIC et les organismes délégués : RIPE, NIC France, NIC Angleterre, etc.

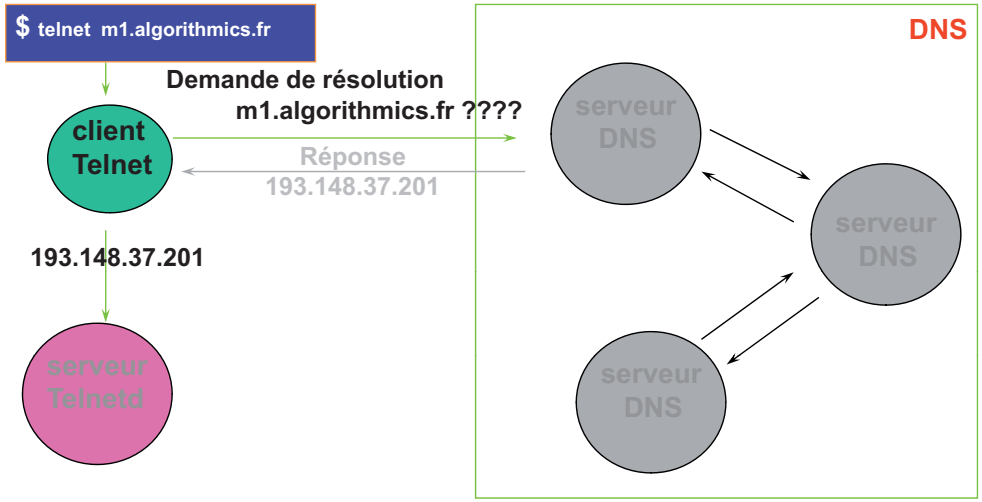
DNS, Le principe

- basé sur le modèle client / serveur
- le logiciel client interroge un serveur de nom; typiquement :
  - l'utilisateur associe un nom de domaine à une application ; exemple :

    telnet m1.algorithmics.fr
  - l'application cliente requiert la traduction du nom de domaine auprès d'un serveur de nom (DNS) : cette opération s'appelle la résolution de nom
  - le serveur de nom interroge d'autres serveurs de nom jusqu'à ce que l'association nom de domaine / adresse IP soit trouvée

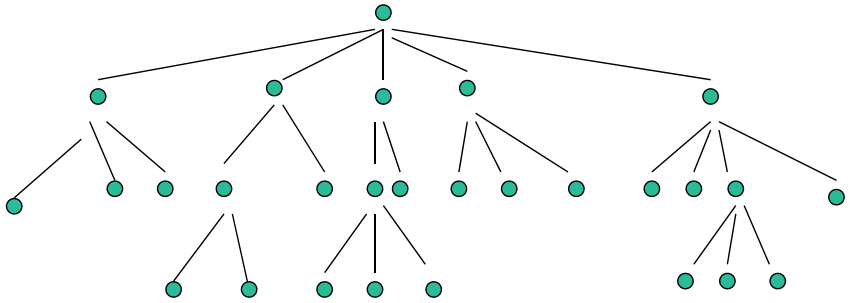
- le serveur de nom retourne l'adresse IP au logiciel client : 193.148.37.201
- le logiciel client contacte le serveur (telnetd) comme si l'utilisateur avait spécifié une adresse IP : telnet 193.148.37.201

DNS, Principe (illustration)



DNS, L'espace Nom de domaine

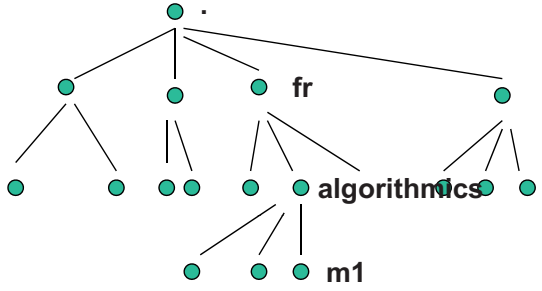
- Chaque unité de donnée dans la base DNS est indexée par un nom
- Les noms constituent un chemin dans un arbre inversé appelé l'espace Nom de domaine
- Organisation similaire à un système de gestion de fichiers



• Chaque nœud est identifié par un nom
• Racine appelée root, identifiée par «.»
• 127 niveaux au maximum

DNS, Les noms de domaine

Un nom de domaine est est la séquence de labels depuis le nœud de l'arbre correspondant jusqu'à la racine

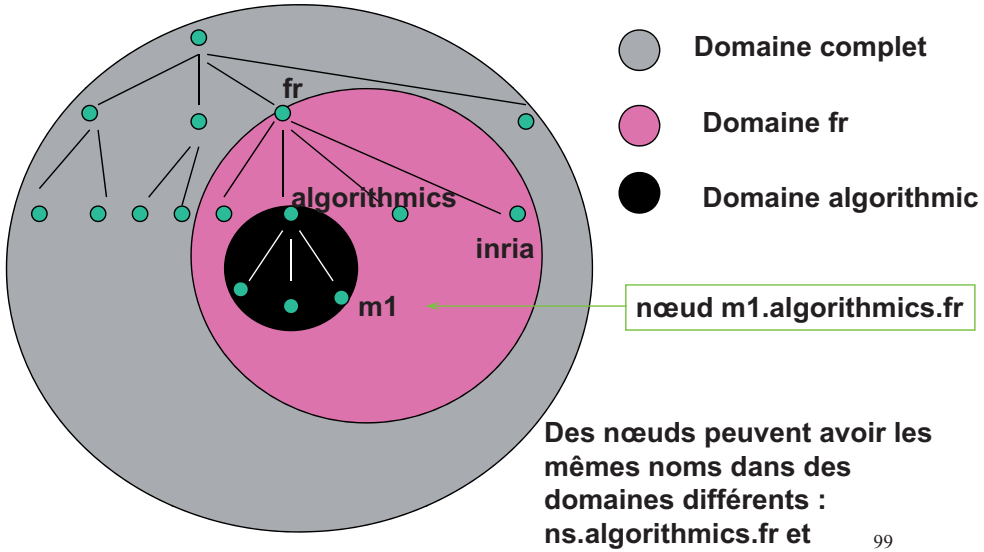


M1.algorithmics.fr

Deux nœuds fils ne peuvent avoir le même nom ==> unicité d'un nom de domaine au niveau mondial

DNS, Le domaine

Un domaine est un sous-arbre de l'espace nom de domaine



Des nœuds peuvent avoir les mêmes noms dans des domaines différents : ns.algorithmics.fr et ns.renault.fr

Concepts, résumé et extension

- Un domaine est un sous-arbre de l'espace Nom de domaine
- Un domaine est constitué de noms de domaine et d' autres domaines
- Un domaine intérieur à un autre domaine est appelé un sous domaine
- Exemple : le domaine fr comprend le nœud fr et tous les nœuds contenus dans tous les sous-domaines de fr

- Un nom de domaine est un index dans la base DNS; exemple :
  - m1.algorithmics.fr pointe vers une adresse IP
  - algorithmics.fr pointe vers des informations de routage de mail et éventuellement des informations de sous-domaines
  - fr pointe vers des informations structurelles de sous-domaines

- Les machines sont reliées entre elles dans un même domaine logiquement et non par adressage. Exemple : 10 machines d'un même domaine appartiennent à 10 réseaux différents et recouvrent 6 pays différents.

DNS, Domaines racine

- Le système DNS impose peu de règles de nommage :
  - noms < 63 caractères
  - Noms complet < 255 caractères
  - majuscules et minuscules non significatives
  - pas de signification imposée pour les labels
- Le premier niveau de l'espace DNS fait exception à la règle :
  - 7 domaines racines prédéfinis :
    - com : organisations commerciales ; ibm.com
    - edu : organisations concernant l'éducation ; mit.edu
    - gov : organisations gouvernementales ; nsf.gov
    - mil : organisations militaires ; army.mil
    - net : organisations réseau Internet ; worldnet.net
    - org : organisations non commerciales ; eff.org
    - int : organisations internationales ; nato.int
  - arpa : domaine réservé à la résolution de nom inversée
  - organisations nationales : fr, uk, de, it, us, au, ca, se, etc.

DNS, Domaines racine (suite)

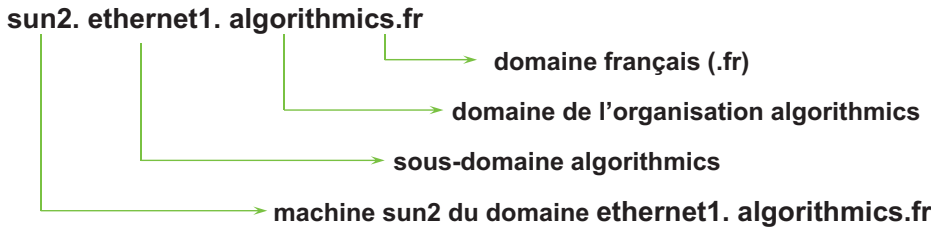
- Nouveaux domaines racine en cours de normalisation:
  - firm, store, web, arts, rec, info, nom
- Certaines organisations nationales peuvent être gérées administrativement par un consortium : RIPE
- Les divisions en sous-domaines existent dans certains pays et pas dans d'autres :
  - edu.au, com.au, etc.
  - co.uk, ac.uk, etc.
  - ca.ab, ca.on, ca.gb
  - pas de division du .fr

DNS, Lecture des noms de domaine

- A l'inverse de l'adressage IP la partie la plus significative si situe à gauche de la syntaxe :

sun2.ethernet1.algorithmics.fr ← vers le plus significatif

193.148.37.201 → vers le plus significatif

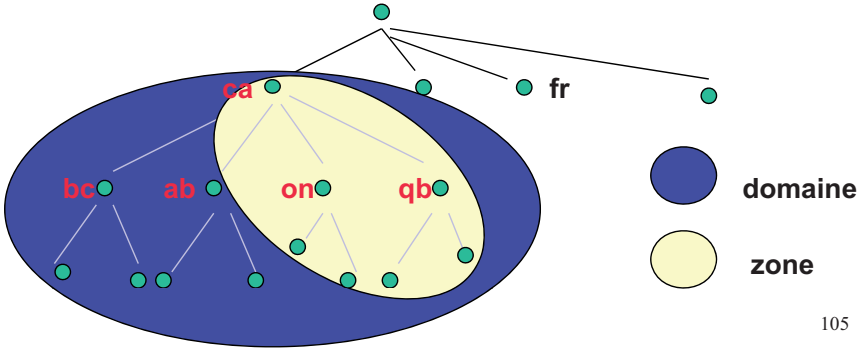


Délégation

- Le système DNS est entièrement distribué au niveau planétaire;
- A tout domaine est associé une responsabilité administrative
- Une organisation responsable d'un domaine peut
  - découper le domaine en sous-domaines
  - déléguer les sous-domaines à d'autres organisations :
    - qui deviennent à leur tour responsables du (des) sous-domaine(s) qui leurs sont délégué(s)
    - peuvent, à leur tour, déléguer des sous-domaines des sous-domaines qu'elles gèrent
- Le domaine parent contient alors seulement un pointeur vers le sous-domaine délégué; exemple :
  - algorithmics.fr est délégué à l'organisation algorithmics
  - La société algorithmics gère donc les données propres à ce domaine.
  - algorithmics.fr (en théorie seulement) pourrait être géré par l'organisation responsable du domaine .fr (NIC France) qui gèrerait alors les données de algorithmics.fr

Les serveurs de noms

- Les logiciels qui gèrent les données de l'espace nom de domaine sont appelés des serveurs de nom (*name servers*)
- Les serveurs de nom enregistrent les données propres à une partie de l'espace nom de domaine dans une zone.
- Le serveur de nom à autorité administrative sur cette zone.
- Un serveur de nom peut avoir autorité sur plusieurs zone.
- Une zone contient les informations d'un domaine sauf celles qui sont déléguées.



Différences entre Domaine et Zone

- Un domaine définit un ensemble de noms qui ont le même suffixe ⇒ ces **un découpage syntaxique** de l'espace de nommage Internet
- Une zone est un ensemble de noms ayant un même suffixe est servis par le même *name servers* ⇒ ces **un découpage administratif** définissant la portée d'action des serveurs de noms.

Exemple (schéma précédent)

- La zone **CA** possède des *name servers* qui servent aussi les sous-domaines **ON** et **QB**
- Le serveur **CA** ne traite pas le service du sous-domaine **AB** qui à sont propre *name servers* de zone

Types de serveurs de nom

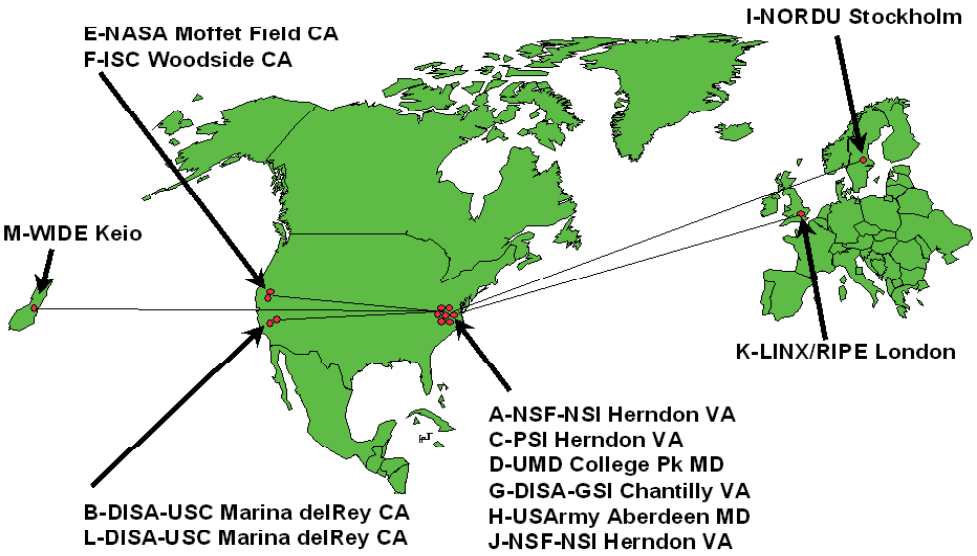
- Serveur de nom primaire : maintient la base de données de la zone dont il a l'autorité administrative
- Serveur de nom secondaire : obtient les données de la zone via un autre serveur de nom qui a également l'autorité administrative
  - interroge périodiquement le serveur de nom primaire et met à jour les données
- Il y a un serveur primaire et généralement plusieurs secondaires
- La redondance permet la défaillance éventuelle du primaire et du (des) secondaire(s)
- Un serveur de nom peut être primaire pour une (des) zone(s) et secondaire pour d'autre(s).

Serveurs racine

- Les serveurs racine connaissent les serveurs de nom ayant autorité sur tous les domaines racine
- Les serveurs racine connaissent au moins les serveurs de noms pouvant résoudre le premier niveau (.com, .edu, .fr, etc.)
- Pierre angulaire du système DNS : si les serveurs racine sont inoperationnels ==> plus de communication sur l'Internet
  - ==> multiplicité des serveurs racines
  - actuellement jusqu'à 14 éparpillés sur la planète
  - chaque serveur racine reçoit environ 100000 requêtes / heure
- Exemple de résolution : m1.algorithmics.fr à partir de ....edu

DNS Root Servers

Designation, Responsibility, and Locations

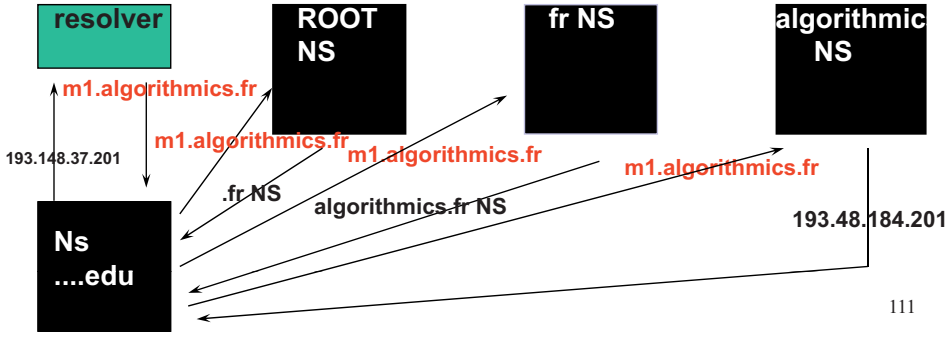


Resolver

- Les «resolvers» sont les processus clients qui contactent les serveurs de nom
- Fonctionnement :
  - contacte un name serveur (dont l' (les) adresse(s) est (sont) configurées sur la machine exécutant ce resolver)
  - interprète les réponses
  - retourne l'information au logiciel appelant
  - gestion de cache (dépend de la mise en œuvre)
- Le serveur de nom interroge également d'autres serveurs de nom, lorsqu'il n'a pas autorité sur la zone requise (fonctionnement itératif ou récursif)
- Si le serveur de nom est en dehors du domaine requis, il peut être amené à contacter un serveur racine (ne pas confondre avec un domaine racine)

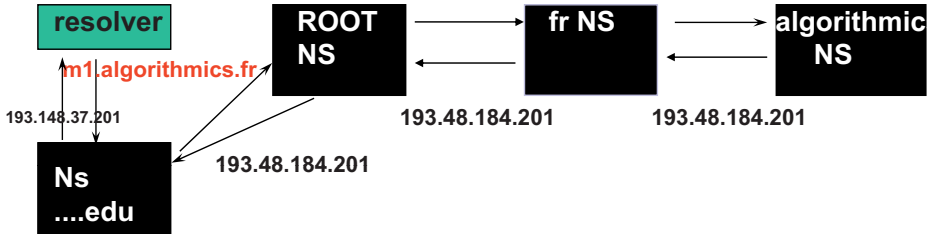
La recherche itérative

- On interroge successivement les serveurs
- **Exemple:** pour trouver le nom m1.algorithmics.fr le resolver interroge:
  1. Le serveur local **edu**
  2. Le serveur local interroge le serveur **root**, puis le serveur fr, puis le serveur algorithmics.fr



La recherche récursive

- Chaque serveur visité prend l'initiative d'interroger le serveur suivant pour obtenir pour lui même la réponse à la question posée.
- La réponse revient en visitant tous les sites.
- On note la résolution effectuée dans les caches de tous les serveur visités



•Pour plus d'efficacité, on utilise des caches.

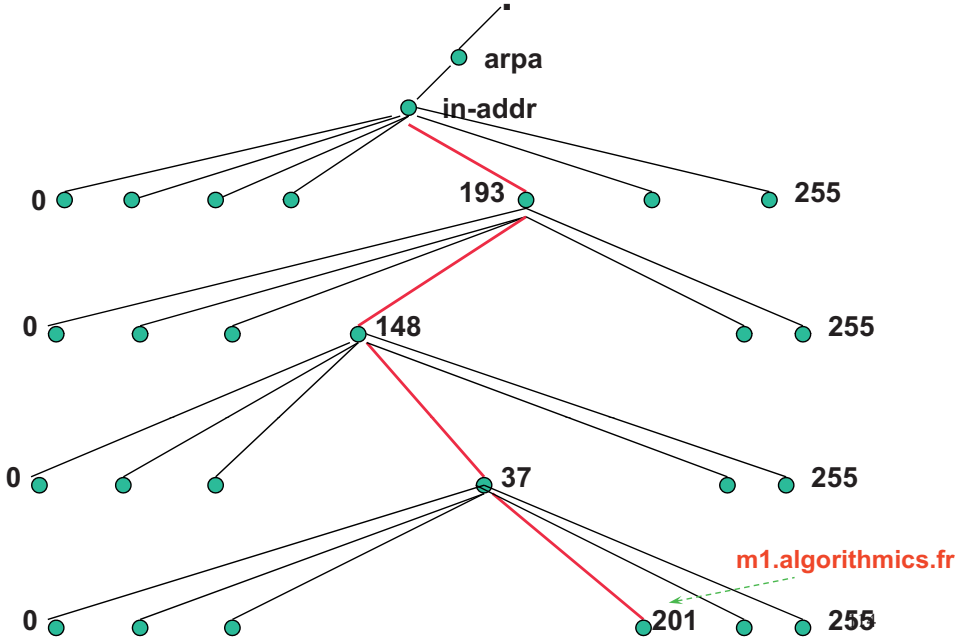
• Les noms récemment utilisés sont enregistrés dans une mémoire cache.

• Lorsqu'un client demande un nom, le serveur vérifie que celui-ci n'est pas dans la mémoire cache.

Résolution inverse

- Consiste a obtenir le nom de domaine à partir de l'adresse IP
  - pour faciliter la compréhension des humains
  - pour des raisons de sécurité
- Plus délicate que nom -> IP car le système DNS est organisé pour la résolution de nom ==> recherche exhaustive ???
- Solution : utiliser les adresses comme des noms :
  - le domaine in-addr.arpa
  - les noms des nœuds correspondent aux octets de l'adresse IP en ordre inverse
  - le domaine in-addr.arpa a 256 sous-domaines,
  - chacun de ces sous-domaines a 256 sous-domaines,
  - chacun de ces sous-domaines a, à son tour, 256 sous-domaines,
  - le 4ème niveau correspond à un NS connaissant le nom de domaine associé à cette adresse IP

Résolution inverse (suite)



Résolution inverse (suite)

- le nom de domaine associé à la résolution inverse est noté selon l'adresse IP inversée :
  - car la résolution d'un nom de domaine se fait de droite à gauche
  - exemple : 210.37.148.193.in-addr.arpa
  - résolution :
    - in-addr.arpa -> A.ROOT-SERVER.NET
    - 193.in-addr.arpa -> NS.RIPE.NET
    - 148. 193.in-addr.arpa -> NS.RIPE.NET
    - 37.148. 193.in-addr.arpa -> first.tvt.fr
  - Organismes gérant les classes
    - Classe A et B -> internic US.
    - Classe C
      - **192 : internic**
      - **193, 194, 195 RIPE avec délégations nationales**

La base de données du DNS

- Le DNS gère une base de données répartie d'unité d'information appelées Resources records « RR »

  RR = { nom-de-domaine, durée-de-vie, classe, type, valeur}
- nom-de-domaine: identifie un nœud de l'arborescence.
- durée-de-vie: définit la durée de validité de l'information dans un cache (entier et en seconde)
- Classe: identifie le protocole utilisateur (essentiellement IN pour Internet)
- Valeur: contient les données significatives associées au type (adresse, nom de domaine, chaîne de caractères).
- Type: identifie le type de donnée stockée.
  - SOA: décrit l'autorité administrative, (serveur primaire, mail admin, num de serie des information utilisées, durée entre deux mise à jour, délai entre deux tentatives mise à jour d'un secondaire, durée de vie des infos)

ex {algorithmics.fr 3600 IN SOA ns.algorithmics.fr.
fplaye.algorithmics.fr. =mail adr
64 = serial number
21600 = refresh
3600 = retry
86400 ; expire}

La base de données du DNS

- NS : liste de serveurs de nom pour ce domaine

{algorithmics.fr 88600 IN NS ns.algorithmics.fr.}

- A : correspondance nom -> adresse

{m1 88600 IN A 193.48.184.201}

- PTR : correspondance adresse -> nom

{201.184.48.193 88600 IN PTR m1. algorithmics.fr }

- CNAME : alias

{ftp. algorithmics.fr 88600 IN CNAME toto. algorithmics.fr }

- TXT : associe un texte a un nom de domaine

{algorithmics.fr 88600 IN TXT bienvenu chez algorithmics.}

- HINFO : description machine

{toto.algorithmics.fr 88600 IN HINFO sun unix.}

- MX : serveur de courrier électronique d'un domaine (avec priorité)

{algorithmics.fr 88600 IN MX 10, bobo.algorithmics.fr.}

Liste complète des types d'enregistrement

Symbole (Code)	Signification	RFC
A (1)	Address.	(RFC 1035).
AAAA(28)	IPv6 address.	(RFC 1886).
AFSDB	AFS Data Base locat.	(RFC 1183).
CNAME(5)	Canonical NAME.	(RFC 1035).
HINFO(13)	Host INFOrmation.	(RFC 1035).
ISDN	ISDN.	(RFC 1183).
KEY	Public KEY.	(RFC 2065).
KX	Key eXchanger.	(RFC 2230).
LOC	LOCation.	(RFC 1876).
MB	MailBox.	(RFC 1035).
MG		(RFC 1035).
MINFO		(RFC 1035).
MR		(RFC 1035).
MX (15)	Mail eXchanger.	(RFC 1035).
NULL		(RFC 1035).
NS (2)	Name Server.	(RFC 1035).
NSAP	Network Service Access Point Redéfini par le RFC 1706.	
NXT	NeXT.	(RFC 2065).
PTR (12)	PoinTeR.	(RFC 1035).
PX	Pointer X400/RFC822	(RFC 1664).
RP	Responsible Person.	(RFC 1183).
RT	Route Through.	(RFC 1183).
SIG	Crypto SIGnature.	(RFC 2065).
SOA (6)	Start Of Authority.	(RFC 1035).
SRV	SeRVer.	(RFC 2052).
TXT	TeXT.	(RFC 1035).
WKS	Well-Known Service.	(RFC 1035).
X25	X25.	(RFC 1183).

Utilisation du système DNS

- Utiliser un serveur de nom
  - machine elle-même serveur de nom : 127.0.0.1
  - machine non serveur de nom : spécfier un ou plusieurs serveur de nom : adresses IP obligatoirement. éventuellement son domaine.
  - sous UNIX : fichier /etc/resolv
  - sous NT, W95 : administration TCP/IP
- Administrer un serveur de nom
  - plateformes UNIX, NT
  - mémoire importante : mini 16/32 MB pour le service.
  - impératif : ne pas swapper
  - opérationnelle 24/24
  - laisser passer le port 53 sur UDP et TCP
- Debugging : Nslookup

SMTP

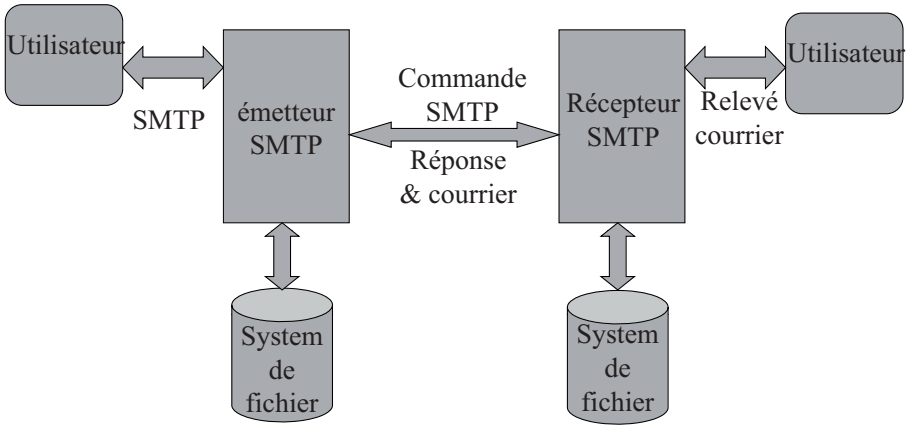
Simple Mail Transfert Protocol

SMTP
Simple Mail Transfert Protocol

définition

- Protocole d'échange de messages électroniques indépendant du protocole de transport sou jacent.
- utilise le service TCP via le port par défaut: **25**
- **Un serveur SMTP** est une machine « cible » qui ce présente comme un « bureau de poste » vis à vis du **client SMTP**
- Protocole directement accessible vis telnet.

SMTP
Principes de fondateurs



SYNTAXE DE COMMANDES SMTP

- `HELO <SP> <domaine> <CRLF>` ouverture de session entre le client et le serveur
- `MAIL <SP> FROM:<route-inverse> <CRLF>` définit l'adresse mail de l'émetteur
- `RCPT <SP> TO:<route-directe> <CRLF>` définit l'adresse mail du destinataire
- `DATA <CRLF>` définit le corps du message
- `RSET <CRLF>` abandonner le courrier en cours de transmission et restaurer la connexion
- `SEND <SP> FROM:<route-inverse> <CRLF>` le message doit être affiché sur le terminal utilisateur
- `SOML <SP> FROM:<route-inverse> <CRLF>`le message doit être affiché sur le terminal utilisateur sinon copié dans la boite au lettres
- `SAML <SP> FROM:<route-inverse> <CRLF>` le message doit être affiché sur le terminal utilisateur et copié dans la boite au lettres

SYNTAXE DE COMMANDES SMTP

- `VRFY <SP> <chaîne> <CRLF>` vérifier une adresse de destinataire sans lui transmettre de courrier
- `EXPN <SP> <chaîne> <CRLF>` expansion d'une liste de diffusion
- `HELP [<SP> <chaîne>] <CRLF>` demande d'information d'aide au serveur
- `NOOP <CRLF>` oblige simplement le serveur a répondre OK
- `QUIT <CRLF>` termine un courrier
- `TURN <CRLF>` inversion des rôles client serveur pour envoyer un courrier dans l'autre sens sans ouvrir une nouvelle connexion TCP

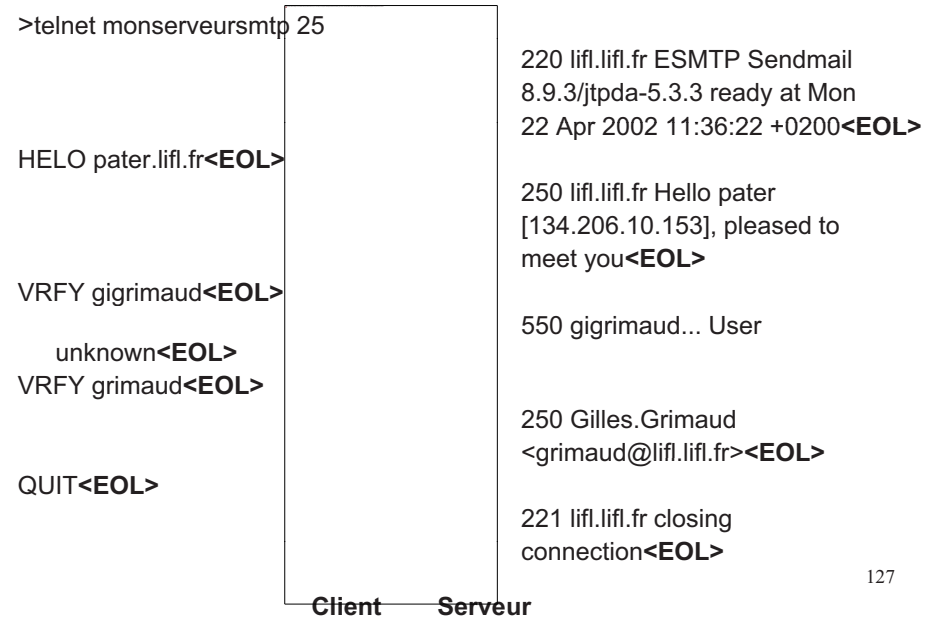
CODES REPONSE PAR ORDRE NUMERIQUE

- 211 État système, ou réponse d'aide système
- 214 Message d'aide [Informations sur l'utilisation d'un récepteur ou signification d'une commande non standard particulière; en clair pour un opérateur humain]
- 220 <domaine> Service disponible
- 221 <domaine> Canal de transmission en cours de fermeture
- 250 Action de messagerie effectuée, succès
- 251 Utilisateur non local ; réémission vers <route-directe> (avec relais automatique)
- 354 Début de message ; arrêt par <CRLF>.<CRLF>
- 421 <domaine> Service non disponible, canal en fermeture [Réponse à émettre sur tous les canaux lorsque le système exécute une séquence d'arrêt]
- 450 Action non effectuée : boîte-aux-lettres non disponible [Ex., bloquée par un autre utilisateur]

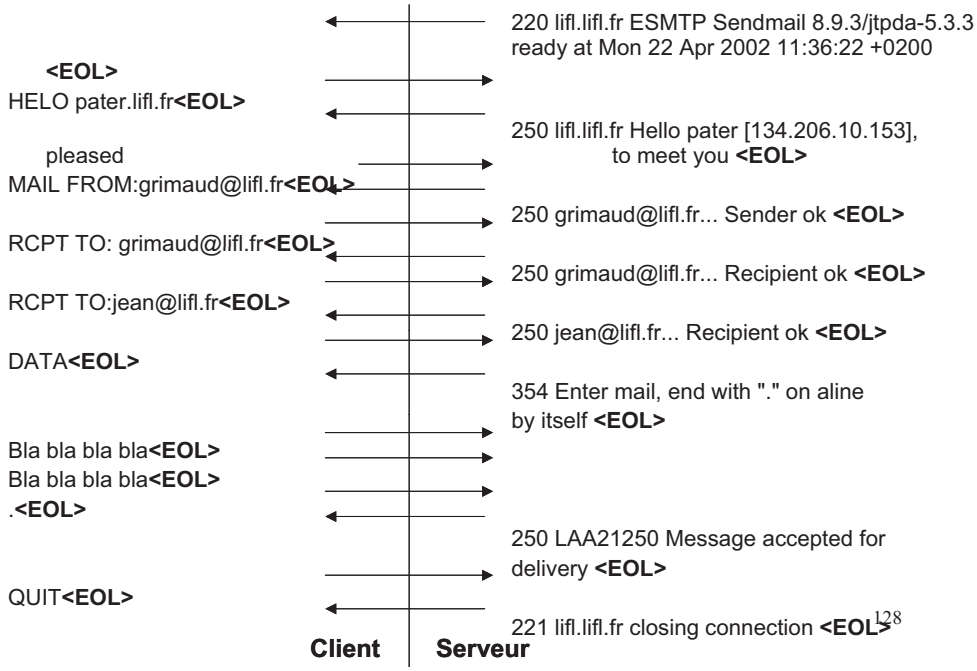
CODES REPONSE PAR ORDRE NUMERIQUE

- 451 Action arrêtée : erreur de traitement
- 452 Action non effectuée : manque de ressources système
- 500 Erreur de syntaxe, commande non reconnue [y compris des erreurs de type "ligne de commande trop longue"]
- 501 Erreur de syntaxe dans des paramètres ou arguments
- 502 Commande non implémentée
- 503 Mauvaise séquence de commandes
- 504 Paramètre de commande non implémenté
- 550 Action non effectuée : boîte-aux-lettres non disponible [Ex., boîte-aux-lettres non trouvée, pas d'accès]
- 551 Utilisateur non local ; essayer <route> (sans relais automatique)
- 552 Action arrêtée : manque de ressources de stockage
- 553 Action non effectuée : nom de boîte-aux-lettres non autoris é [Ex., erreur de syntaxe dans le nom de boîte]
- 554 Transaction échouée

Scénario 1 : validation d'une adresse mail



Scénario 2 : envoi d'un message



Nommage

- Les noms d'émetteurs et de destinataires sont des couples (nom de boite au lettres, nom de domaine ou est gérée la boite à lettre)

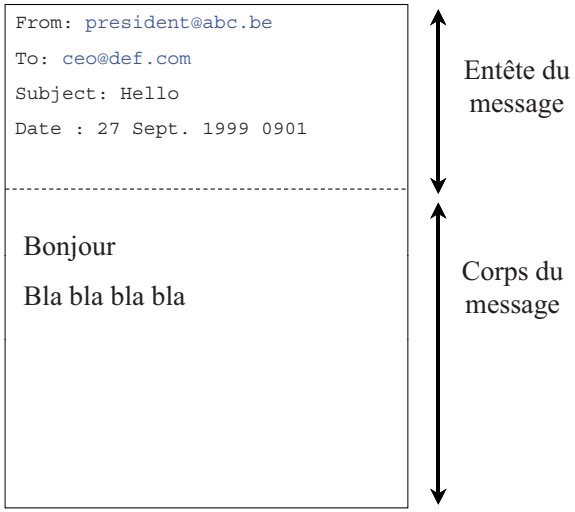
  Ex

  titi@yahoo.fr

- Le DNS permet de déterminer les serveurs de courrier d'un domaine (enregistrement MX)
- Les nom utilisables pour les boites aux lettres sont quelconques.

  forme usuelle: prénom.nom

Le format des messages (version de base)



Le format des messages (version de base)

Format de l'entête

- composée de caractères ASCII
- seuls les caractères US-ASCII (7bits) sont autorisés
- Minimum trois lignes terminées par <CRLF>

  From: expediteur@domaine

  To: destinataire@domaine

  Date: <date de creation du message>
- Champs optionnels (terminés par <CRLF>)

  Subject: sujet du message

  cc: copie@domaine

  Message-ID: <numero@domaine>

  Received: information sur le chemin suivi par le message

  In-Reply-To: <message-ID>
- Entête se termine par ligne vide (<CRLF>)

Le format des messages (version de base)

Received: from limbes.fundp.ac.be (limbes.fundp.ac.be [138.48.4.10])
by leibniz.info.fundp.ac.be (8.9.1/8.9.1) with ESMTP id QAA16962
for <Olivier.Bonaventure@info.fundp.ac.be>; Fri, 17 Sep 1999 16:51:20 +0200 (MET DST)

Received: from mailhub.fokus.gmd.de (mailhub.fokus.gmd.de [193.174.154.14])
by limbes.fundp.ac.be (8.9.1/8.9.1) with ESMTP id QAA05665
for <Olivier.Bonaventure@info.fundp.ac.be>; Fri, 17 Sep 1999 16:51:16 +0200 (MET DST)

Message-ID: <37E24CEF.7E38F9E3@fokus.gmd.de>

Date: Fri, 17 Sep 1999 16:15:11 +0200

From: Eckhart Koerner <koerner@fokus.gmd.de>

To: Olivier.Bonaventure@info.fundp.ac.be

Subject: Re: Nouveaux coordonnes

In-Reply-To: <3768C3AB.11C52680@info.fundp.ac.be>

Bonjour

Bla, bla, bla...

Le format des messages (version de base)

A l'origine

- message composé de lignes ASCII 7 bits terminées par <CRLF>
- pas de limite a priori sur la taille des lignes ou des messages, mais de nombreux serveurs en imposent

Problème

- Comment dans un émail transmettre des caractères accentués ?
- autre chose que du texte ASCII?
  - audio
  - vidéo
  - programmes...

Solution

- Redéfinition du format des messages (le format MIME)

MIME
Multipurpose Internet Mail Extensions

objectifs

- Rester le + possible compatible avec d'anciens serveurs d'email
- Supporter des textes dans une autre langue que l'anglais
  - Nécessité d'identifier le type d'encodage des caractères puisque ASCII 7bits ne suffit plus
- Pouvoir transférer autre chose que du texte
  - Nécessité d'identifier les différents types de contenu
- Pouvoir créer des messages composés de plusieurs parties
  - Fichiers attachés

Solution choisie

- ajouter de nouveaux champs optionnels dans l'entête du message
- ajouter lorsque c'est nécessaire certains éléments dans le corps du message

L'entête MIME

En plus des champs de l'entête de la version de base des champs supplémentaires en été rajoutés

- MIME-Version: indique la version de MIME utilisée pour créer le message
- Content-Description: texte ASCII (commentaire) décrivant le contenu
- Content-Type: indique la nature du message
- Content-Transfer-Encoding: indique la façon dont le message a été encodé
- Content-Id: identificateur unique pour le contenu

L'entête MIME

Content-Type est composé de deux partie

1 Type du contenu

- Texte :text/plain , text/html
- Image: image/gif, image/jpeg
- Son: audio/basic
- Vidéo: vidéo/mpeg, vidéo/quicktime
- Application: application/octet-stream, application/Postscript
- multipart/alternative: message composé de plusieurs parties représentant le même contenu dans des formats différents (ex: deux formats d'image)
- multipart/mixed message composé de plusieurs contenus différents

  exemple : partie texte et fichier attaché

L'entête MIME

2 charset:

paramètre indiquant le code de caractères utilisé pour encoder le message type text)

- charset=us-ascii
  - Norme définie fin des années 1960s aux USA
  - Caractères ASCII 7bits, défaut
- charset=iso-8859-1
  - Code de caractère Latin1 pour Europe défini par l'ISO
  - 8 bits pour chaque caractère
  - Les caractères 0-127 sont équivalents à US-ASCII
- charset=unicode
  - Code de caractère universel, supporte toutes les langues
  - 16 bits pour chaque caractère
  - Partiellement compatible avec ISO-8859-1

L'entête MIME

Problèmes à résoudre

- de nombreux serveurs d'email n'accepte que les caractères ASCII 7 bits
- comment transmettre à travers ces serveurs
  - Des caractères accentués (ISO8859-1 ou Unicode)
  - Des fichiers binaires

Solution

encodage Base64

- utilise les caractères ASCII A...Z,a...z,0...9, "+" et "/«
- A=0, B=1, C=2, ... +=62 et /=63
- Un caractère est utilisé pour encoder 6 bits du message initial
- groupe de 24 bits -> encodé sur 4 caractères ASCII
- Rôle du caractère "=" pour les groupes < 24 bits

L'entête MIME

Dans le type Multipart/mixed comment identifier les différentes parties d'un message ?

- choisir une chaîne de caractères "spéciale" qui servira de délimiteur
- Le délimiteur ne doit pas apparaître lui-même dans le message
- annoncer ce délimiteur dans l'entête

Date: Mon, 20 Sep 1999 16:33:16 +0200
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Test
MIME-Version: 1.0
Content-Type: multipart/mixed; **boundary="simple boundary"**
preamble, to be ignored
--simple boundary
Content-Type: text/plain; charset=us-ascii
partie 1
--simple boundary
Content-Type: text/plain; charset=us-ascii
partie 2
--simple boundary

L'entête MIME

Exemple 1

Received: from loriot.info.fundp.ac.be (loriot.info.fundp.ac.be [138.48.32.96])
by leibniz.info.fundp.ac.be (8.9.1/8.9.1) with SMTP id QAA19679;
Mon, 20 Sep 1999 16:37:25 +0200 (MET DST)
Message-Id: <3.0.5.32.19990920163316.00866340@info.fundp.ac.be>
Date: Mon, 20 Sep 1999 16:33:16 +0200
To: pers-aca, pers-sci
From: Gysele HENRARD <ghe@info.fundp.ac.be>
Subject: listes
Mime-Version: 1.0
Content-Type: multipart/mixed; boundary="==========_937830796==_"
--==========_937830796==_
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
Bonjour,
Bla bla bla
--==========_937830796==_
Content-Type: application/octet-stream; name="1M_99_00.xls";
--==========_937830796==_

L'entête MIME

Message-Id: <37E74BA9.18C8A6B5@kt.agh.edu.pl>
Date: Tue, 21 Sep 1999 11:11:05 +0200
From: Andrzej Pach <pach@kt.agh.edu.pl>
Mime-Version: 1.0
To: xtp-relay@cs.concordia.ca, sigmedia@bellcore.com, tccc@ieee.org
Subject: Broadband Access Conference (BAC'99)
Content-Type: multipart/alternative; boundary="-------D109973074C44AEF3C30671C"
---------D109973074C44AEF3C30671C
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit
Bla bla bla
---------D109973074C44AEF3C30671C
Content-Type: text/html; charset=iso-8859-1
Content-Transfer-Encoding: 8bit
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
...
</html>
----------D109973074C44AEF3C30671C--

POP

Post Office Protocol

Protocole POP 3

Objectif

- Permettre la récupération de messages email à distance avec authentification des utilisateurs

Fonctionnement

- utilise le service TCP via le port par défaut: **110**
- le serveur POP ne conserve normalement que les messages nouvellement arrivés qui n'ont pas encore été transférés vers la machine de l'utilisateur
- les messages sont archivés sur la machine de l'utilisateur.
- La connexion ce fait en trois phase

  AUTORISATION , TRANSACTION, MISE A JOUR

SYNTAXE DE COMMANDES POP

Valides dans l'état AUTORISATION

- USER nom : nom d'utilisateur
- PASS mot-de-passe : mot de passe utilisateur
- QUIT: termine la lecture des courriers

Valides dans l'état TRANSACTION

- STAT : donne le nombre de message dans la boite à lettre et la taille totale en octets

  ex: `C : STAT`
  `S : +OK 2 420`
- LIST < SP> <msg> : donne les information sur le numéro et la t'aille de msg (optionnelle)

Ex 1:

```
C : LIST
S : +OK 2 messages (320 octets)
S : 1 220
S : 2 200
S : .
```

Ex 2:

```
C : LIST 2
S : + OK 2 200
```

SYNTAXE DE COMMANDES POP

- RETR < SP> <msg> : pour lire le message msg (obligatoire)

  ex

```
C : RETR 1
S : +OK 220 octets
S : <le serveur POP3 envoie le message entier ici>
S : .
```

- DELE < SP> <msg> : effacer le message msg

  ex

```
C : DELE 1
S : +OK message 1 deleted
```

- NOOP : pour teste le serveur POP

  ex

```
C : NOOP
S : +OK
```

- RSET abandonner puis restaurer la connexion

  ex

```
C : RSET
S : +OK maildrop has 2 messages (420 octets)
```


SYNTAXE DE COMMANDES POP

- QUIT: termine la lecture des courriers avec mise à jours sur le serveur

  ex

```
C : QUIT
S : +OK dewey POP3 server signing off (2
messages left)
```

Commandes POP3 optionnelles :

Valides dans l'état TRANSACTION

- TOP < SP> <msg> < SP> <n>: afficher l'entêtes et les n (obligatoire) premières lignes du message msg (obligatoire)

  ex

```
C : TOP 1 10
S : +OK
S : <entêtes du message et les 10 premières
    lignes du corps du message>
S : .
```

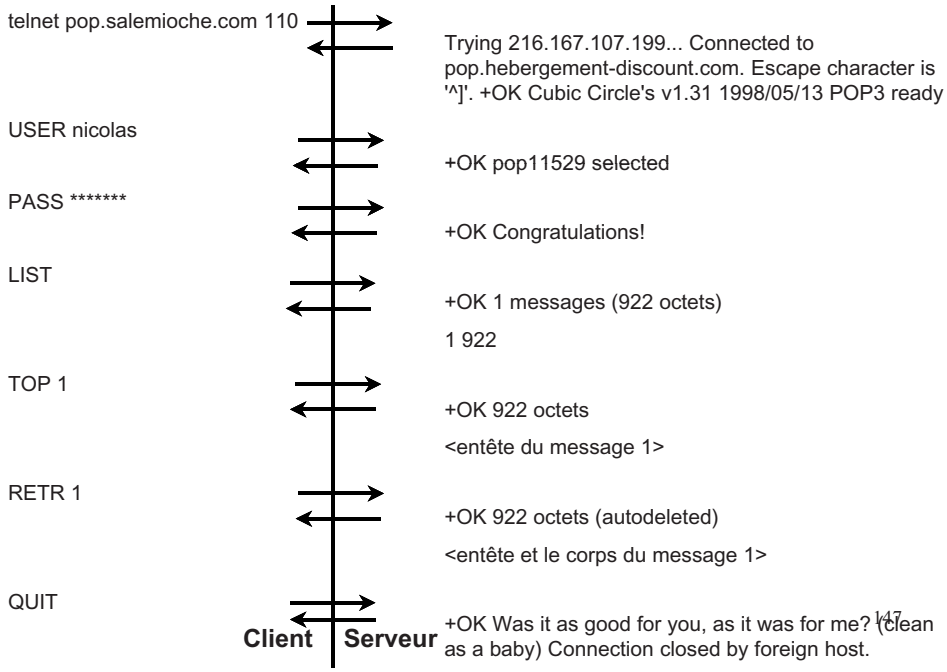
- UIDL < SP> <msg> afficher l'identificateur du message msg (optionnelle)

  ex

```
C : UIDL
S : +OK
S : 1 whqtsw0OOWBw418f9t5JxYwZ
S : 2 QhdPYR:00WBw1Ph7x7
```

```
C : UIDL
S : +OK 2 QhdPYR:00WBw1Ph7x7
```


Scénario de lecture d'un message



IMAP4

Internet Message Access Protocol

Protocole IMAP4

- utilise le service TCP via le port par défaut : **143**
- les messages sont archivés sur le serveur IMAP
- le fait que les messages soient archivés sur le serveur fait que l'utilisateur peut accéder à tous ses messges depuis n'importe où sur le réseau et que l'administrateur peut facilement faire des copies de sauvegarde..
- le protocole comporte des commandes pour gérer des boites aux lettres, gérer des messages, rechercher dans le contenu des messages et transférer de manière sélective des messages vers l'utilisateur.
- intègre l'extension MIME, permettant, par exemple, àl'utilisateur de ne demander le transfert que d'une partie d'un message qui aurait de volumineuses annexes.
- particulièrement bien adapté à l'accès à travers des connexions lentes.

SYNTAXE DE COMMANDES IMAP

- LOGIN<SP> <nom utilisateur> <SP> <mots de passe> : identifie les clients pour le serveur

  ex C: a001 LOGIN SMITH SESAME
  S: a001 OK LOGIN completed
- SELECT <SP> <nom de la boite> : sélectionne une boîte aux lettres

  ex C: A142 SELECT INBOX
  S: * 172 EXISTS
  S: * 1 RECENT
  S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
  S: * OK [PERMANENTFLAGS (\Deleted \Seen *)] Limited
  S: A142 OK [READ-WRITE] SELECT completed
- CREATE <SP> <nom de la boite> : crée une boîte aux lettres avec un nom **ex** C: A003 CREATE owatagusiam/
  S: A003 OK CREATE completed
- DELETE <SP> <nom de la boite> : efface la boite aux lettres qui est passée en argument.

  ex C: A683 DELETE toto
  S: A683 OK DELETE completed
- LOGOUT :arrête la connexion avec le serveur

SYNTAXE DE COMMANDES IMAP

- RENAME <SP> <ancien nom > <SP> < nouveau nom > : change le nom d'une boîte aux lettres

  ex C: A684 RENAME toto bobo
  S: A684 OK RENAME Completed
- LIST <SP> < référence > <SP> < nom de la boite > : donne les noms disponibles pour le client

  ex Z432 LIST "" *
  S: * LIST () "." INBOX
  S: * LIST () "." INBOX.bar
  S: Z432 OK LIST completed
- STATUS <SP> < nom de la boite > <SP> < référence > : donne les informations sur une boite à lettre.

  ex C: A042 STATUS blurdybloop (UIDNEXT MESSAGES)
  S: * STATUS blurdybloop (MESSAGES 231 UIDNEXT 44292)
  S: A042 OK STATUS completed
- CLOSE : ferme la boite mail sélectionner en effectuant une mise à jour

  ex C: A341 CLOSE
  S: A341 OK CLOSE completed

SYNTAXE DE COMMANDES IMAP

- EXPUNGE: enlève de façon définitive de la boîte aux lettres sélectionnée tous les messages qui ont le marqueur établi à \Delete.

  ex C: A202 EXPUNGE
  S: * 3 EXPUNGE
  S: * 3 EXPUNGE
  S: A202 OK EXPUNGE completed
- SEARCH <SP> < critère> : commande qui permet de rechercher des messages selon des critères spécifiques.

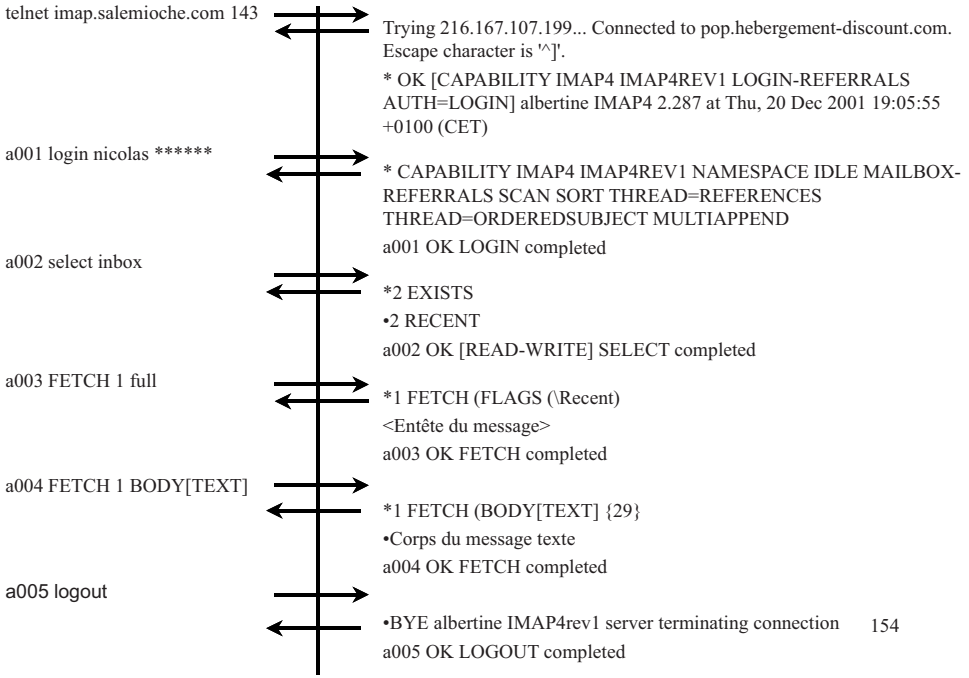
  ex C: A282 SEARCH FLAGGED SINCE 1-Feb-1994 NOT FROM « Smith"
  S: * SEARCH 2 84 882
  S: A282 OK SEARCH completed
  S: A282 OK SEARCH completed
- FETCH <SP> < num_msg> <SP> < critère>: récupère, dans la boîte aux lettres, les données associées à un message

  ex C: A654 FETCH 2 BODY[TXT)
  S: * 2 FETCH ....
  S: corps de message
  S: A654 OK FETCH completed

CODES REPONSE IMAP

- tag * OK : la commande c'est bien déroulée.
- tag * NO : échec de la commande.
- BAD : erreur de protocole.
- PREAUTH : message d'accueil, indique des fois qu'il n'est pas nécessaire de se loger.
- BYE : le serveur va fermer sa session.

Scénario de lecture d'un message



Comparaison POP IMAP

	POP	IMAP
Lieu de stockage du courrier	Site du client de messagerie	Sur le serveur de messagerie
Archives de courriers	Généralement Sur le site client	Sur le serveur
Contrôle des messages ramenés du serveur	Tous les message sont transférés	Possibilité de récupérer des parties d'un message
Accès au courrier de différents client	Très difficile voir impossible	Prévu pour
connexionx	Se connecte périodiquement au serveur, récupère le courriers reçus et se déconnecte	Maintient une connexion sur le serveur tant que le client est opérationnel

FTP

File Transfer Protocol

FTP

Rapide Historique :

- 1971 : Première version du protocole définit par le M.I.T.
- 1973 : Première documentation officielle du protocole FTP.
- 1975 : Evolution de FTP pour pouvoir fonctionner au dessus de TCP (jusqu'alors FTP utilisé NCP).
- 1982 : Finalisation de la définition du rôle de FTP : « *Le File Transfert Protocol* est désormais définit comme un protocole de transfert de fichier entre des hôtes d'un ARPANET, afin de profiter de l'utilisation d'une capacité de stockage de données distante »

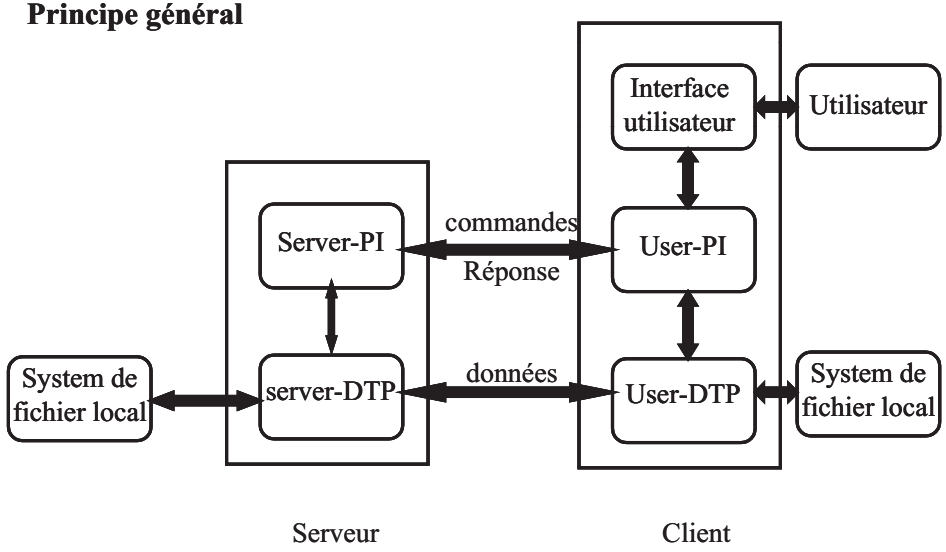
FTP

Caractéristiques

- utilise le service TCP via le port par défaut : 21
- Fonctionne en mode client\serveur
  - Le client FTP est la machine de l'utilisateur.
  - Le serveur est la machine sur laquelle est placée le système de fichier.
- Prévu pour être exploité par l'intermédiaire de clients dédiés, mais disponible à une exploitation directe (via un client telnet).
- transfert des fichier sous deux format :
  - En binaire (raw data sur le Mac), destiné aux exécutables (logiciels), aux formats spéciaux (images, son, films)...
  - En ASCII (text) pour les fichiers de données, les fichiers Postscript ...

FTP

Principe général



FTP

Établissement d'une connexion FTP:

Ce fait en deux étapes

Étape 1:

- établir un canal de control sur le port 21

Ex telnet 194.210.10.50 21

Étape 2:

- établir un canal pour la transmission de données

Établissement d'un canal de transmission de données

Peux être établie de de manière :

- **cas par défaut:** le client choisie le port en utilisant la commande **PORT**
- **cas passif:** demander au serveur de choisir le port en utilisant la commande **PASV**

FTP

Les commandes FTP

Commandes de gestion du canal de contrôle :

- **USER <username>** Définir l'utilisateur de la session en cours.
- **PASS <password>** Envoyer le mot de passe associé à l'utilisateur.
- **REIN** Réinitialiser la session en cours, sans perdre la connexion.
- **QUIT** Termine la connexion en cours.

Commandes du paramétrage des transferts :

- **PORT <IP1>,<IP2>,<IP3>,<IP4>,<PORT1>,<PORT2>** Configurer la cible d'émission pour le canal de contrôle avec : IP, les 4 octets de l'adresse IP de la cible ; Port, deux nombres entre 0 et 255 qui donne le port sur 16bits.
- **PASV** Positionne le serveur en mode passif. Il retourne l'adresse et le port sur lequel il attend une connexion du client.
- **TYPE <param>** Définit le type d'échange réalisé via la socket de donnée :
  <param> = **A** ASCII, **E** EBCDIC, **I** Image.
- **MODE <param>** Détermine le monde d'encodage des données.
  <param> = **S** Flux, **B** Block, **C** Compressé.

FTP

Commandes de transfert FTP

- **RETR <filename>** Déclenche la transmission par le serveur du fichier <filename> sur le canal de données.
- **STOR <filename>** Déclenche la réception d'un fichier qui sera enregistré sur le disque sous le nom <filename>. Si un fichier avec le même nom existe déjà il est remplacé par un nouveau avec les données transmisses.
- **APPE <filename>** Déclenche la réception d'un fichier qui sera enregistré sur le disque sous le nom <filename>. Si un fichier avec le même nom existe déjà, les nouvelles données lui sont concaténée.
- **REST <offset>** Redémarrage en cas d'échec d'un transfert précédent. L'offset précise le numéro du dernier octet reçu.
- **ABOR** : abandon d'un transfert en cours.

FTP

Commandes de gestion du système de fichier distant et du serveur FTP:

- **PWD** : impression du répertoire courant.
- **LIST** : catalogue du répertoire courant (canal donnée).
- **CWD <repname>** : changement de répertoire courant pour <repname>.
- **MKD <repname>** : création du nouveau répertoire <repname>.
- **RMD <repname>** : suppression du répertoire <repname>.
- **DELE <filename>** : suppression du fichier <filename>.
- **RNFR <filename1>** : définit le nom actuel d'un fichier à renommer.
- **RNTO <filename2>** : définit le nouveau nom d'un fichier à renommer.
- **STAT :** status courant de la session FTP.
- **STAT <repname> :** équivalent à LIST mais réponse sur le canal de contrôle.
- **HELP** : affiche l'aide sur les opérations du site.
- **SYST :** permet de connaître le système sous lequel tourne le serveur
- **NOOP** : no operation.

FTP

Les codes de retour :

- 1yz : réponse positive prélimiaire
- 2yz : réponse positive définitive
- 3yz : réponse positive intermédiaire
- 4yz : réponse négative transitoire
- 5yz : réponse négative définitive

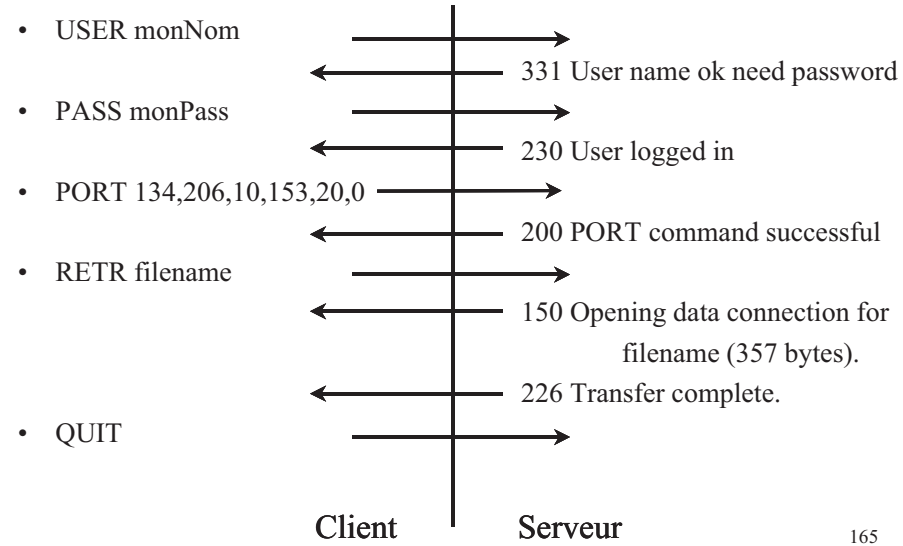
- •x0z : erreur de syntaxe
- •x1z : réponse contenant des informations
- •x2z : réponse vis à vis de la connexion
- •x3z : identification et authentification
- •x4z : non spécifié
- •x5z : système de fichier

La troisième digit raffine la description d'erreur, exemple :

- 501 Erreur de syntaxe, commande non connue.
- 502 Erreur de syntaxe dans les paramètres/arguments.
- 503 Erreur de syntaxe, commande non implémentée.
- 504 Erreur de syntaxe, mauvaise séquence de commandes.
- 505 Erreur de syntaxe, commande non implémentée pour ce param.

FTP

EX 1: réception d'un fichier



FTP

EX 2: émission d'un fichier