

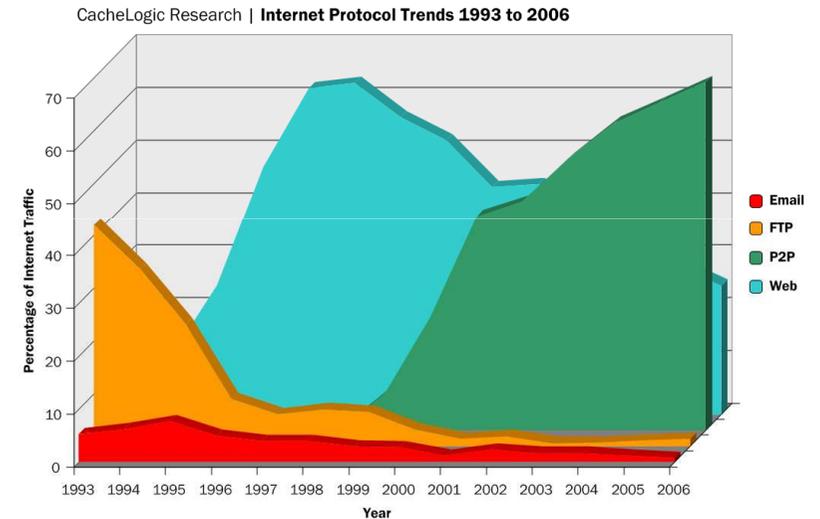
Applications TCP/IP

3. La couche Application

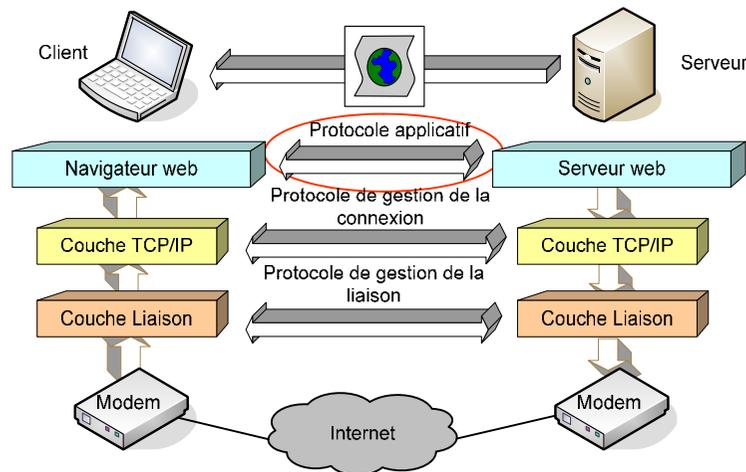
- ❑ Protocoles applicatifs
- ❑ Service DNS
- ❑ Web et HTTP
- ❑ Messagerie (SMTP, POP, IMAP)
- ❑ Transfert de fichiers (FTP, P2P)
- ❑ Voix sur IP
- ❑ Vidéo sur IP



Protocoles applicatifs Répartition du trafic sur Internet



Protocoles applicatifs Localisation / modèle OSI



Protocoles applicatifs Applications et services

- ❑ La couche application comprend deux formes de programmes ou processus logiciels permettant d'accéder au réseau :
 - Les applications orientées réseau qui permettent aux utilisateurs de communiquer (clients de messagerie, navigateurs Web...)
 - Les services de couche application qui établissent l'interface avec le réseau et préparent les données à transférer (transfert de fichiers, mise en file d'attente de tâches d'impression réseau...)

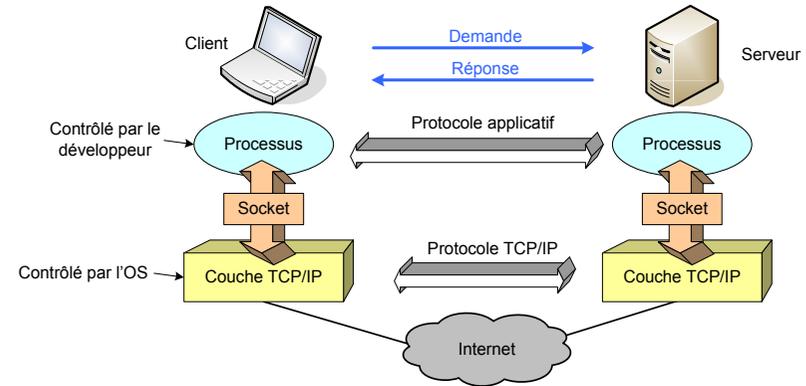
Nom de l'image	Nom de l'utilisateur	P...	Util. mé...	
wcescomm.exe	lohier	00	4 684 Ko	Application
explorer.exe	lohier	01	30 436 Ko	Application réseau
OUTLOOK.EXE	lohier	00	14 076 Ko	Application réseau
svchost.exe	SERVICE RÉSEAU	00	4 888 Ko	Service réseau
svchost.exe	SYSTEM	00	5 208 Ko	Service réseau
lsass.exe	SYSTEM	00	7 196 Ko	Plusieurs occurrences d'un même programme dans leurs propres processus
services.exe	SYSTEM	03	3 528 Ko	Plusieurs occurrences d'un même programme dans leurs propres processus
winlogon.exe	SYSTEM	00	3 336 Ko	Plusieurs occurrences d'un même programme dans leurs propres processus
csrss.exe	SYSTEM	00	5 248 Ko	Plusieurs occurrences d'un même programme dans leurs propres processus
sched.exe	SYSTEM	00	532 Ko	OS
scardsvr.exe	SERVICE LOCAL	00	2 564 Ko	OS
smss.exe	SYSTEM	00	392 Ko	OS
spoolsv.exe	SYSTEM	00	6 144 Ko	OS
NMIndexStoreSvr...	lohier	00	20 076 Ko	Service
ISUSPM.exe	lohier	00	3 508 Ko	Service
svchost.exe	SERVICE LOCAL	00	4 128 Ko	Service

Protocoles applicatifs Rôle du protocole

- ❑ Les protocoles sont les règles de dialogue qui permettent de mettre en œuvre une application entre un client et un serveur.
- ❑ Par exemple, l'application web est constitué :
 - de langages de description de pages statiques ou dynamiques : html, xhtml...
 - de navigateurs web : Firefox, Opera, IE...
 - de serveurs web : Apache, IIS...
 - de protocoles de niveau applicatif : HTTP...
- ❑ Un protocole applicatif (HTTP, SMTP, FTP...) définit :
 - la séquence de messages échangés entre client et serveur (quand et comment les processus client et serveur doivent envoyer des demandes ou des réponses) ;
 - le type de messages échangés (message de demande, de réponse, de confirmation...) ;
 - la syntaxe adoptée par les différents types de messages (les différents champs et leurs délimitations) ;
 - La sémantique des différents champs (le sens des informations qu'ils contiennent).

Protocoles applicatifs Côté développeur

- ❑ Une application en réseau implique :
 - 2 processus engagés sur 2 hôtes reliés par un réseau ;
 - des interfaces de connexion (*socket*) permettant de faire passer les messages vers la couche de transport ;
 - une infrastructure de transport (TCP ou UDP) permettant de transporter les messages lorsque la connexion est établie.



Protocoles applicatifs Besoins des applications

- ❑ L'Internet n'est pas conçu au départ pour offrir des garanties en termes de QoS (Qualité de Service)
- ❑ Certaines applications ont besoin de QoS : débit minimum, délai maximum, taux de pertes...
- ❑ Les protocoles de niveau inférieur (TCP, IP, RTP...) devront donc être choisis en fonction des ces besoins : TCP pour retransmettre des paquets perdus, RTP pour garantir des délais...

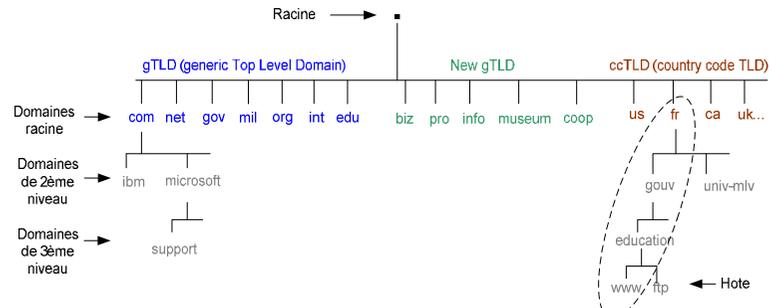
Application	Perte de données	Débit	Contrainte de temps
Transfert de fichier	Interdite	Flexible	Non
e-mail	Interdite	Flexible	Non
web	Interdite	Flexible	Non
Fichiers audio/vidéo enregistrés	Acceptable	Flexible	Non
Fichiers audio/vidéo en temps réel	Acceptable	Audio : 10 kbit/s à 1 Mbit/s Vidéo : 100 kbit/s à 5 Mbit/s	Oui - forte
Messagerie instantanée	Interdite	Flexible	Oui - faible
P2P	Interdite	Flexible	Non

Le service DNS RFC 1034-1035 Principe

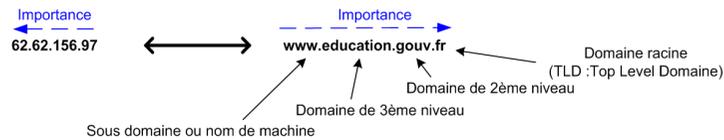
- ❑ Pour simplifier l'identification, un service de résolution permettant d'utiliser des noms symboliques de machines à la place des adresses IP est utilisé sur tous les réseaux TCP/IP.
- ❑ Pour circuler sur Internet, un paquet IP doit contenir l'adresse IP de destination et non le nom associé. Il faut donc interroger ce service de résolution qui contient toutes les correspondances.
- ❑ La méthode la plus simple passe par l'utilisation d'un fichier ASCII (fichier `/etc/hosts` sous UNIX par exemple) sur la machine émettrice qui comprend des noms et les adresses IP correspondantes (méthode impossible pour de gros réseaux).
- ❑ Pour Internet, la gestion des noms est centralisée sur des machines spécifiques (les serveurs de noms) à l'aide d'un service permettant une organisation hiérarchisée : le DNS (*Domain Name Service*).
- ❑ Ce service de nom de domaine travaille suivant une organisation arborescente en divisant le réseau global en un ensemble de domaines primaires, secondaires...

Le service DNS Arborescence

- Dans la structure arborescente sont définis des domaines racine (appelés TLD, pour *Top Level Domains*), rattachés à un noeud racine représenté par un point.



- Un nom complet d'hôte ou FQDN (*Fully Qualified Domain Name*) est constitué des domaines successifs séparés par un point.



Master TTT

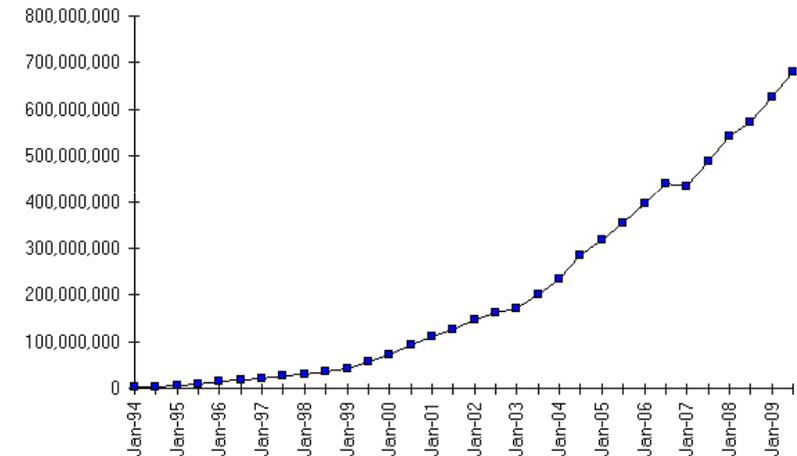
La couche Application

S. Lohier

9

Le service DNS Statistiques 1

Internet Domain Survey Host Count



Source: Internet Systems Consortium (www.isc.org)

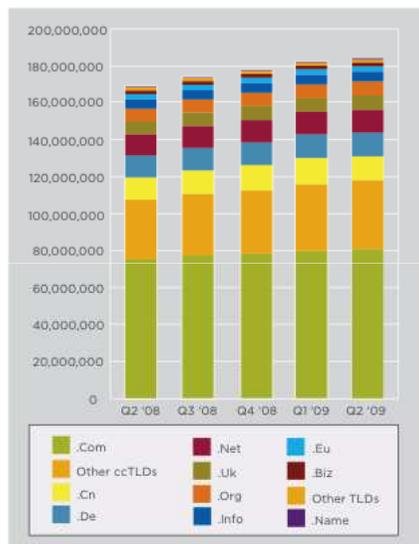
Master TTT

La couche Application

S. Lohier

10

Le service DNS Statistiques 2



From these 184 million total registrations :

- gTLDs account for 109.6 million registered domains.
- ccTLDs have a registration base of 74.4 million addresses.

Source: Zooknic, July 2009; VeriSign, July 2009

Master TTT

La couche Application

S. Lohier

11

Le service DNS Statistiques 3

<http://www.domaintools.com/internet-statistics/>

Welcome to Domain Tools' daily domain statistics page. Our stats show how many domains are currently registered and how many domains used to be registered but are now deleted.

Active	Deleted	Daily Changes (last 24hrs)			TLD
		New	Expired	Transferred	
82,515,300	306,343,962	49,296	54,450	60,337	.Com
12,484,242	32,305,388	7,308	7,840	8,764	.Net
7,848,737	20,344,317	4,854	4,282	5,030	.Org
5,287,786	8,828,015	5,511	7,083	5,804	.Info
2,034,625	1,904,917	1,071	1,607	1,596	.Biz
1,651,466	1,499,602	827	777	861	.Us
111,953,990	371,887,778	68,867	76,039	82,392	Total

Last Updated : 2009-10-05

Master TTT

La couche Application

S. Lohier

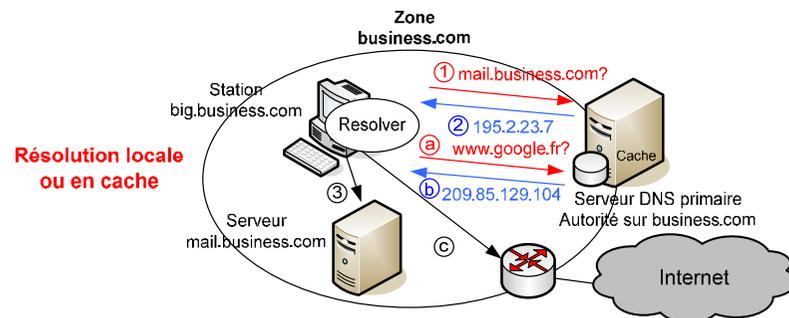
12

Other quick facts about the domain name industry:

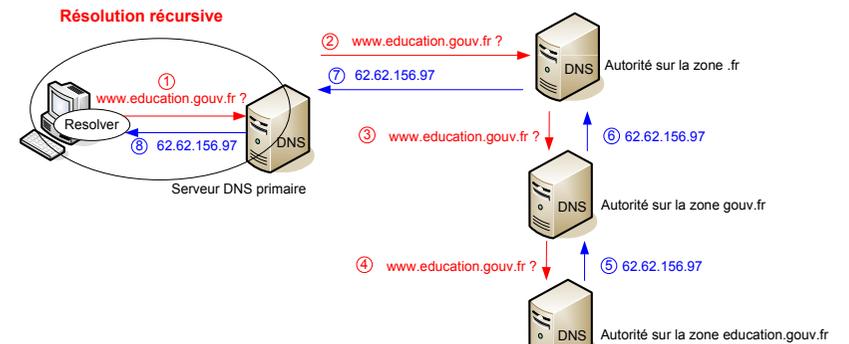
- ❑ 20.9 million new domain names registered in the first six months of 2009.
- ❑ .COM & .NET account for 50% of all domains registered worldwide.
- ❑ The top 10 country code domain extensions comprise 66 per cent of all ccTLD registrations.
- ❑ The top 3 ccTLDs .DE, .CN & .UK represent 45 percent of all registered ccTLD names.
- ❑ .AR (Argentina), .RU (Russia) and .BR (Brazil) are currently the fastest growing ccTLDs.



- ❑ Chaque serveur de noms DNS gère une ou plusieurs zones du réseau.
- ❑ Chacune des zones possède au moins un serveur de noms ayant la connaissance complète des adresses des machines de la zone.
- ❑ Côté client, chaque machine possède au moins l'adresse d'un serveur DNS (serveur primaire) et éventuellement l'adresse d'un second (serveur secondaire).
- ❑ Lorsqu'une application (navigateur, client FTP, client mail...) à besoin de résoudre un nom symbolique en une adresse réseau, elle envoie une requête au **résolveur local** (processus spécifique sur la machine client pour l'application DNS).
- ❑ Le résolveur local transmet au serveur de nom de la zone locale (**serveur primaire**).
- ❑ Si le nom est local ou en cache, le serveur primaire renvoie directement l'adresse IP demandée.



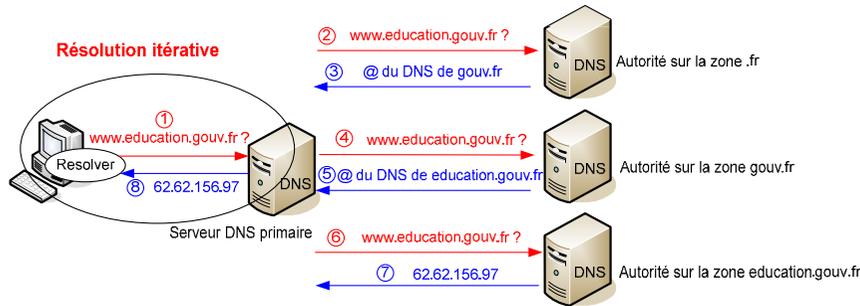
- ❑ Si le nom ne peut être résolu localement (zone distante ou nom absent du cache), le serveur primaire transmet à un serveur distant ayant autorité sur le domaine racine concerné (.fr, .com, .net...).
- ❑ La requête est ensuite relayée de manière récursive jusqu'à atteindre le serveur DNS ayant autorité sur la zone demandée. L'adresse IP de la machine est alors renvoyée.
- ❑ Si le serveur primaire ne connaît pas l'adresse ayant autorité sur le domaine racine, il doit au préalable interroger un serveur racine.
- ❑ Les serveurs racine connaissent au moins les serveurs de noms pouvant résoudre le premier niveau (.fr, .com, .net...)



Le service DNS

Résolution itérative

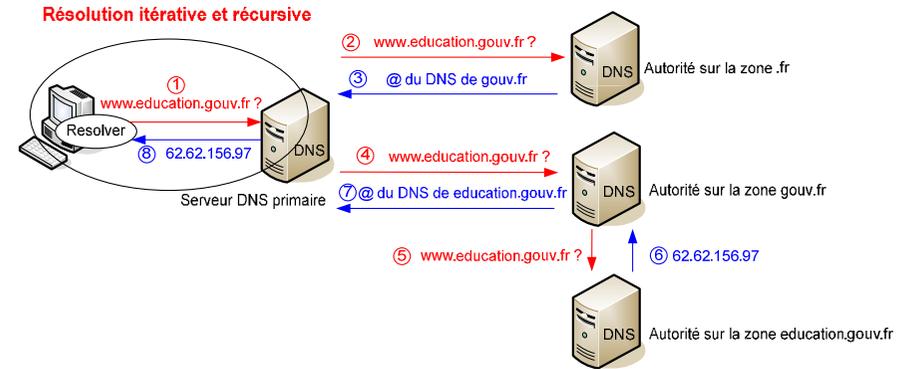
- Le protocole DNS autorise également l'usage de recherches itératives.
- Lorsqu'un serveur DNS n'est pas en mesure de relayer la requête de manière récursive, il transmet au serveur primaire l'adresse du prochain serveur.
- Dans l'exemple suivant, toutes les recherches sont de type itératives, c'est le serveur primaire qui est chargé de toutes les requêtes.



Le service DNS

Résolution itérative et récursive

- Dans la plupart des cas, le serveur de nom racine, fortement sollicité, ne relaie pas les requêtes de manière récursive.
- Seule la requête vers le serveur de nom racine est itérative, les autres sont récursives.



Le service DNS

Enregistrements DNS

- La base de données des serveurs de noms est constituée "d'enregistrements de ressources" ou "Resource Records" (RRs).
- Ces enregistrements sont répartis en classes. La seule classe d'enregistrement usuellement employée est la classe Internet (IN).
- À chaque nom de domaine est donc associé un RR. Il existe plusieurs types de RR :
 - A : Adress, correspondance nom -> adresse IP (le plus usuel)
 - NS : Name Server, serveur(s) de nom pour ce domaine
 - CNAME : Canonical NAME, nom d'origine (des alias peuvent exister)
 - SOA : Start Of Authority, serveur(s) faisant autorité sur la zone
 - PTR : PoinTeR, correspondance adresse IP -> nom (résolution inverse)
 - MX : Mail eXchange, indique le serveur de messagerie.
- Exemple de RR :

FQDN (nom de domaine)	TTL	Type	Classe	Rdata
www.education.gouv.fr	3600	A	IN	62.62.156.97

Le service DNS

Format des messages DNS

- Pour lire les différents RR, le protocole associé utilise le même format de message pour les demandes et les réponses.

Identificateur	Drapeaux
Nombre de questions	Nombre de réponses
Nombre d'enregistrements « autorité »	Nombre d'enregistrements « information supplémentaire »
Questions	
Réponses	
Autorité	
Informations supplémentaires	

- Le champ « **identificateur** » est positionné par le client et permet de faire correspondre les réponses aux demandes.
- Les **drapeaux** donnent des indications supplémentaires sur le message (demande ou réponse, demande standard ou inverse, erreur de nom, récursivité disponible...).
- Les **4 valeurs suivantes** de 16 bits précisent le nombre de RR dans les 4 champs de longueurs variables qui terminent le message.
- Les champs « **questions** » et « **réponses** » précisent le type et le contenu de la demande ou de la réponse (résolution IP, nom demandé, adresse retournée...).
- Les champs « **autorité** » et « **informations supplémentaires** » donnent éventuellement dans une réponse des indications sur les serveurs DNS ayant participé à la résolution.

- Les figures suivantes montrent un exemple de dialogue DNS relevé à l'aide d'un analyseur de protocoles.
- Pour la requête, seul le champ « queries » comporte un enregistrement, il contient le nom demandé.

```

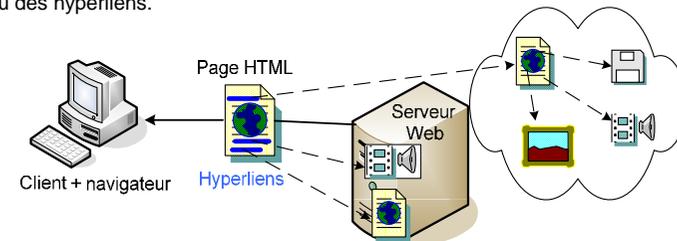
# Frame 1 (81 bytes on wire, 81 bytes captured)
# Ethernet II, Src: DellComp_e0:d2:22 (00:08:74:e0:d2:22), Dst: D-Link_aa:61:fd (00:50:ba:aa:61:fd)
# Internet Protocol, Src: 192.168.0.4 (192.168.0.4), Dst: 212.198.2.51 (212.198.2.51)
# User Datagram Protocol, Src Port: 1084 (1084), Dst Port: domain (53)
# Domain Name System (query)
  [Response In: 2]
  Transaction ID: 0x0080
  Flags: 0x0100 (Standard query)
    0... .. = Response: Message is a query
    .000 0... .. = Opcode: Standard query (0)
    ... .. = Truncated: Message is not truncated
    ... .. = Recursion desired: Do query recursively
    ... .. = Z: reserved (0)
    ... .. = Non-authenticated data OK: Non-authenticated data is unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  Queries
    www.education.gouv.fr: type A, class IN
      Name: www.education.gouv.fr
      Type: A (Host address)
      Class: IN (0x0001)
  
```

- Le message de réponse comporte les quatre champs avec un enregistrement pour le champ « queries » et 2 enregistrements pour chacun les 3 autres champs.
- L'adresse IP demandée est donnée dans l'un des 2 enregistrements de réponse.

```

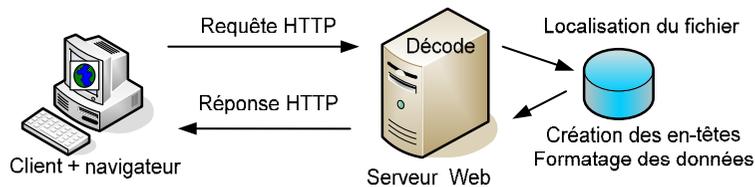
# Frame 2 (219 bytes on wire, 219 bytes captured)
# Ethernet II, Src: D-Link_aa:61:fd (00:50:ba:aa:61:fd), Dst: dellcomp_e0:d2:22 (00:08:74:e0:d2:22)
# Internet Protocol, Src: 212.198.2.51 (212.198.2.51), Dst: 192.168.0.4 (192.168.0.4)
# User Datagram Protocol, Src Port: domain (53), Dst Port: 1084 (1084)
# Domain Name System (response)
  Transaction ID: 0x0080
  Flags: 0x8180 (Standard query response, No error)
  Questions: 1
  Answer RRs: 2
  Authority RRs: 2
  Additional RRs: 2
  Queries
    www.education.gouv.fr: type A, class IN
  Answers
    www.education.gouv.fr: type CNAME, class IN, cname crepidula.gasteropode.jmsp.net
    crepidula.gasteropode.jmsp.net: type A, class IN, addr 62.62.156.197
  Authoritative nameservers
    jmsp.net: type NS, class IN, ns ullinn.fast.jmsp.net
    jmsp.net: type NS, class IN, ns helgi.fast.jmsp.net
  Additional records
    helgi.fast.jmsp.net: type A, class IN, addr 212.23.164.29
    ullinn.fast.jmsp.net: type A, class IN, addr 194.153.92.13
  
```

- HTTP (*HyperText Transmission Protocol*) : protocole de communication entre le navigateur du client et les serveurs Web, basé sur le principe des liens hypertextes ou des hyperliens.

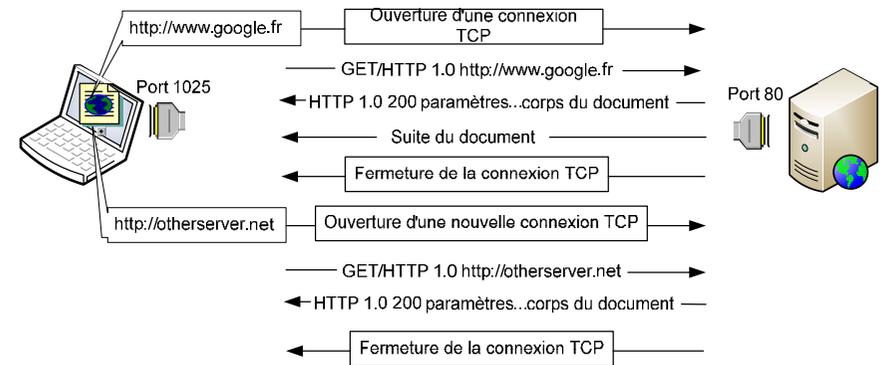


- Documents accessibles par un **URL (Uniform Ressource Locator)** :
 syntaxe : `protocol://[user:password@[host]:port]/[path]/[document]`
 - user : nom d'utilisateur utilisé par certains protocoles (FTP...)
 - password : mot de passe pour user
 - host : nom complet d'hôte FQDN (www.education.gouv.fr)
 - port : port sur lequel se connecter, la plupart des protocoles ont un port par défaut (80 pour HTTP...)
 - path : détail sur la façon d'accéder à la ressource (/~lohier)
 - document : nom de la ressource (index.html, video.mpeg...)

- ❑ Fonction de base de HTTP : transfert de fichier (essentiellement au format HTML) localisé grâce à son URL entre client et un serveur Web.
- ❑ A partir de la version 1.0 : possibilité de transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage de type MIME.
- ❑ La communication entre le navigateur et le serveur se fait en deux temps :
 - le navigateur effectue une requête HTTP ;
 - le serveur traite la requête puis envoie une réponse HTTP.



- ❑ A chaque nouvel échange requête/réponse entre le client et le serveur, une nouvelle connexion TCP/IP est établie.
- ❑ Une fois la séquence requête/réponse achevée, la connexion est refermée.
- ❑ Si la connexion est persistante (keep-alive, possible avec HTTP 1.1) tous les éléments de la page correspondant au même URL de base, seront chargés sur la même connexion TCP.



- ❑ Syntaxe d'une requête HTTP (<crLf> signifie retour chariot et saut de ligne) :
 - METHODE URL VERSION<crLf>
 - EN-TETE : Valeur<crLf>
 - EN-TETE : Valeur<crLf>
 - Ligne vide<crLf>
 - CORPS DE LA REQUETE
- ❑ La ligne de requête comprend 3 éléments séparés par un espace :
 - La méthode (GET, HEAD, POST...);
 - L'URL ;
 - La version du protocole utilisé par le client (actuellement HTTP/1.1).
- ❑ Les champs d'en-tête : ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la requête et/ou le client (navigateur, OS...).
- ❑ Le corps de la requête : ensemble de lignes optionnel devant être séparé des lignes précédentes par une ligne vide (par exemple pour un envoi de données par la méthode POST lors de l'utilisation de formulaire).

- ❑ Syntaxe d'une réponse HTTP (<crLf> signifie retour chariot ou saut de ligne) :
 - VERSION-HTTP CODE INDICATION<crLf>
 - EN-TETE : Valeur<crLf>
 - EN-TETE : Valeur<crLf>
 - Ligne vide<crLf>
 - CORPS DE LA REPONSE
- ❑ L'état de la réponse comprend 3 éléments :
 - version HTTP ;
 - un code d'état décrivant le résultat (200 pour un acquittement, 404...);
 - Une chaîne de caractère donnant une indication sur le résultat (OK, Not Found...)
- ❑ Les champs d'en-tête : ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse (type du serveur, version de MIME...)
- ❑ Le corps de la réponse qui contient les données HTTP, généralement au format html.

- ❑ La méthode GET permet de récupérer une ressource localisée par un URL.
- ❑ Syntaxe :
 GET URL VERSION<crLf>
 EN-TETE : Valeur<crLf>.
 EN-TETE : Valeur<crLf>
 Ligne vide<crLf>
- ❑ Exemple de requête GET :

```
GET http://www.voila.fr/ HTTP/1.0\r\n
Accept: application/vnd.ms-excel, application/msword\r\n
Accept-Language: fr\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)\r\n
Host: www.voila.fr\r\n
Proxy-Connection: Keep-Alive\r\n
\r\n
```

- ❑ Pour HTTP 1.1, l'URL est passé dans le champ d'en-tête « Host » qui devient obligatoire.
- ❑ Exemple de requête GET 1.1:

```
GET / HTTP/1.1\r\n
Host: www.voila.fr\r\n
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.8.1.6) Gecko/20070725 Firefox/2.0.0.6\r\n
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8\r\n
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Keep-Alive: 300\r\n
Connection: keep-alive\r\n
Cookie: PersoHome=d:4&nVer:1.2; PersoModRecherche=Cd:acc_search_sld1&nVer:1.2; \r\n
\r\n
```

- ❑ Description de quelques en-têtes de requête :

Nom de l'en-tête	Description
Accept	Type de contenu accepté par le navigateur (par exemple <i>text/html</i>).
Accept-Charset	Jeu de caractères attendu par le navigateur.
Accept-Encoding	Codage de données accepté par le navigateur.
Accept-Language	Langage attendu par le navigateur (anglais par défaut).
Content-Type	Type de contenu du corps de la requête (par exemple <i>text/html</i>).
Date	Date de début de transfert des données.
User-Agent	Chaîne donnant des informations sur le client : nom et version du navigateur, de l'OS...
Host	URL demandé.
Connection	Permet à l'émetteur de spécifier les options utilisées pour cette connexion : keep-alive, close...
Keep-Alive	En cas de connexions persistentes : spécifie la durée ou le nombre maximum de requêtes multiples sur la même connexion TCP/IP.

- ❑ La première ligne d'une réponse à un GET reprend la syntaxe générale avec les 3 champs :
 VERSION-HTTP CODE INDICATION<crLf>
- ❑ Après les en-têtes et une ligne vide, les données HTTP sont envoyées.
- ❑ Exemple de réponse à un GET :

```
HTTP/1.1 200 OK\r\n
Date: Thu, 13 Sep 2007 09:59:46 GMT\r\n
Server: Apache/2.0.50 (Unix) mod_ssl/2.0.50 OpenSSL/0.9.7c\r\n
Last-Modified: Mon, 03 Sep 2007 12:52:39 GMT\r\n
ETag: "51007b-146c-a5c6d3c0"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 5228\r\n
Keep-Alive: timeout=15, max=99\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html\r\n
Content-Language: fr\r\n
\r\n
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"\r\n
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">\r\n
<html xmlns="http://www.w3.org/1999/xhtml">\r\n
<!-- DW6 -->\r\n
<head>\r\n
...
```

- Les en-têtes sont fonction du type de réponse.
- Autres exemples de réponse à un GET :

```
HTTP/1.1 304 Not Modified\r\n
Date: Wed, 12 Sep 2007 14:12:36 GMT\r\n
Server: Apache\r\n
Connection: Keep-Alive\r\n
Keep-Alive: timeout=15, max=100\r\n
\r\n
```

```
HTTP/1.1 404 Not Found\r\n
Date: Thu, 13 Sep 2007 10:49:41 GMT\r\n
Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)\r\n
Connection: close\r\n
Transfer-Encoding: chunked\r\n
Content-Type: text/html; charset=iso-8859-1\r\n
\r\n
```

- Description de quelques en-têtes de réponse :

Nom de l'en-tête	Description
Content-Encoding	Type de codage du corps de la réponse
Content-Language	Type de langage du corps de la réponse
Content-Length	Longueur du corps de la réponse
Content-Type	Type de contenu du corps de la réponse (par exemple <i>text/html</i>)
Server	Caractéristiques du serveur ayant envoyé la réponse
Date	Date de début de transfert des données
Last-Modified	Date de la dernière modification de la ressource
Transfer-Encoding	Spécifie un encodage particulier des données transférées
Forwarded	Utilisé par les machines intermédiaires entre le navigateur et le serveur
Location	Redirection vers une nouvelle URL associée au document

- Le code de réponse est constitué de trois chiffres : le premier indique la classe d'état (réussite, redirection, erreur) et les suivants la nature exacte de l'erreur.

20x	Réussite	Bon déroulement de la transaction
200	OK	La requête a été accomplie correctement
201	CREATED	Elle suit une commande POST, elle indique la réussite, le corps du reste du document est sensé indiquer l'URL auquel le document nouvellement créé devrait se trouver.
202	ACCEPTED	La requête a été acceptée, mais la procédure qui suit n'a pas été accomplie
203	PARTIAL INFORMATION	Lorsque ce code est reçu en réponse à une commande GET, la réponse n'est pas complète.
204	NO RESPONSE	Le serveur a reçu la requête mais il n'y a pas d'information à renvoyer
30x	Redirection	La ressource n'est plus à l'emplacement indiqué
301	MOVED	Les données demandées ont été transférées a une nouvelle adresse
302	FOUND	Les données demandées sont à une nouvelle URL, mais ont cependant peut-être été déplacées depuis...
303	METHOD	Cela implique que le client doit essayer une nouvelle adresse, en essayant de préférence une autre méthode que GET
304	NOT MODIFIED	Lorsque le client a effectué une commande GET conditionnelle (en demandant si le document a été modifié depuis la dernière fois) et que le document n'a pas été modifié.

40x	Erreur dûe au client	Requête incorrecte
400	BAD REQUEST	Syntaxe de la requête mal formulée ou impossible à satisfaire
401	UNAUTHORIZED	Le paramètre du message donne les spécifications des formes d'autorisation acceptables.
402	PAYMENT REQUIRED	Le client doit reformuler sa demande avec les bonnes données de paiement
403	FORBIDDEN	L'accès à la ressource est interdit
404	NOT FOUND	Classique! Le serveur n'a rien trouvé à l'adresse spécifiée.
500	INTERNAL ERROR	Le serveur a rencontré une condition inattendue qui l'a empêché de donner suite à la demande
50x	Erreur dûe au serveur	Erreur interne du serveur
501	NOT IMPLEMENTED	Le serveur ne supporte pas le service demandé.
502	SERVICE TEMPORARILY OVERLOADED	Le serveur ne peut pas répondre pour l'instant, le trafic est trop dense.
503	GATEWAY TIMEOUT	La réponse du serveur a été trop longue vis à vis du temps pendant lequel la passerelle était préparée à l'attendre.

- Similaire à la méthode GET mais ne demande que l'en tête du message et non le corps (page html).
- Utilisé par exemple pour vérifier l'existence d'une ressource en cache ou la date de dernière modification d'un fichier...

□ Exemple :

Requête

```
HEAD / HTTP/1.1\r\n
Host: www.univ-mlv.fr\r\n
```

Réponse

```
HTTP/1.1 200 OK
Date: Thu, 13 Sep 2007 09:08:06 GMT
Server: Apache/2.0.54 (Debian GNU/Linux) mod_jk/2.0.4 PHP/4.3.10-19 mod_ssl/2.0.54 OpenSSL/0.9.7e
Last-Modified: Mon, 10 Sep 2007 12:02:27 GMT
X-Powered-By: PHP/4.3.10-19
Content-Length: 79
Connection: close
Content-Type: text/html
```

- Permet au client d'envoyer des données au serveur, par exemple le contenu d'un formulaire renseigné par l'utilisateur.
- Syntaxe :
POST URL VERSION<crlf>
EN-TETE : Valeur<crlf>
EN-TETE : Valeur<crlf>
Ligne vide<crlf>
name1=value1&name2=value2...
- Exemple de requête POST:

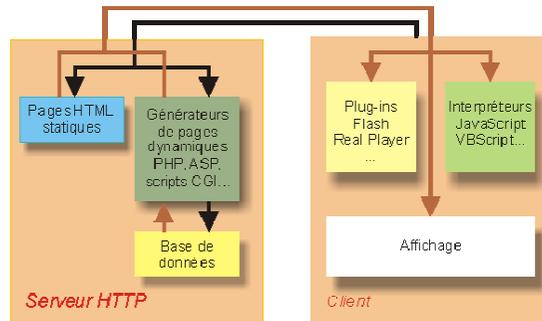
```
POST /index.php HTTP/1.1\r\n
Host: jungle.net\r\n
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.8.1.6) Gecko/20070725
Firefox/2.0.0.6\r\n
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8\r\n
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Keep-Alive: 300\r\n
Connection: keep-alive\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 65
\r\n
username=tarzan%40jungle.net&password=jane&logon=Log+on
```

- Après traitement sur le serveur (authentification, recherche dans une base de données...) par un script (PHP, ASP, CGI...), les informations sont renvoyées comme pour la méthode GET.
- Exemple de réponse à un POST

```
HTTP/1.1 200 OK\r\n
Date: Thu, 13 Sep 2007 13:11:47 GMT\r\n
Server: Apache/2.2.3 (Unix) mod_ssl/2.2.3 OpenSSL/0.9.7a PHP/5.2.3\r\n
X-Powered-By: PHP/5.2.3\r\n
Pragma: no-cache\r\n
Expires: 0\r\n
Content-Length: 4723
Keep-Alive: timeout=5, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html;charset=UTF-8\r\n
\r\n
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
'http://www.w3.org/TR/html4/strict.dtd">\n
<head>\n
...
```

- En plus des 3 méthodes de base (GET, HEAD et POST), la version 1.1 propose 5 méthodes supplémentaires :
 - **OPTIONS** : permet d'interroger le serveur sur les options disponibles pour obtenir la ressource.
 - **TRACE** : méthode de contrôle. Demande au serveur de renvoyer la requête telle qu'elle a été reçue.
 - **CONNECT** : permet de créer un tunnel HTTP (ou HTTPS) de bout en bout, intervient dans le fonctionnement des serveurs de proxy.
 - **PUT** : permet d'envoyer au serveur un document à enregistrer à l'URI spécifiée. (c'est directement le serveur qui traite le document envoyé, et non pas un script comme pour la méthode POST).
 - **DELETE** : efface la ressource spécifiée.
Ces deux dernières méthodes sont introduites pour permettre une mise à jour des sites distants via HTTP.
- Par défaut, HTTP 1.1 utilise des connexions persistantes : la connexion n'est pas immédiatement fermée après une requête, mais reste disponible pour une nouvelle requête (*keep-alive*).
- La persistance des connexions permet d'accélérer le chargement de pages contenant plusieurs ressources.
- Il est aussi permis à un client HTTP 1.1 d'envoyer plusieurs requêtes sur la même connexion sans attendre les réponses (*pipelining*)

- ❑ Une page HTML peut contenir des éléments dynamiques construits à partir de sources d'informations diverses au moment de l'envoi au client.
- ❑ Les méthodes associées permettent de fournir des d'informations impossibles à réaliser avec des pages purement statiques :
 - > résultat d'un calcul à partir d'éléments transmis par le client ;
 - > informations issues d'une base de données mise à jour.
- ❑ Ces informations peuvent évoluer à tout instant et leur affichage via HTTP nécessite leur intégration en temps réel dans le document.



- ❑ Plusieurs possibilités pour réaliser des pages dynamiques :
 - > des exécutables CGI (*Common Gateway Interface*) qui construisent intégralement un flux HTML au moment de leur appel ;
Ces exécutables peuvent être écrits dans un langage compilé comme le C ou dans un langage interprété comme le Perl (plus fréquent).
 - > des langages plus spécialisés comme PHP (*Personal Home Page*) ou ASP (*Active Server Pages*), qui s'intègrent dans le code html.
- ❑ Ces technologies sont dites *Server Side* : les traitements sur l'information sont effectués sur le serveur.
- ❑ L'avantage du *Server Side* est que le code HTML reçu par le client est du HTML pur, ce qui veut dire qu'à priori, tout navigateur peut l'afficher correctement...
- ❑ L'inconvénient est que le serveur voit sa charge augmenter (calculs, formulaires, recherche dans une base de données...) ce qui peut ralentir considérablement la navigation en cas de connexion lente.

- ❑ Les traitements d'informations peuvent aussi être utiles côté client :
 - > contrôle de la validité des informations saisies dans un formulaire ;
 - > traitement local d'informations pour afficher un résultat (calculatrice par ex.)
 - > animations, menus déroulants...
- ❑ Ces traitements peuvent être réalisés de diverses manières :
 - > les **applets Java** : portion de code Java téléchargé depuis le serveur et exécuté dans une fenêtre du navigateur, si une machine virtuelle Java est installée.
 - > les **JavaScripts** : petits applicatifs transmis dans le document HTML et exécutés côté client par le navigateur ;
 - > Les **composants ActiveX** : exécutables compilés qui ne peuvent s'exécuter eux que dans Internet Explorer...
 - > Les « **plug-in** » sont des composants enfichables qui étendent les possibilités intrinsèques des navigateurs, comme l'affichage de documents "flash" par exemple.
- ❑ Les avantages sont de deux sortes :
 - > tout traitement de données réalisé localement est rapide et sans surcharge pour le serveur ;
 - > les effets d'animation, comme les menus déroulants ou les bandeaux défilants ne peuvent être que réalisés localement.
- ❑ Les inconvénients viennent des incompatibilités entre navigateurs et des trous de sécurité introduits par des exécutables téléchargés sur le client.

- ❑ MIME (*Multipurpose Internet Mail Extension*) est une méthode d'encodage des données conçue au départ pour transmettre des documents par courrier électronique.
- ❑ HTTP utilise MIME pour décrire les différents types de données qu'il doit manipuler.
- ❑ La ligne **Content type** est une description MIME des données transférées.
 - > Par exemple : `text/html` indique que les données sont de type texte, sous type HTML. Lorsque le navigateur reconnaît ce type texte, il sait qu'il doit interpréter les données comme du code HTML.
- ❑ Autres types :
 - > text : html, plain, xml...
 - > image : gif, jpeg...
 - > audio : mp3, wav....
 - > video : avi, mpeg...
- ❑ Si le document demandé est un document HTML, les données de réponse contiendront le texte du document HTML. Par exemple :


```
<HTML>
<HEAD><TITLE>Document HTML simple</TITLE></HEAD>
<BODY>
<P> Cela est un document HTML minimaliste.
</BODY>
</HTML>
```

- ❑ Dans HTTP, le dialogue Client-Serveur est *sans état* ou *state-less* : le serveur ne stocke aucune information relative à une transaction.
- ❑ Le système des *cookies* inventé par Netscape et normalisé dans HTTP1.1 permet d'établir des transactions avec état.
- ❑ Un cookie est une information envoyée par le serveur, stockée coté client et renvoyée par le client lors d'une nouvelle connexion.
- ❑ Applications courantes :
 - mémorisation de la navigation du client pour une exploitation commerciale ;
 - prise de commandes en ligne via un serveur web : les libellés des différents articles sont stockés en local sur le client.

- ❑ En tête spécifique introduit dans la réponse du serveur :
Set-Cookie: Nom=Valeur; expires=Date; path=Chemin; domain=Domaine; secure.
- ❑ Set-Cookie comporte 5 champs :
 - Le premier permet d'envoyer au client une valeur associée à un identifiant afin que ces informations soient stockées en local, sous forme d'une chaîne de caractères.
 - *expires* permet d'indiquer la date d'expiration du cookie, une date d'expiration passée permet à un serveur d'effacer un cookie,
 - *Domain* spécifie le nom de domaine d'application du cookie. La valeur du cookie ne sera envoyée qu'aux serveurs appartenant au domaine précisé. Si le domaine n'est pas spécifié, la valeur par défaut est l'adresse DNS de la machine ayant généré le cookie.
 - *path* est utilisé pour désigner un sous-ensemble de ressources auxquelles le cookie est accessible. Si le champ *Domain* est renseigné, on concatène la valeur de ce champ à celle du *path*.
Exemple : si *path=/doc*, alors les ressources */doc/index.html*, */documents/toc.html*, etc. recevrons le cookie, à condition bien sûr que la valeur éventuelle du champ *Domain* corresponde à la machine considérée.
 - *Secure* est utilisé pour que le cookie ne soit envoyé que si la transmission s'effectue via une version sécurisée de type HTTPS.

- ❑ Les cookies stockés sur le client sont ensuite communiqués au serveur lors des nouvelles requêtes.
- ❑ Avant d'envoyer une nouvelle requête vers l'URL, le navigateur parcourt la liste des cookies qu'il possède.
- ❑ S'il y a occurrence entre l'URL de la ressource contenue dans la requête et les différents champs définissant le domaine d'application d'un cookie, la valeur de celui-ci est inséré dans la requête.
- ❑ Si plusieurs cookies sont applicables, le client les renvoie sur une ligne suivant la syntaxe :
Cookie: Nom1=valeur1; Nom2=valeur2; ...



- La première trame correspond à une demande d'URL (www.voilà.fr) à partir du navigateur du client.

```

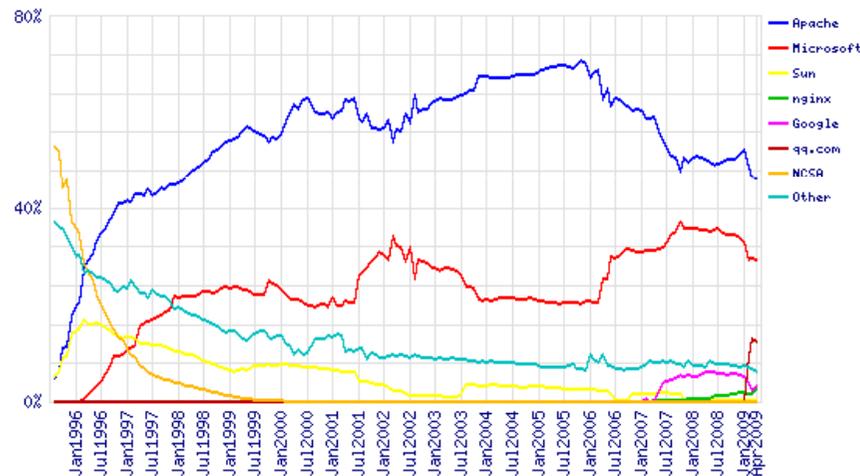
FRAME: Base frame properties
ETHERNET: ETYPE = 0x0800 : Protocol = IP: DOD Internet Protocol
IP: ID = 0xF416; Proto = TCP; Len: 391
TCP: A..., len: 351, seq: 8810009-8810359, ack: 23624030, win: 8736, src: 1125 dst: 80
HTTP: GET Request (from client using port 1125)
  HTTP: Request Method = GET
  HTTP: Uniform Resource Identifier = http://www.voila.fr/
  HTTP: Protocol Version = HTTP/1.0
  HTTP: Accept = application/vnd.ms-excel, application/msword, application/vnd.ms-powe
  HTTP: Accept-Language = fr
  HTTP: Accept-Encoding = gzip, deflate
  HTTP: User-Agent = Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)
  HTTP: Host = www.voila.fr
  HTTP: Undocumented Header = Proxy-Connection: Keep-Alive
  HTTP: Undocumented Header Fieldname = Proxy-Connection
  HTTP: Undocumented Header Value = Keep-Alive
  
```

- La deuxième trame contient la réponse (état, en-tête, données HTML).

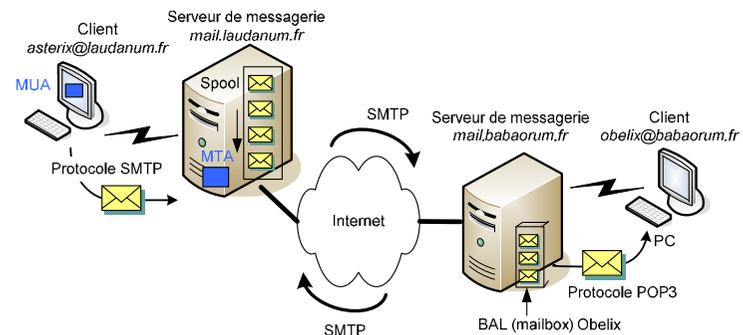
```

FRAME: Base frame properties
ETHERNET: ETYPE = 0x0800 : Protocol = IP: DOD Internet Protocol
IP: ID = 0x8C11; Proto = TCP; Len: 1496
TCP: A..., len: 1456, seq: 23624030-23625485, ack: 8810360, win: 8385, src: 80 dst: 1125
HTTP: Response (to client using port 1125)
  HTTP: Protocol Version = HTTP/1.0
  HTTP: Status Code = OK
  HTTP: Reason = OK
  HTTP: Undocumented Header = Via: 1.0 PROXYA
  HTTP: Undocumented Header Fieldname = Via
  HTTP: Undocumented Header Value = 1.0 PROXYA
  HTTP: Content-Type = text/html
  HTTP: Date = Thu, 20 Jan 2006 09:27:08 GMT
  HTTP: Server = Apache/1.3.9 (Unix)
  HTTP: Data: Number of data bytes remaining = 1329 (0x0531)
  
```

- Apache est depuis plusieurs années le logiciel serveur HTTP le plus utilisé (source Netcraft).



- Plus connu sous le nom de **Email** (*Electronic Mail* ou courrier électronique), ce service permet d'échanger des messages et des fichiers.
- Il nécessite :
 - un **serveur de messagerie** et un **logiciel serveur** ou **MTA** (*Mail Transfert Agent*, ex : sendmail, Postfix, exchange...);
 - un **client de messagerie** et un **logiciel client** ou **MUA** (*Mail user Agent*, ex : outlook, thunderbird...);
 - des **BAL** (boîte à lettre) sur le serveur pour chaque client géré par la messagerie;
 - des **protocoles d'échange** (SMTP, POP3, IMAP...).



La messagerie Email de base

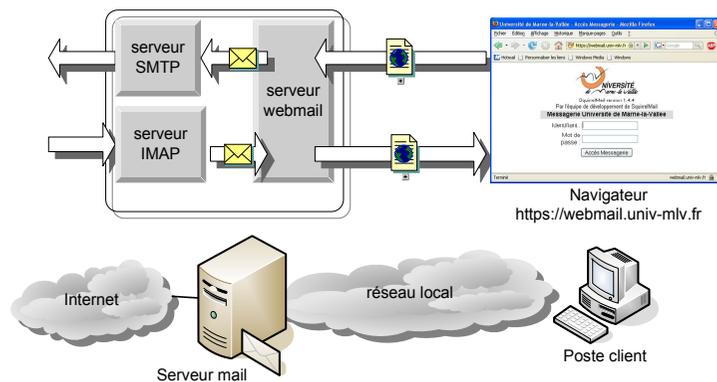
- ❑ Les messages envoyés par le client expéditeur sont stockés sur son serveur de messagerie puis transférés jusqu'au serveur destination dans la BAL appropriée.
- ❑ Le client destinataire consulte sa BAL et lit ses messages :
 - Le message peut rester stocké sur le serveur et lu à distance (fonctionnement en mode *online*) ou être déplacé vers la station du client et détruit sur le serveur (mode *offline*).
- ❑ Pour envoyer le courrier (courrier sortant) ou pour les échanges entre serveurs : protocole **SMTP** (*Simple Mail Transfer Protocol*).
- ❑ Pour lire le courrier (courrier entrant) :
 - protocole **POP3** (*Post Office Protocol*), orienté *offline*, permet de vérifier l'identité du client ;
 - protocole **IMAP** (*Interactive Mail Access Protocol*) destiné à la lecture interactive des messages
- ❑ Pour la mise en forme des messages (jeux de caractères, encodage, codage des fichiers joints...) : protocole **MIME** (*Multipurpose Internet Mail Extensions*). Il permet également la mise en forme du texte (mots soulignés, caractères en gras...).

La messagerie Listes et news

- ❑ Les **listes de diffusions** (*mailing lists*) permettent d'envoyer un même courrier à plusieurs personnes en utilisant une adresse commune de liste.
- ❑ Les destinataires doivent être abonnés à la liste, celle-ci est gérée sur un serveur de liste qui comprend les commandes d'abonnement, de désabonnement, de consultation d'archives...
- ❑ Les **forums** ou **news** permettent également de regrouper des abonnés appartenant à une même communauté ou intéressés par un même sujet.
- ❑ Contrairement aux listes de diffusion pour lesquelles un seul courrier est envoyé vers plusieurs destinataires, les messages des forums sont stockés sur un serveur et consultés ou enrichis lorsque l'utilisateur le souhaite.

La messagerie Webmail

- ❑ Le cœur du serveur webmail est un module logiciel d'interfaçage entre un serveur IMAP et le navigateur du client.
- ❑ Les paramètres de session navigateur-serveur sont stockés dans une base de données (MySQL par exemple).
- ❑ Cette solution ne nécessite pas de configuration, ni d'installation sur les postes clients.
- ❑ La page de gestion des messages reçus et d'émission des messages est fournie par le serveur webmail lors de la connexion du client au serveur.



La messagerie Chat

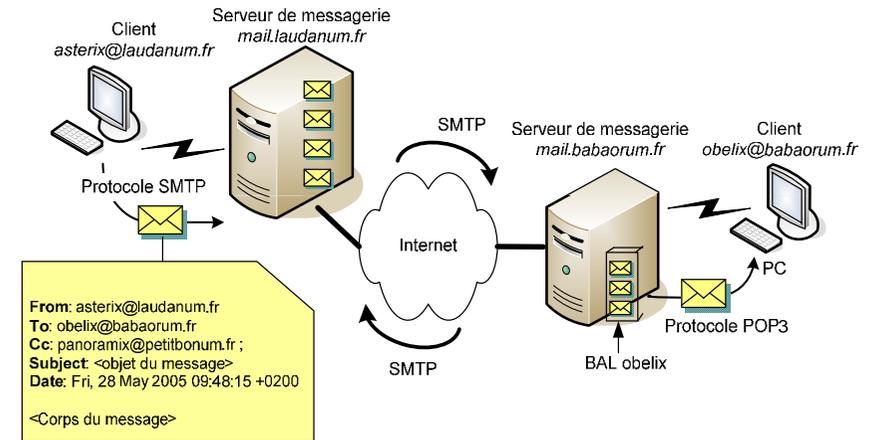
- ❑ Messagerie instantanée.
- ❑ Chaque utilisateur peut dialoguer à tout moment avec tous les contacts qu'il a référencé au préalable, si ceux-ci sont en ligne...
- ❑ Systèmes client serveur.
- ❑ Deux familles :
 - Protocoles de communication ouvert : XMPP, IRC, SIP
 - Protocoles propriétaires : Windows MSN, AIM, ICQ, Skype, Yahoo...
- ❑ Les protocoles propriétaires enferment des groupes d'utilisateur au sein d'un réseau fermé par la technologie : le contraire des protocoles historiques comme HTTP, SMTP qui permettent l'échange d'information avec le monde entier.



La messagerie RFC 2821 Le protocole SMTP (1)

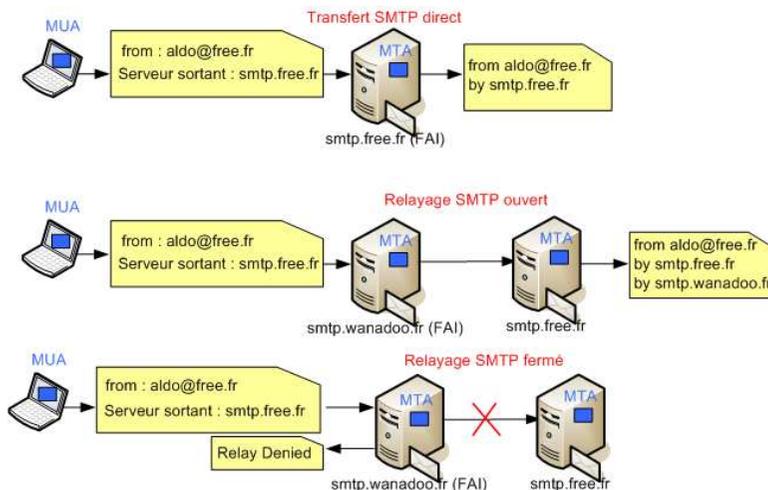
- SMTP (*Simple Mail Transport Protocol*) est le protocole courant de gestion du courrier électronique sur Internet.
- C'est un protocole point à point :
 - communication entre le client expéditeur et son serveur de messagerie ;
 - communication entre 2 serveurs de messagerie : celui de l'expéditeur et celui du destinataire.
- Ces serveurs sont chargés du stockage dans les BAL et du transport du courrier, ils doivent acheminer régulièrement les messages stockés vers les destinations mentionnées dans les champs adresse.
- SMTP est conçu au départ pour des systèmes reliés en permanence. Un utilisateur connecté de façon intermittente (*Dial up*) via le RTC utilisera SMTP pour le **courrier sortant**, et un protocole tel POP3 pour le **courrier entrant**.

La messagerie Protocole SMTP (2)



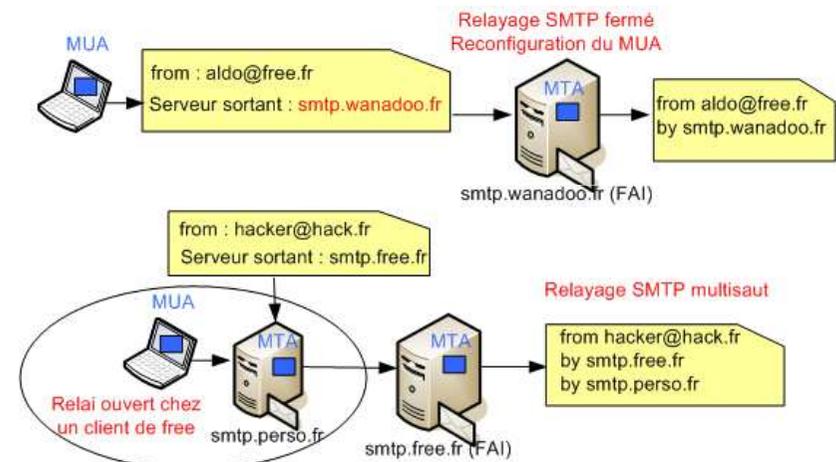
La messagerie Relayage SMTP

- Le « relayage » SMTP (relaying) permet à un abonné d'envoyer du courrier à partir de son compte email même s'il n'est pas connecté par le FAI lui ayant fourni ce compte.
- Pour éviter les spams, le relayage est fermé chez la plupart des FAI.



La messagerie Relayage SMTP

- Lorsque le relayage est fermé, la reconfiguration du client permet de contourner le problème.
- Un relai ouvert sur le MTA d'un client est une porte d'entrée pour le relayage multisaute.



La messagerie **Format SMTP**

- Le protocole SMTP spécifie :
 - le format des adresses des utilisateurs suivant une notation Internet classique faisant figurer le nom de l'utilisateur suivi du nom de domaine (asterix@babaorum.fr) ;
 - les champs des courriers (*from*, *to*...) ;
 - les possibilités d'envoi groupé :
 - ✓ Le champ CC pour *Carbone Copy* ou Copie Conforme permet d'envoyer le même message à plusieurs personnes, tous les noms des destinataires apparaîtront en clair dans le champ *to* ;
 - ✓ Le champ BCC (*Blinde Carbone Copy*) ou CCI (Copie Conforme Invisible) permet de masquer le nom des autres destinataires dans le champ *to* ;
- la gestion des heures ;
- le codage utilisé pour le message et les fichiers joints (« attachés ») :
 - texte pur codé en **ASCII 7** (RFC 822) ou 8 bits pour une prise en compte des caractères accentués ;
 - standard **MIME** pour du texte formaté, des images ou du son.

La messagerie **Format des messages SMTP**

- Le tableau suivant donne les principaux champs d'en-tête **RFC 822** liés au transport du message (*to*, *from*...) et ses caractéristiques (*date*, *sujet*...).

En-tête	Signification
To :	Adresse(s) électronique(s) du(des) destinataire(s) primaire(s)
Cc :	Adresse(s) électronique(s) du(des) destinataire(s) secondaire(s)
Bcc :	Adresse(s) électronique(s) du(des) destinataire(s) en copie caché
From :	Adresse électronique de l'auteur du message
Received	Adresse électronique de l'expéditeur du message
Return Path :	Peut être utilisé pour identifier un chemin de retour vers l'émetteur
Date :	Date et heure de l'envoi du message
Reply to :	Adresse électronique du destinataire d'une réponse
Message id :	Numéro servant à référencer le message
In Reply to :	Numéro (message id) du message auquel on répond
Subject :	Résumé en une ligne du message

La messagerie **SMTP et MIME**

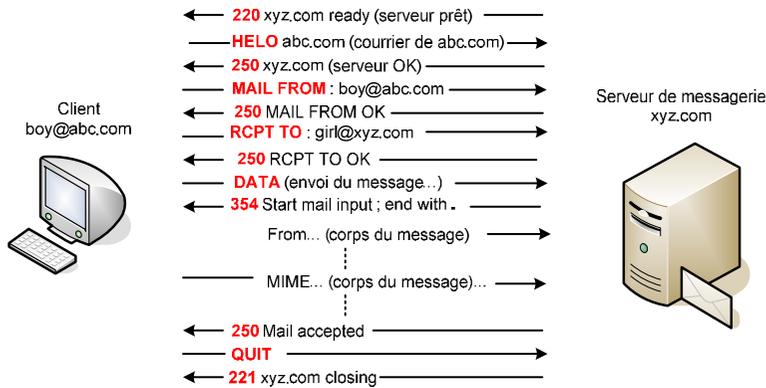
- Le standard **MIME** (RFC1521) reprend le standard simple de codage ASCII (RFC 822) mais en structurant le corps du message et en définissant des règles de codage pour le corps du message et les fichiers joints :
 - un codage base64 pour les messages binaires (groupes de 24 bits segmentés en 6 bits et ASCII légal : A pour 0, B pour 1...) ;
 - un codage QP (*Quoted Printable*) pour les messages texte : codage ASCII sur 7 bits et une séquence spécifique composé du signe égal et de la valeur du caractère pour les codes supérieurs à 127 (Ex : St=E9phane) ;
 - un codage spécifique lié à l'application (son, vidéo...).
- Pour définir ces nouveaux codages, MIME utilise d'autres en-têtes spécifiques, par exemple :
 - « Content-type : text.html » précise que le texte du message est codé en langage html ;
 - « Content-type : video:mpeg » donne le format d'un fichier vidéo attaché...

La messagerie **Commandes SMTP**

- Une fois formatés, les messages sont envoyés en utilisant les commandes SMTP :
 - commandes d'envoi constituées de quatre lettres ;
 - commandes de réponse du serveur constituées d'un code sur trois chiffres suivi d'un message texte.

Commandes d'envoi	Fonction
HELO <i>exp</i>	Requête de connexion en provenance d'un expéditeur SMTP.
MAIL FROM : <i>adr_exp</i>	Adresse de l'expéditeur, annonce le début d'un échange.
RCPT TO : <i>ad_dest</i>	Spécifie un destinataire, la commande peut être répétée.
DATA	Le récepteur interprète toutes les données suivantes comme faisant partie du message SMTP jusqu'à l'apparition d'un point correspondant à 2 sauts de ligne (CR LF CR LF).
QUIT	Demande au récepteur d'envoyer la réponse OK et de refermer la connexion.
Commandes de réponse	
250	Action demandée correctement effectuée (message OK).
354	Le message peut être transmis.
451	Action demandée annulée : erreur pendant le traitement.
550	Action non effectuée : boîte au lettre inaccessible.

- Les 3 phases classiques de dialogue se retrouvent au niveau SMTP :
 - établissement de la connexion au niveau SMTP et identification de la source et de la destination ;
 - envoi du message avec les différents en-tête RFC 822 et RFC 1521 ;
 - libération de la connexion.



- La requête de connexion est envoyée par le client SMTP par une commande HELO ou EHLO pour ESMTP (Extended SMTP).

```

# Frame 7 (68 bytes on wire, 68 bytes captured)
# Ethernet II, Src: Intel_31:ec:34 (00:16:6f:31:ec:34), Dst: Cisco-Li_24:f5:7f (00:0f:66:24:f5:7f)
# Internet Protocol, Src: 192.168.0.3 (192.168.0.3), Dst: 82.216.111.2 (82.216.111.2)
# Transmission Control Protocol, Src Port: 1329 (1329), Dst Port: smtp (25), Seq: 1, Ack: 46, Len: 14
# Simple Mail Transfer Protocol
  # Command: EHLO dell1810\r\n
    Command: EHLO
    Request parameter: dell1810
    
```

- Le serveur répond par le code de contrôle 250 (OK) suivi d'un message identifiant le serveur de mail. La connexion au niveau SMTP est réalisée.

```

# Frame 9 (144 bytes on wire, 144 bytes captured)
# Ethernet II, Src: Cisco-Li_24:f5:7f (00:0f:66:24:f5:7f), Dst: Intel_31:ec:34 (00:16:6f:31:ec:34)
# Internet Protocol, Src: 82.216.111.2 (82.216.111.2), Dst: 192.168.0.3 (192.168.0.3)
# Transmission Control Protocol, Src Port: smtp (25), Dst Port: 1329 (1329), Seq: 46, Ack: 15, Len: 90
# Simple Mail Transfer Protocol
  # Response: 250-smtp3.tech.numericable.fr\r\n
    Response code: 250
    Response parameter: smtp3.tech.numericable.fr
    
```

- Suivent les commandes « MAIL FROM » et « RCPT To » précisant l'expéditeur et le destinataire...

```

# Frame 10 (88 bytes on wire, 88 bytes captured)
# Ethernet II, Src: Intel_31:ec:34 (00:16:6f:31:ec:34), Dst: Cisco-Li_24:f5:7f (00:0f:66:24:f5:7f)
# Internet Protocol, Src: 192.168.0.3 (192.168.0.3), Dst: 82.216.111.2 (82.216.111.2)
# Transmission Control Protocol, Src Port: 1329 (1329), Dst Port: smtp (25), Seq: 15, Ack: 136, Len: 34
# Simple Mail Transfer Protocol
  # Command: MAIL FROM: <bush@whitehouse.gov>\r\n
    Command: MAIL
    Request parameter: FROM: <bush@whitehouse.gov>
    
```

```

# Frame 12 (85 bytes on wire, 85 bytes captured)
# Ethernet II, Src: Intel_31:ec:34 (00:16:6f:31:ec:34), Dst: Cisco-Li_24:f5:7f (00:0f:66:24:f5:7f)
# Internet Protocol, Src: 192.168.0.3 (192.168.0.3), Dst: 82.216.111.2 (82.216.111.2)
# Transmission Control Protocol, Src Port: 1329 (1329), Dst Port: smtp (25), Seq: 49, Ack: 144, Len: 31
# Simple Mail Transfer Protocol
  # Command: RCPT TO: <lohier@univ-mlv.fr>\r\n
    Command: RCPT
    Request parameter: TO: <lohier@univ-mlv.fr>
    
```

- ...le message est transmis au serveur avec les différents en-têtes et suivant les formats RFC 822 et RFC 1521.

```

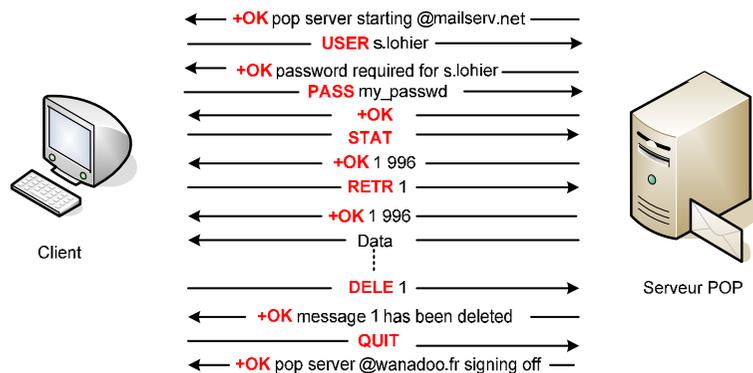
# Frame 25 (481 bytes on wire, 481 bytes captured)
# Ethernet II, Src: Intel_31:ec:34 (00:16:6f:31:ec:34), Dst: Cisco-Li_24:f5:7f (00:0f:66:24:f5:7f)
# Internet Protocol, Src: 192.168.0.3 (192.168.0.3), Dst: 82.216.111.2 (82.216.111.2)
# Transmission Control Protocol, Src Port: 1355 (1355), Dst Port: smtp (25), Seq: 157, Ack: 226, Len: 427
# Simple Mail Transfer Protocol
  # Message: From: "bush" <bush@whitehouse.gov>\r\n
    Message: To: <lohier@univ-mlv.fr>\r\n
    Message: Subject: test\r\n
    Message: Date: Fri, 14 Sep 2007 16:27:55 +0200\r\n
    Message: MIME-Version: 1.0\r\n
    Message: Content-Type: text/plain;\r\n
    Message: \tcharset="iso-8859-1"\r\n
    Message: Content-Transfer-Encoding: quoted-printable\r\n
    Message: X-Mailer: Microsoft office outlook, Build 11.0.5510\r\n
    Message: X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3138\r\n
    Message: Thread-Index: Acf223C0T1acPvgRlYeyfGx0P1v4g==\r\n
    Message: \r\n
    Message: Comment =E7a va ?\r\n
    Message: \r\n
    Message: \r\n
    Message: \r\n
    Message: \r\n
    Message: \r\n
    
```

- POP3 (*Post Office Protocol version 3*) conçu pour récupérer le courrier sur une machine distante pour un utilisateur non connecté en permanence à Internet, gère :
 - l'authentification, c'est-à-dire la vérification du nom et du mot de passe ;
 - la réception des courriers et fichiers attachés à partir du serveur de messagerie ;
 - la réception de messages d'erreur ou d'acquiescement.
- L'envoi de messages n'est pas supporté par le protocole POP3 de base. Il n'est pas sécurisé au niveau de la confidentialité dans la mesure où les messages sont stockés « en clair » sur le serveur de courrier.
- Par ailleurs, il est nécessaire de télécharger l'intégralité du courrier sur la station avant la lecture, sans possibilité de manipuler directement les messages sur le serveur.
- Le protocole IMAP (*Interactive Mail Access Protocol*) est une alternative à POP3. Il permet d'accéder aux messages sans les télécharger et d'effectuer des recherches de courrier selon critères.

- Les commandes POP3 (RFC 1939) reprennent la syntaxe sur 4 lettres de SMTP.
- Les réponses du serveur sont transmises sous forme d'une chaîne de caractères sont de deux types : +OK et -ERR suivi d'un texte.

Commande	Fonction
USER <u>nom</u>	Spécifie une boîte aux lettres.
PASS <u>password</u>	Spécifie un mot de passe.
STAT	Permet de récupérer le nombre et la taille des messages en attente.
LIST [<u>msg</u>]	Demande des informations sur le message dont le numéro est fourni.
RETR <u>msg</u>	Permet de récupérer une liste de messages.
DELE <u>msg</u>	Suppression du message spécifié.
QUIT	Le serveur supprime les messages lus et referme la connexion.

- La première trame correspond à la réponse (+OK) à une demande de connexion TCP au serveur POP3.
- Le client POP3 s'identifie ensuite au serveur, ce dernier demande un mot de passe au client qui lui transmet sous forme non crypté sur le réseau.
- Après acceptation, le client peut relever le courrier choisi à l'aide de la commande RETR (sans la commande DELE le courrier reste sur le serveur).



- La première trame correspond à la réponse (+OK) à une demande de connexion au serveur POP3.

```

* Frame 4 (72 bytes on wire, 72 bytes captured)
* Ethernet II, Src: Cisco-L1_24:f5:7f (00:0f:66:24:f5:7f), Dst: Intel_31:ec:34 (00:16:6f:31:ec:34)
* Internet Protocol, Src: 80.168.46.11 (80.168.46.11), Dst: 192.168.0.3 (192.168.0.3)
* Transmission Control Protocol, Src Port: pop3 (110), Dst Port: 1384 (1384), Seq: 1, Ack: 1, Len: 18
* Post Office Protocol
  +OK Hello there.\r\n
    Response indicator: +OK
    Response description: Hello there.
    
```

- Le client POP3 s'identifie au serveur.

```

* Frame 21 (67 bytes on wire, 67 bytes captured)
* Ethernet II, Src: Intel_31:ec:34 (00:16:6f:31:ec:34), Dst: Cisco-L1_24:f5:7f (00:0f:66:24:f5:7f)
* Internet Protocol, Src: 192.168.0.3 (192.168.0.3), Dst: 132.227.74.3 (132.227.74.3)
* Transmission Control Protocol, Src Port: 1388 (1388), Dst Port: pop3 (110), Seq: 1, Ack: 21, Len: 13
* Post Office Protocol
  USER lohier\r\n
    Request command: USER
    Request parameter: lohier
    
```

- Le serveur demande un mot de passe au client.

```

# Frame 22 (78 bytes on wire, 78 bytes captured)
# Ethernet II, Src: Cisco-Li_24:f5:7f (00:0f:66:24:f5:7f), Dst: Intel_31:ec:34 (00:16:6f:31:ec:34)
# Internet Protocol, Src: 193.50.159.100 (193.50.159.100), Dst: 192.168.0.3 (192.168.0.3)
# Transmission Control Protocol, Src Port: pop3 (110), Dst Port: 1386 (1386), Seq: 19, Ack: 14, Len: 24
# Post office Protocol
  # +OK Password required.\r\n
    Response indicator: +OK
    Response description: Password required.
    
```

- Le mot de passe est transmis non crypté sur le réseau...

```

# Frame 23 (69 bytes on wire, 69 bytes captured)
# Ethernet II, Src: Intel_31:ec:34 (00:16:6f:31:ec:34), Dst: Cisco-Li_24:f5:7f (00:0f:66:24:f5:7f)
# Internet Protocol, Src: 192.168.0.3 (192.168.0.3), Dst: 193.50.159.100 (193.50.159.100)
# Transmission Control Protocol, Src Port: 1386 (1386), Dst Port: pop3 (110), Seq: 14, Ack: 43, Len: 15
# Post office Protocol
  # PASS [redacted]\r\n
    Request command: PASS
    Request parameter: [redacted]
    
```

- Après acceptation, le client peut relever son courrier à l'aide de la commande RETR.

```

Return-Path: <bush@whitehouse.gov>
Delivered-To: lohier@univ-mlv.fr
Received: from localhost (localhost [127.0.0.1])
  by saintex.univ-mlv.fr (Postfix) with ESMTP id BE61E18C088
  for <lohier@univ-mlv.fr>; Fri, 14 Sep 2007 16:34:19 +0200 (CEST)
Received: from saintex.univ-mlv.fr ([127.0.0.1])
  by localhost (saintex [127.0.0.1]) (amavisd-new, port 10024)
  with ESMTP id 02997-04 for <lohier@univ-mlv.fr>;
  Fri, 14 Sep 2007 16:34:19 +0200 (CEST)
From: "bush" <bush@whitehouse.gov>
To: <lohier@univ-mlv.fr>
Subject: test
Date: Fri, 14 Sep 2007 16:34:00 +0200
MIME-Version: 1.0
Content-Type: text/plain;
  charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
X-Mailer: Microsoft Office Outlook, Build 11.0.5510
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3138
Thread-Index: Acf223COT1aCPVgIRlyefGKOIP4g==
Message-Id: <20070914143419.67CF116380A@saintex.univ-mlv.fr >
X-Virus-Scanned: by amavisd-new-20030616-p10 (Debian) at univ-mlv.fr
X-Spam-Status: No, hits=2.8 tagged_above=1.5 required=4.5 tests=BAYES_60,
  BLANK_LINES_70_80
X-Spam-Level: **
Comment ça va ?

Georges
    
```

- IMAP (Interactive Mail Access Protocol) : destiné à la lecture interactive des messages.
- Le système webmail fonctionne avec le protocole IMAP.
- Parmi les principales fonctionnalités d'IMAP :
 - Lecture des objets des messages seulement (sans le corps) ;
 - lecture des messages en les laissant sur le serveur ;
 - effacement de message sans l'avoir lu ;
 - marquage des messages sur le serveur...
 - création de dossiers sur le serveur ;
 - déplacement de messages sur le serveur d'un dossier à l'autre,
- Avantages
 - Gestion simplifiée de la messagerie en cas de mobilité de l'utilisateur (gestion des dossiers et messages sur le serveur).
 - Il est plus facile de changer de client de messagerie (aucun message à transférer, etc.)
- Inconvénients
 - Il faut gérer son espace disque du serveur.
 - Certains clients email sont plutôt lents pour récupérer des dossiers volumineux.
 - Moins rapide pour les recherches dans les messages sur le serveur.

- Structure des commandes IMAP proche de celle des commandes POP3.
- Les commandes supplémentaires permettent d'effectuer des opérations telles que :
 - la création, la suppression ou le changement de nom de boîtes aux lettres ;
 - la vérification de la présence de nouveaux messages ;
 - la suppression définitive de messages ;
 - la pose et la suppression de marqueurs (flags) ;
 - la recherche de messages ;
 - la récupération de parties de messages.
- Quelques commandes IMAP de base :
 - Authentification auprès du serveur :
 - Login <user> <mot_de_passe>
 - Choix d'une boîte aux lettres :
 - Select inbox
 - La commande Fetch est très puissante elle permet de faire de nombreuses et diverses sélections auprès de la liste des messages et des messages eux-mêmes :
 - Fetch <id(s)_msg(s)> <action>
 - Quitter la session de dialogue avec le serveur :
 - Logout

La messagerie Exemple de dialogue IMAP

```

* OK [CAPABILITY IMAP4REV1...] ← connexion au serveur
001 login testimap2 testimap2
001 OK [CAPABILITY IMAP4REV1...] ← sélection d'un dossier
002 select INBOX
* 1 EXISTS
* NO Trying to get mailbox lock from process 28032
* 0 RECENT
* OK [UIDVALIDITY 1068367935] UID validity status
* OK [UIDNEXT 2] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS {}] Permanent flags
002 OK [READ-WRITE] SELECT completed

003 fetch 1 BODY[HEADER] ← pour lire l'en-tête de l'unique message disponible
* 1 FETCH (BODY[HEADER] (471))
Return-path: <root@gw2.maison.mrs>
Envelope-to: testimap2@gw2.maison.mrs
Received: from root by gw2.maison.mrs with local (Exim 3.35 #1 (Debian))
 id 1A11IA-0007H1-00
 for <testimap2@gw2.maison.mrs>; Sun, 09 Nov 2003 09:51:30 +0100
To: testimap2@gw2.maison.mrs
Subject: test simple
Message-Id: <E1A11IA-0007H1-00@gw2.maison.mrs>
From: Christian Caleca <root@gw2.maison.mrs>
Date: Sun, 09 Nov 2003 09:51:30 +0100
)
003 OK FETCH completed

004 fetch 1 BODY[TEXT] ← pour lire le texte du message
* 1 FETCH (BODY[TEXT] (8))
coucou
)
004 OK FETCH completed ← déconnexion du serveur
005 logout
* BYE gw2.maison.mrs IMAP4rev1 server terminating connection
005 OK LOGOUT completed
    
```

Master TTT

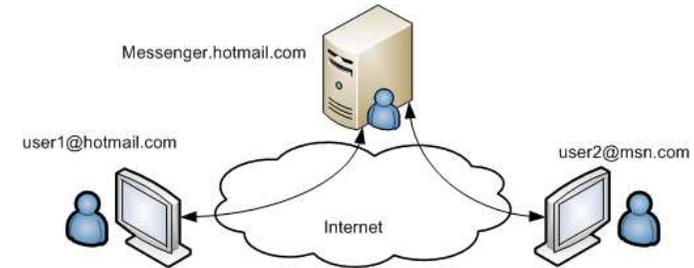
La couche Application

S. Lohier

77

La messagerie instantanée Protocole MSNP

- MSN Messenger utilise le protocole MSNP (*Mobile Status Notification Protocol*) au dessus de TCP (et éventuellement au dessus d'HTTP lorsque l'on passe par un Proxy).
- La version courante est la version 15 (MSNP15), utilisée par MSN Messenger version 8.1 et d'autres types de clients tiers.
- Le protocole n'a pas été toujours complètement secret : Microsoft a dévoilé la version 2 (MSNP2) aux développeurs en 1999.
- MSNP utilise une architecture centralisée : tous les messages d'un expéditeur passent par les serveurs de Microsoft avant d'être acheminés au destinataire.



Master TTT

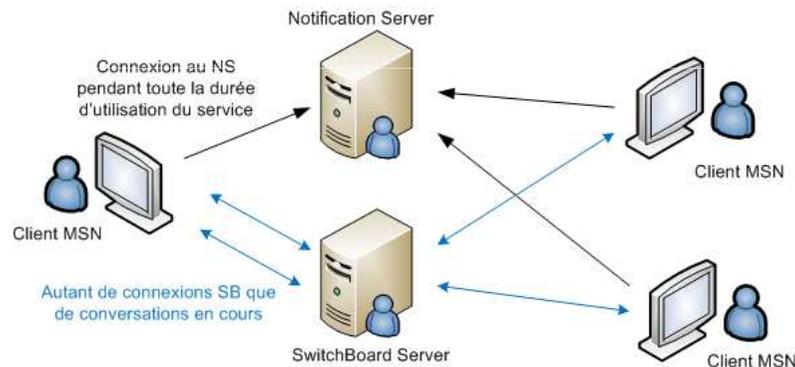
La couche Application

S. Lohier

78

La messagerie instantanée Entités MSNP

- Le client se connecte sur le port 1863 au NS (*Notification Server*) qui tient à jour une liste de présence.
- Le NS autorise alors le client à se connecter à un SB (*SwitchBoard Server*) qui rend le service de messagerie instantanée en liaison avec le destinataire.
- Tous les messages échangés passent par le SB.
- Le SB agit comme un proxy pour les messages échangés.



Master TTT

La couche Application

S. Lohier

79

La messagerie instantanée Commandes MSNP

- Les commandes peuvent être asynchrones, elles n'attendent pas forcément de réponses.
- Messages encodées en UTF-8 (donc en clair), utilise aussi l'encodage des URL et le format XML pour certains échanges.
- En générale, trois lettres en majuscules, un ID de transaction (chiffre compris entre 0 et 232) et des paramètres.
- Les commandes peuvent être suivi d'en tête comme en http : variable : valeur
- Différents types de commandes :
 - commandes avec les messages utiles ;
 - commandes pour gérer les erreurs ;
 - commandes pour l'identification ;
 - ...

Master TTT

La couche Application

S. Lohier

80

La messagerie instantanée Exemple d'échange MSNP

- ❑ bob@passport.com invite Alice a une discussion qu'il a initié avec Carol. Alice reçoit ce message sur sa connexion avec le NS :
<<< RNG 17342299 207.46.108.78:1863 CKI 1056411141.26158 bob@passport.com Bob\r\n
- ❑ La connexion avec le NS reste ouverte mais Alice ouvre une nouvelle connexion TCP avec le SB (207.46.108.78:1863) où a lieu la discussion entre bob et carol :
>>> ANS 1 alice@passport.com 1056411141.26158 17342299\r\n
- ❑ L'authentification a réussi, le SB envoie la liste des participants :
<<< IRO 1 1 2 bob@passport.com Bob\r\n<<< IRO 1 2 2 carol@passport.com Carol\r\n<<< ANS 1 OK\r\n
- ❑ Bob envoie alors un message à Alice avec une police *Lucida Sans Unicode*, grasse et de couleur bleu
<<< MSG bob@passport.com Bob 89\r\nMIME-Version: 1.0\r\nContent-Type: text/x-msmsgscontrol\r\nTypingUser: bob@passport.com\r\n\r\n\r\n<<< MSG bob@passport.com Bob 143\r\nMIME-Version: 1.0\r\nContent-Type: text/plain; charset=UTF-8\r\nX-MMS-IM-Format: FN=Lucida%20Sans%20Unicode; EF=B; CO=ff0000; CS=0; PF=22\r\n\r\n\r\nHello

Master TTT

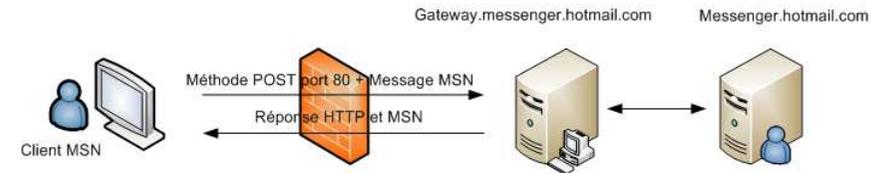
La couche Application

S. Lohier

81

La messagerie instantanée MSNP sur HTTP

- ❑ Pour passer les pare-feux et les réseaux avec proxy, possibilité d'utiliser HTTP pour transporter les messages MSNP.
- ❑ Les requêtes sont adressées à gateway.messenger.hotmail.com
- ❑ Pour maintenir la connexion ouverte, le client envoie des requêtes toutes les 2s (charge importante pour le réseau).



Master TTT

La couche Application

S. Lohier

82

La messagerie instantanée Dialogue MSNP via HTTP (1)

- ❑ Exemple de requête vers un NS via HTTP

POST

```
http://gateway.messenger.hotmail.com/gateway/gateway.dll?Action=open&
Server=NS&IP=messenger.hotmail.com HTTP/1.1\r\n
Accept: */*\r\n
Accept-Language: en-us\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: MSMSGSR\r\n
Host: gateway.messenger.hotmail.com\r\n
Proxy-Connection: Keep-Alive\r\n
Connection: Keep-Alive\r\n
Pragma: no-cache\r\n
Content-Type: application/x-msn-messenger\r\n
Content-Length: 18\r\n
\r\n
VER 5 MSNP8 CVR0\r\n
```

Master TTT

La couche Application

S. Lohier

83

La messagerie instantanée Dialogue MSNP via HTTP (2)

- ❑ Réponse vers un NS via HTTP

```
HTTP/1.0 200 OK\r\n
Server: Microsoft-IIS/5.0\r\n
Date: Tue, 18 Mar 2003 07:39:53 GMT\r\n
X-MSN-Messenger: SessionID=954547325.13160; GW-IP=207.46.110.18\r\n
Content-Length: 18\r\n
Content-Type: application/x-msn-messenger\r\n
Age: 0\r\n
Via: HTTP/1.1 ntl_site (Traffic-Server/5.2.0-R [c sSf ])\r\n
X-Cache: MISS from nautilus.localdomain\r\n
X-Cache-Lookup: MISS from nautilus.localdomain:80\r\n
Proxy-Connection: keep-alive\r\n
\r\n
VER 5 MSNP8 CVR0\r\n
```

Master TTT

La couche Application

S. Lohier

84

La messagerie instantanée Dialogue MSNP via HTTP (3)

- Message pour maintenir la connexion

POST

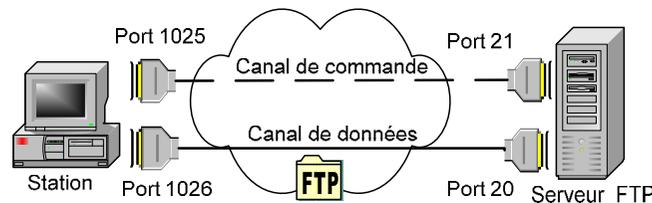
```
http://207.46.110.18/gateway/gateway.dll?Action=poll&SessionID=954
547325.32498 HTTP/1.1\r\n
Accept: */*\r\n
Accept-Language: en-us\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: MSMSGSL\r\n
Host: 207.46.110.18\r\n
Proxy-Connection: Keep-Alive\r\n
Connection: Keep-Alive\r\n
Pragma: no-cache\r\n
Content-Type: application/x-msn-messenger\r\n
Content-Length: 0\r\n
```

Transfert de fichiers RFC 959 FTP client/serveur

- Permet à un client de télécharger des fichiers auprès d'un serveur de fichiers
- Connexion et dialogue entre la station du client et le serveur suivant le protocole **FTP** (*File Transfer Protocol*).
- Identification de type **login/password**.
- Compte *user anonymous* pour servir des clients ne disposant pas de compte permanent (password demandé mais non analysé).
- Les logiciels client (Cute FTP, FileZilla...) disposent des commandes permettant de se déplacer dans l'arborescence du serveur, de définir le type des données transférées (binaire ou ASCII) et de télécharger un fichier.
- Les navigateurs Web intègrent les fonctions permettant la connexion aux serveurs et le transfert des fichiers (URL ftp://ServerName). Windows et Linux disposent de l'exécutable « ftp ».
- Il est souvent plus pratique d'utiliser un « vrai » logiciel client permettant de mémoriser les comptes (login, password) et leurs caractéristiques (ports, répertoire distant, cryptage...).
- Les logiciels serveur (Proftpd sous Linux, IIS sous Windows Server...) intègrent des fonction de sécurité sur les répertoires (fichiers *.htaccess...*), d'authentification par page...

Transfert de fichiers Le protocole FTP

- FTP (*File transfert Protocol*) permet le transfert de fichiers et de répertoires entre un serveur et un client sur un réseau IP.
- FTP utilise deux canaux TCP :
 - Le port TCP 20 est le **canal de données** ;
 - Le port TCP 21 est le **canal de commande**.
 - La connexion peut être effectuée de deux façons :
 - en **anonyme** (name : *anonymous*, password : *guest@home*) pour un accès limité aux répertoires publics ;
 - en **non anonyme** pour un accès associé à des permissions sur des fichiers ou des répertoires particuliers



Transfert de fichiers Commandes FTP

- Le client se connecte au serveur, s'identifie, puis travaille sur les répertoires distants.
- Toute commande envoyée par le client comprend de 3 à 4 lettres majuscules suivies éventuellement d'un espace et d'informations.

USER <login>

USER : ouvre une session FTP avec le nom d'utilisateur <login>. En FTP anonyme, taper « anonymous » ou rien du tout (dépend des serveurs FTP).

PASS <mdp>

PASS : <mdp> représente le mot de passe de l'utilisateur <login>. En FTP anonyme, taper « anonymous » ou votre adresse électronique.

CWD <path>

Change Working Directory : change le répertoire de travail. <path> doit être le chemin du nouveau répertoire depuis la racine du serveur (chemin absolu).

CDUP

Change Directory UP : change le répertoire de travail et l'établit sur le dossier parent.

PWD

Print Working Directory : récupère le chemin sur le serveur du dossier courant.

LIST

LIST : Liste les fichiers présents dans le répertoire courant du serveur.

STAT

STATUS : permet de s'informer sur l'état du serveur. Peut servir à connaître l'avancement d'un transfert en cours et accepte en argument un chemin d'accès, retourne alors les mêmes informations que LIST mais sur le canal de contrôle.

HELP ou « ? »

HELP : Permet de connaître toutes les commandes que le serveur accepte (Informations retournées sur le canal de contrôle).

DELE <fichier>

DELEte : supprime le fichier <fichier>, le chemin devant être donné par rapport à la racine du serveur distant.

RMD <dossier>

ReMove Directory : supprime <dossier>, le chemin devant être donné par rapport à la racine du serveur distant.

RNFR <ancien nom>

ReName FRom : 1ère étape pour renommer un fichier.

RNTO <nouveau nom>

ReName TO : 2ème étape pour renommer un fichier.

MKD <nom>

MaKe a new Directory : crée un nouveau répertoire <nom> dans le dossier courant.

QUIT

QUIT : « permet de quitter le serveur.

GET <fichier>

GET : récupère <fichier> du répertoire distant courant pour le sauvegarder dans le répertoire local.

MGET <pattern>

Multiple GET : récupère un ensemble de fichiers définis par le <pattern>. Exemple : MGET *.tgz qui définit l'ensemble des fichiers dont l'extension est tgz.

PUT <fichier>

PUT : envoi <fichier> dans le répertoire courant du serveur.

MPUT <pattern>

Multiple PUT : envoi un ensemble de fichiers définis par le <pattern>. Exemple : MPUT *.tgz qui envoi l'ensemble des fichiers dont l'extension est tgz.

- ❑ Lorsque le serveur reçoit une requête, il doit automatiquement répondre par un nombre suivi d'un espace, puis de texte commentaire.
- ❑ Comme pour HTTP ou SMTP, chaque code de réponse comprend trois chiffres :
 - le premier signifie une exécution réussie (1,2 ou 3) ou ratée (4 ou 5) ;
 - les deuxième et troisième chiffres précisent le code de retour ou une erreur.
- ❑ Le tableau suivant donne les principaux codes de réponse FTP sur 3 chiffres.
 - Exemple : 220 Service disponible pour nouvel utilisateur (connexion acceptée)

1 ^{er} chiffre	→ Peut prendre 5 valeurs différentes :
1	action amorcée, attente d'une autre réponse
2	action terminée, prêt pour un nouvelle commande
3	commande acceptée mais attente d'autres informations
4	commande refusée mais peut être réitérée
5	commande refusée. La réitérer provoquera la même erreur
2 ^{ème} chiffre	→ Donne une indication sur la nature de la réponse
0	Erreur de syntaxe ou commande illégale
1	Réponse à une requête d'informations
2	Réponse pour une commande de connexion
3	Réponse pour une commande d'authentification

- ❑ Les commandes de l'utilisateur (user, password, dir, get...) utilisée sur le logiciel client sont traduites en commandes internes FTP (USER, PASS, LIST, GET...);
- ❑ Elles sont suivies éventuellement d'arguments (nom d'utilisateur, nom de répertoire) ou de données correspondant au fichier transféré.

```

#ETHERNET: ETYPE = 0x0800 : Protocol = IP: DOD Internet Protocol
#IP: ID = 0x1D04; Proto = TCP; Len: 56
#TCP: .AP... len: 16, seq: 3242514-3242529, ack:2546277303, win: 8639, src: 1031 dst: 21 (FTP)
=FTP: Req. from Port 1031, 'USER anonymous'
FTP: FTP Command = USER
FTP: FTP Data: Number of data bytes remaining = 12 (0x000C)
00000000 00 60 08 58 10 39 00 10 5A 42 56 20 08 00 45 00 ..X.9..ZEV ..E.
00000010 00 38 1D 04 40 00 80 06 5B C6 C0 A8 00 83 C0 A8 -.8.@.Ç.[ã+ç.â+ç
00000020 00 22 04 07 00 15 00 31 7A 12 97 C5 1B B7 50 18 -"...$.iz.ù.â.â.P.
00000030 21 BF 52 4F 00 00 55 53 45 52 20 61 6E 6F 6E 79 !+R0..USER anony
00000040 6D 6F 75 73 0D 0A ..mou..
    
```

- ❑ Les codes de réponse sur 3 chiffres sont généralement suivis d'un message en ASCII.
- ❑ Lors d'une connexion anonyme, le mot de passe, qui n'est pas testé, peut être une adresse email.

```

#ETHERNET: ETYPE = 0x0800 : Protocol = IP: DOD Internet Protocol
#IP: ID = 0x19D3; Proto = TCP; Len: 108
#TCP: .AP... len: 68, seq:2546277303-2546277370, ack: 3242530, win:31744, src: 21 (FTP) dst: 1031
=FTP: Resp. to Port 1031, '331 Guest login ok, send your complete e-mail addr'
FTP: FTP Error Return Code = 331
FTP: FTP Command Arg1 = Guest
FTP: FTP Data: Number of data bytes remaining = 59 (0x003B)
00000000 00 10 5A 42 56 20 00 60 97 0A A8 95 08 00 45 10 ..ZEV..â.çò..E.
00000010 00 6C 19 D3 40 00 40 06 9E B3 C0 A8 00 22 C0 A8 .l.R@.@.x.+ç."+ç
00000020 00 83 00 15 04 07 97 C5 1B B7 00 31 7A 22 80 18 -".â$.ù.â.â.iz"P.
00000030 7C 00 96 01 00 00 33 33 31 20 47 75 65 73 74 20 |.â...331 Guest
00000040 6C 6F 67 69 6E 20 6F 6B 2C 20 73 65 6E 64 20 79 login ok, send y
00000050 6F 75 72 20 63 6F 6D 70 6C 65 74 65 20 65 2D 6D our complete e-m
00000060 61 69 6C 20 61 64 64 72 65 73 73 20 61 73 20 70 ail address as p
00000070 61 73 73 77 6F 72 64 2E 0D 0A ..assword..
    
```

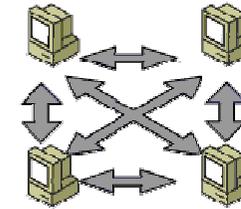
Transfert de fichiers Exemple de relevé d'un dialogue FTP

```

Etat : Connexion à plateforme1.univ-mlv.fr ...
Etat : Connecté à plateforme1.univ-mlv.fr. Attente du message d'accueil...
Réponse : 220 Bienvenue sur le serveur d'hébergement Hollywood.
Commande : USER lohier
Réponse : 331 Please specify the password.
Commande : PASS *****
Réponse : 230 Login successful.
Commande : FEAT
Réponse : 211-Features:
Réponse : MDTM
Réponse : REST STREAM
Réponse : SIZE
Réponse : 211 End
Commande : SYST
Réponse : 215 UNIX Type: L8
Etat : Connecté
Etat : Récupération de la liste de répertoires...
Commande : PWD
Réponse : 257 "/"
Commande : PORT 172,17,15,179,6,159
Réponse : 200 PORT command successful. Consider using PASV.
Commande : TYPE A
Réponse : 200 Switching to ASCII mode.
Commande : LIST
Réponse : 150 Here comes the directory listing.
Réponse : 226 Directory send OK.
Etat : Succès du listage du répertoire
Commande : TYPE A
Réponse : 200 Switching to ASCII mode.
Commande : PWD
    
```

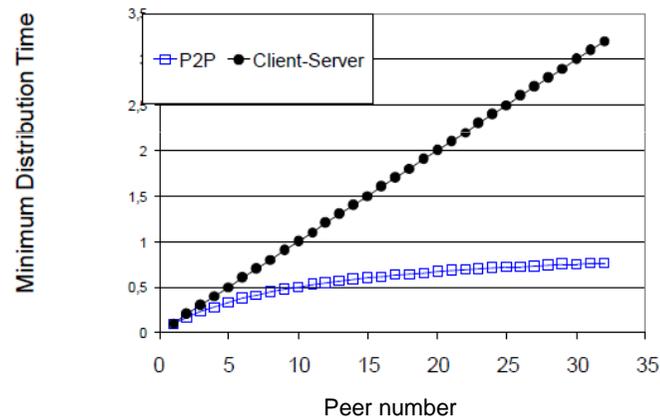
Transfert de fichiers Transfert P2P

- ❑ Les logiciels de transfert peer to peer (« d'égal à égal ») n'utilisent pas de serveur unique et centralisé.
- ❑ Ils permettent à certains postes de se déclarer ponctuellement en tant que serveur, multipliant ainsi les sources et décentralisant les stockages.
- ❑ Lors d'une recherche en tant que client, l'interrogation d'un annuaire de localisation de contenu permet de localiser les utilisateurs possédant la totalité ou une partie du fichier. Le téléchargement est ensuite effectué à partir de sources multiples décentralisées.
- ❑ Dans les applications P2P, les annuaires peuvent être :
 - centralisés sur des serveurs (Napster) ;
 - partiellement décentralisée sur les postes (Kazaa, Emule, edonkey);
 - décentralisés (Gnutella, BitTorrent)



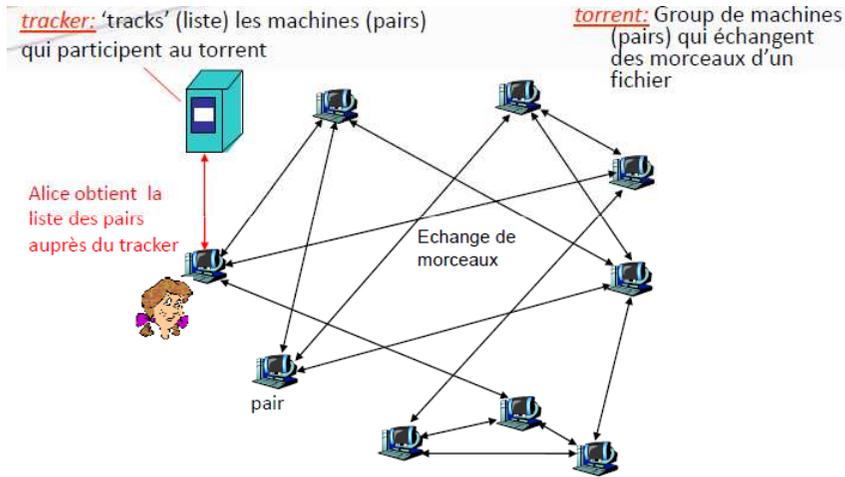
Transfert de fichiers P2P : intérêt

- ❑ Débit montant clients = u , Fichier/ u = 1 heure, débit serveur $u_s = 10 u$,
- ❑ Débit descendant minimal : $d_{\min} \geq u_s$



Transfert de fichiers P2P : cas de BitTorrent (1)

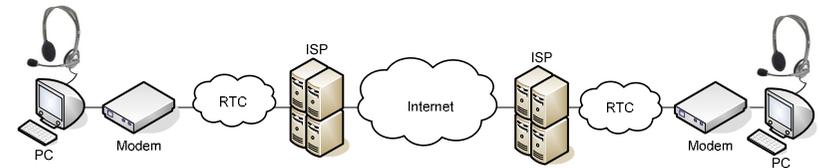
- ❑ **BitTorrent** est un protocole de transfert de données P2P développé en avril 2001 par Bram Cohen.
- ❑ Chaque personne qui a téléchargé le fichier devient une nouvelle source.
- ❑ de téléchargement
- ❑ Inconvénients : la recherche d'un fichier est complexe car il n'y a pas d'annuaire.
- ❑ Utilise un serveur web, ftp ou de news classique pour déposer les fichiers « Torrent »
- ❑ Un **fichier Torrent** est un fichier de description du contenu : table de hachage des différents morceaux du fichier.
- ❑ Ce fichier permet de localiser le « **Tracker** », programme simple mémorisant qui possède des morceaux du dit fichier. Mis à jour à chaque nouveau téléchargement.



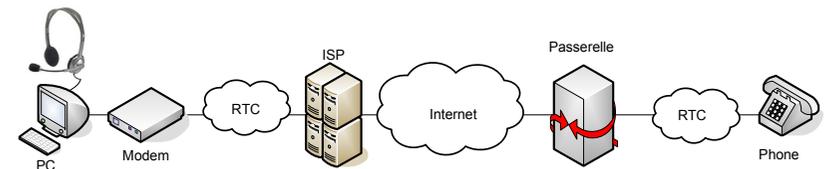
- ❑ Les fichiers sont découpés en morceaux de 256KB.
- ❑ Les pairs qui rejoignent le Torrent :
 - n'ont aucun morceaux du fichier, mais cherchent à en acquérir au fil du temps
 - s'enregistrent auprès du tracker pour recevoir la liste des pairs et se connectent à un sous ensemble de pairs (proches voisins)
- ❑ Au fur et à mesure du téléchargement (download) des morceaux, ceux-ci deviennent disponibles pour les autres pairs (upload)
- ❑ Les pairs peuvent se retirer à tout moment pour revenir plus tard.
- ❑ Quand un pair dispose de tout le fichier, il peut quitter le Torrent (selfishly) ou bien rester pour en faire profiter d'autres pairs (altruistically)

- ❑ La voix sur réseau IP, ou « VoIP » pour *Voice over Internet Protocol*, est une technique qui permet de communiquer par la voix via l'Internet ou tout autre réseau acceptant le protocole TCP/IP.
- ❑ Cette technologie est notamment utilisée pour supporter le service de téléphonie IP (« ToIP » pour *Telephony over Internet Protocol*).
- ❑ Toutes les solutions de téléphonie sur IP commencent par convertir la voix en paquets de données numériques (64 kbit/s en standard).
- ❑ Ils sont ensuite transmis sur le net de la même manière que les autres types de trafic (web, mail, ftp...).
- ❑ **Avantages :**
 - gratuité des appels métropolitains (parfois internationaux) ;
 - simplicité d'utilisation ;
 - options pointues et gratuites (transfert, messagerie, etc).
- ❑ **Inconvénients :**
 - coût parfois plus élevé pour les appelants et appels vers les numéros spéciaux ;
 - maîtrise des coûts parfois difficile (numéros 06, 07, 0800...);
 - qualité encore aléatoire du service.

- ❑ Selon les terminaux utilisés, on distingue trois modes de téléphonie sur IP :
 1. **Téléphonie entre micro-ordinateurs ("PC to PC").**
Les deux correspondants utilisent librement un logiciel de téléphonie sur IP (*soft phone*), type Skype.



2. **Téléphonie entre micro-ordinateur et poste téléphonique ("PC to phone").**
L'un des correspondants appelle l'autre correspondant sur son téléphone. Il doit se connecter sur un service spécial sur Internet, offert par un ISP, qui doit mettre en œuvre une "passerelle" avec le réseau téléphonique (possible avec Skype).

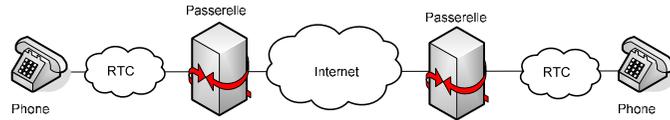


Voix sur IP Différents modes (suite)

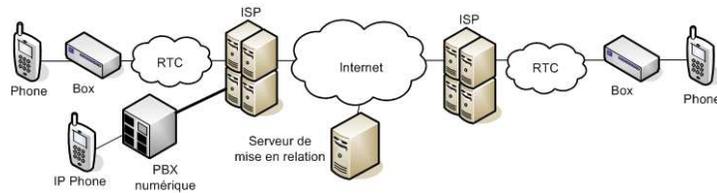
3. Téléphonie entre postes téléphoniques ("phone to phone").

Deux techniques sont possibles :

- utilisation de passerelles : un "pseudo-opérateur" a mis en place des passerelles entre le réseau téléphonique et le réseau IP. L'appelant appelle le numéro d'une passerelle et lui communique le numéro du destinataire.



- utilisation de boîtiers d'adaptation (type Freebox) entre postes téléphoniques et réseau ou dans un contexte d'entreprise : liaison directe d'un IP phone à un PBX numérique.



Master TTT

La couche Application

S. Lohier

101

Voix sur IP Mode PC to PC. Exemple : Skype

- ❑ Skype est un logiciel qui utilise la technologie P2P (les inventeurs de Skype sont les mêmes que ceux de Kazaa ...)
- ❑ Dans le système P2P, tous les nœuds du réseau sont reliés dynamiquement pour partager les fonctions de connexion, de routage et de gestion de bande passante.
- ❑ Par opposition aux systèmes maîtrisés par des serveurs centraux (technologie « client-serveur »).
- ❑ Comme pour toutes les solutions « PC to PC », il est nécessaire d'ajouter un ensemble casque/micro au PC pour communiquer.



Master TTT

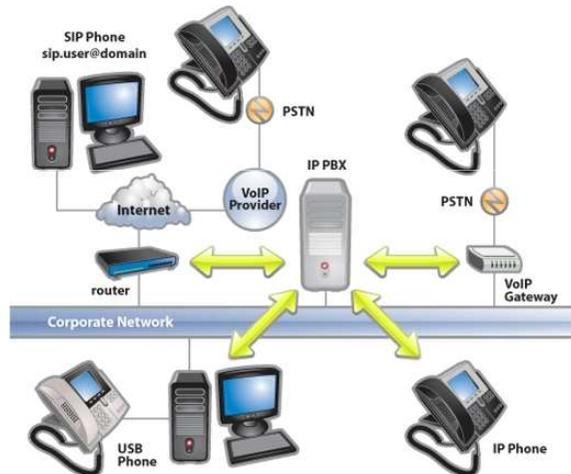
La couche Application

S. Lohier

102

Voix sur IP Solutions entreprise

- ❑ Beaucoup d'entreprises sont équipées avec des IP PBX (Private Branch eXchange).
- ❑ Ce sont des autocommutateurs numériques permettant de gérer en interne les communications à partir de téléphone IP, de soft phone ou de téléphones classiques.



Master TTT

La couche Application

S. Lohier

103

Voix sur IP Protocoles

Base :

- ❑ IP + UDP/TCP
- ❑ RTP (*Real Time Transport Protocol*), protocole au dessus de la couche transport pour gérer le transport temps réel de bout en bout des flux audio et vidéo sur les réseaux IP.
- ❑ RTCP (*Real time Transport Control Protocol*) est utilisé avec RTP pour le contrôle des flux temps réel. RTCP réalise un envoi périodique de paquets de contrôle à tous les participants d'une session.

Au niveau applicatif, les différents protocoles non propriétaires sont les suivants :

- ❑ Le protocole H323 est le plus connu et se base sur les travaux de la série H.320 sur la visioconférence sur RNIS. C'est une norme stabilisée avec de très nombreux produits sur le marché (terminaux, passerelles, logiciels).
- ❑ Le protocole SIP (*Session Initiation Protocol*) est natif du monde Internet et est un concurrent direct de l'H323. A l'heure actuelle, il est moins riche que H.323 au niveau des services offerts, mais il suscite actuellement un très grand intérêt dans la communauté Internet et principalement pour la téléphonie.

Master TTT

La couche Application

S. Lohier

104

Vidéo sur IP Off-line et streaming

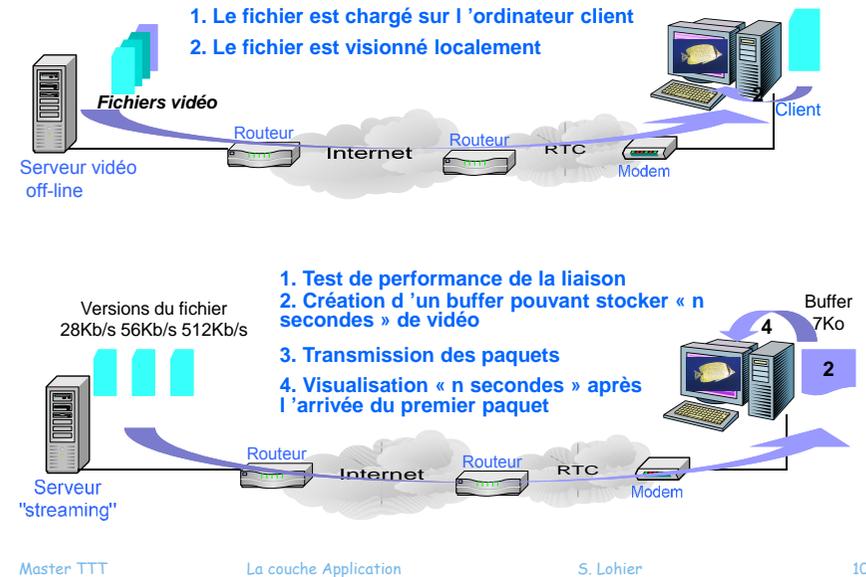
Principe :

- La lecture en continu (*streaming* - *stream=flux*) est un principe utilisé principalement pour l'envoi de contenu en « direct » (ou en léger différé).
- Très utilisé sur Internet, permet la lecture d'un flux audio ou vidéo (cas de la VoD), à mesure qu'il est diffusé.
- S'oppose ainsi à la diffusion par téléchargement qui nécessite de récupérer l'ensemble des données d'un morceau ou d'un extrait vidéo avant de pouvoir l'écouter ou le regarder (*off-line*).

Les outils :

- un logiciel de production audio/vidéo permettant le codage en fonction du débit du réseau emprunté ;
- un serveur pour stocker les fichiers, établir et paramétrer la connexion avec le client ;
- un logiciel de lecture sur le poste client.

Vidéo sur IP Off-line et streaming (2)



Vidéo sur IP Les protocoles

RSVP (Resource ReSerVation Protocol) :

- utilisé par les applications temps réel pour réserver les ressources nécessaires au niveau des routeurs situés le long du chemin de transmission ;
- un Contrôle de Politique (*Policy Control*) détermine si l'utilisateur a la permission administrative de faire de la réservation ;
- le contrôle d'admission (*Admission Control*) détermine si le nœud a suffisamment de ressource pour fournir la QoS demandée.

RTP (Real-time Transport Protocol) :

- assure la reconstruction temporelle, la détection de perte, la sécurité et l'identification du contenu ;
- fonctionne avec le protocole RTCP pour obtenir des feedback concernant la qualité de la transmission ;
- fournit un système d'horloge (timestamping), une numérotation des séquences et d'autres mécanismes pour gérer les problèmes liés au temps.

En-tête IP En-tête UDP En-tête RTP Données vidéo temps-réel

Vidéo sur IP Les protocoles (2)

RTCP (Real-Time Control Protocol) :

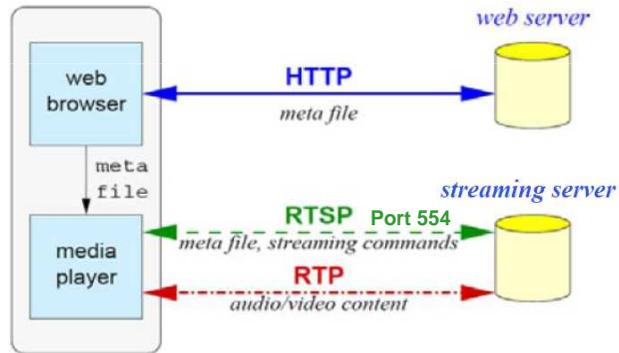
- Les participants envoient périodiquement des paquets RTCP pour donner des informations sur la qualité du service délivré. Fonctionne avec RTP.
- offre les services suivants :
 - ✓ contrôle de la congestion et monitoring de la QoS ;
 - ✓ identification de la source ;
 - ✓ synchronisation inter-média ;
 - ✓ information de contrôle.

RTSP (Real-Time Streaming Protocol) :

- protocole de niveau applicatif prévu pour fonctionner sur des protocoles tels que RTP/RTCP et RSVP.
- initie et commande à distance des flux multimédia stockés sur un serveur de données à travers un réseau IP ;
- les flux peuvent provenir soit de vidéos stockées, soit d'une source temps réel (caméra, micro) ;
- offre des fonctionnalités comme l'arrêt, l'avance rapide, la recherche avancée pour des flux vidéo et audio.
- les données multimédia sont transmises séparément en utilisant le plus souvent RTP.

Vidéo sur IP Protocoles RTSP pour le streaming

- RTSP est similaire, au niveau de la syntaxe et des fonctionnalités à HTTP.
- Chaque présentation et chaque flux média est identifié par un URL RTSP.
- Une présentation peut contenir plus d'un flux média.
- Le fichier de description de la présentation contient les codages, langage et d'autres paramètres qui permettent au client de choisir la combinaison la plus adéquate de médias.



Vidéo sur IP Ouverture d'une session « temps réel »

