

INF3050 Réseaux Informatiques

Bassem Haidar

Plan du cours

- Introduction, Modèle OSI et TCP-IP
- Couche Physique – Supports de transmission
- Couche Liaison – Ethernet
- Couche Réseaux – Adressage IPv4
- ARP - ICMP – DHCP
- Routage statique
- **Couche Transport (UDP - TCP)**
- **Introduction a la couche application**

Introduction

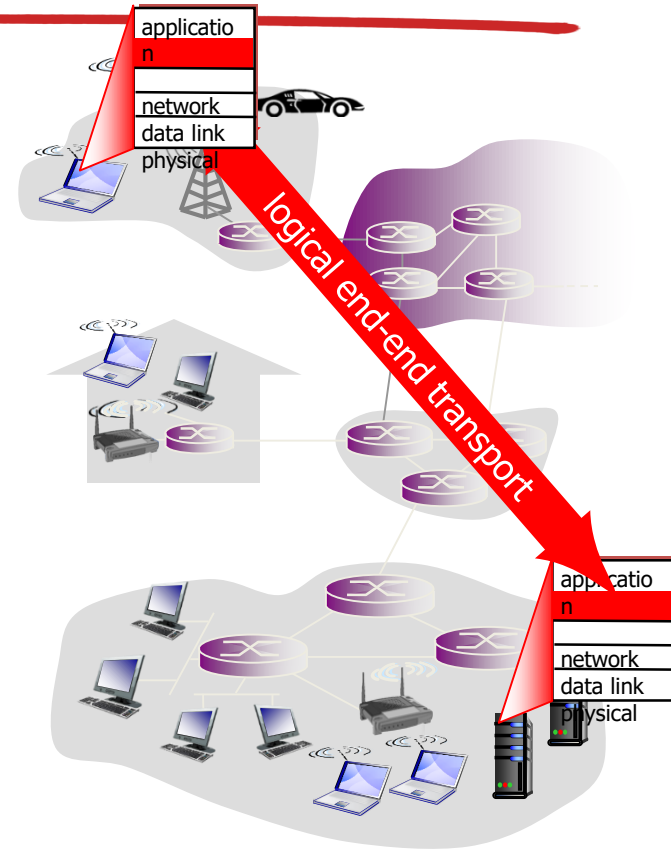
Couche Transport – Couche Application

Chapter 06

Couche de Transport

Services et protocoles de transport

- fournir une communication logique entre les processus d'application s'exécutant sur différents hôtes
- protocoles de transport exécutés dans les systèmes d'extrémité
 - côté envoi : divise les messages de l'application en segments, passe à la couche réseau
 - côté rcv : réassemble les segments en messages, passe à la couche d'application
- plusieurs protocoles de transport disponibles pour les applications
 - Internet : TCP et UDP



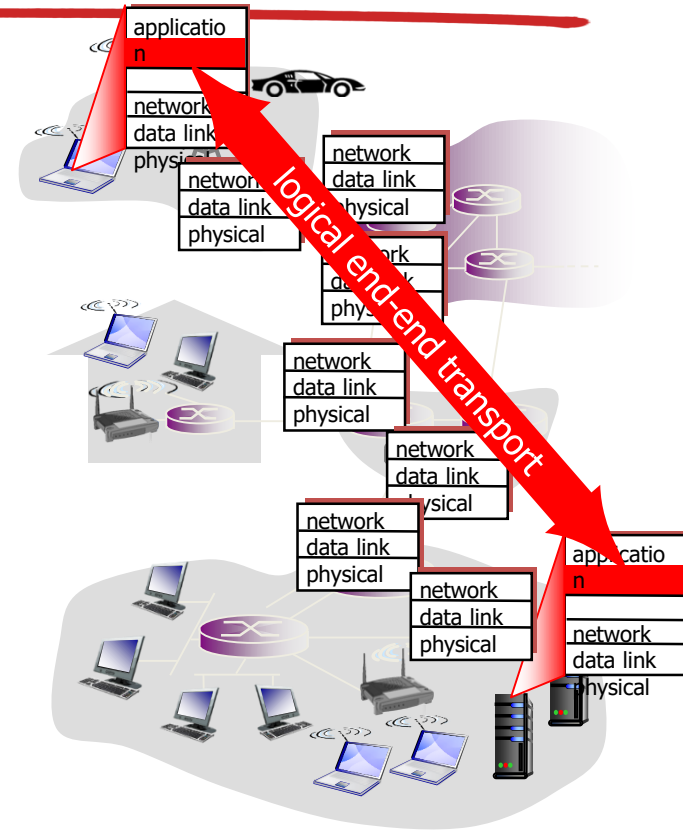
Transport vs. network layer

- couche réseau :
communication logique entre les hôtes
- Couche de transport : la communication logique entre les processus
 - repose sur des services de couche réseau améliorés

- **analogie :**
- 12 enfants dans la maison d'Ann envoyant des lettres à 12 enfants dans la maison de Bill :
- hôtes = maisons
- processus = enfants
- messages d'application = lettres dans des enveloppes
- protocole de transport = Ann et Bill qui démultiplexent leurs frères et sœurs en interne
- protocole de couche réseau = service postal

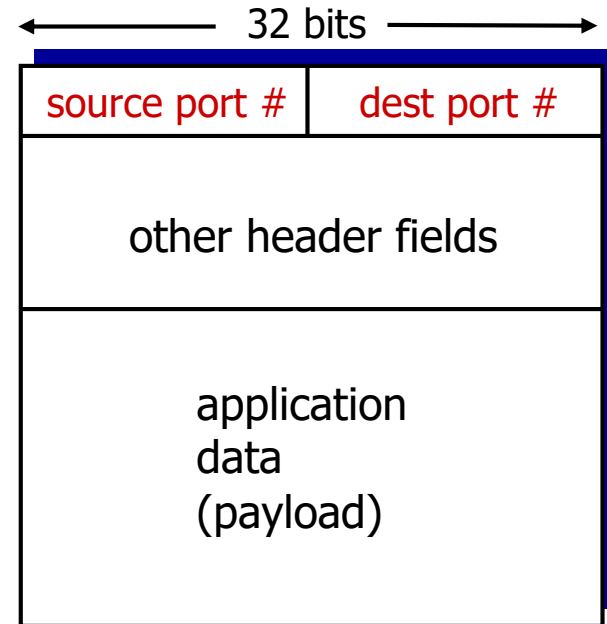
Internet transport-layer protocols

- livraison fiable et en ordre (TCP)
- contrôle de la congestion
 - contrôle de flux
 - configuration de la connexion
- livraison non fiable et non commandée : UDP
 - « au mieux », Best-effort
- services non disponibles :
 - garanties de retard
 - garanties de bande passante



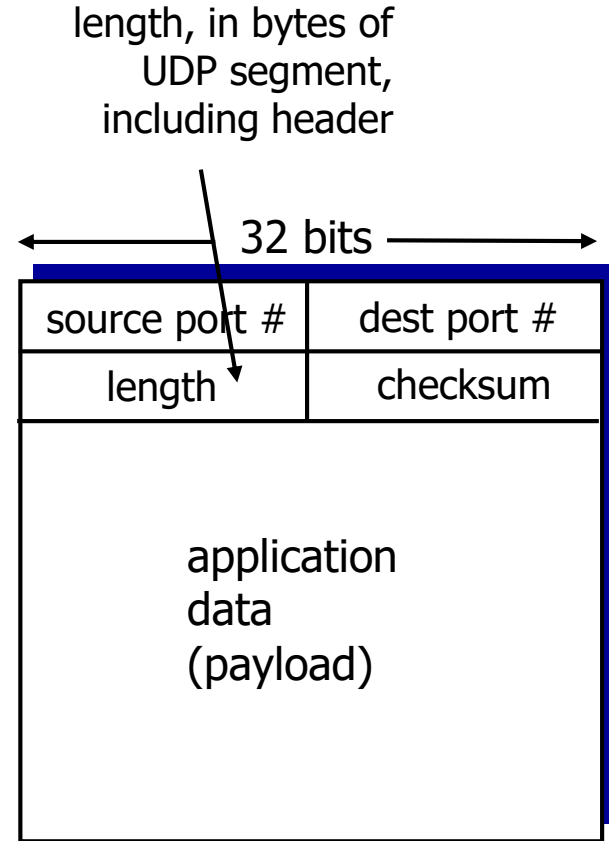
Comment fonctionne le démultiplexage

- l'hôte reçoit des datagrammes IP
 - chaque datagramme a une adresse IP source, une adresse IP de destination
 - chaque datagramme transporte un segment de couche transport
 - chaque segment a une source, un numéro de port de destination
- l'hôte utilise des adresses IP et des numéros de port pour diriger le segment vers le socket approprié



UDP : en-tête de segment

- **Why is there a UDP?**
 - no connection establishment (which can add delay)
 - simple: no connection state at sender, receiver
 - small header size
 - no congestion control: UDP can blast away as fast as desired

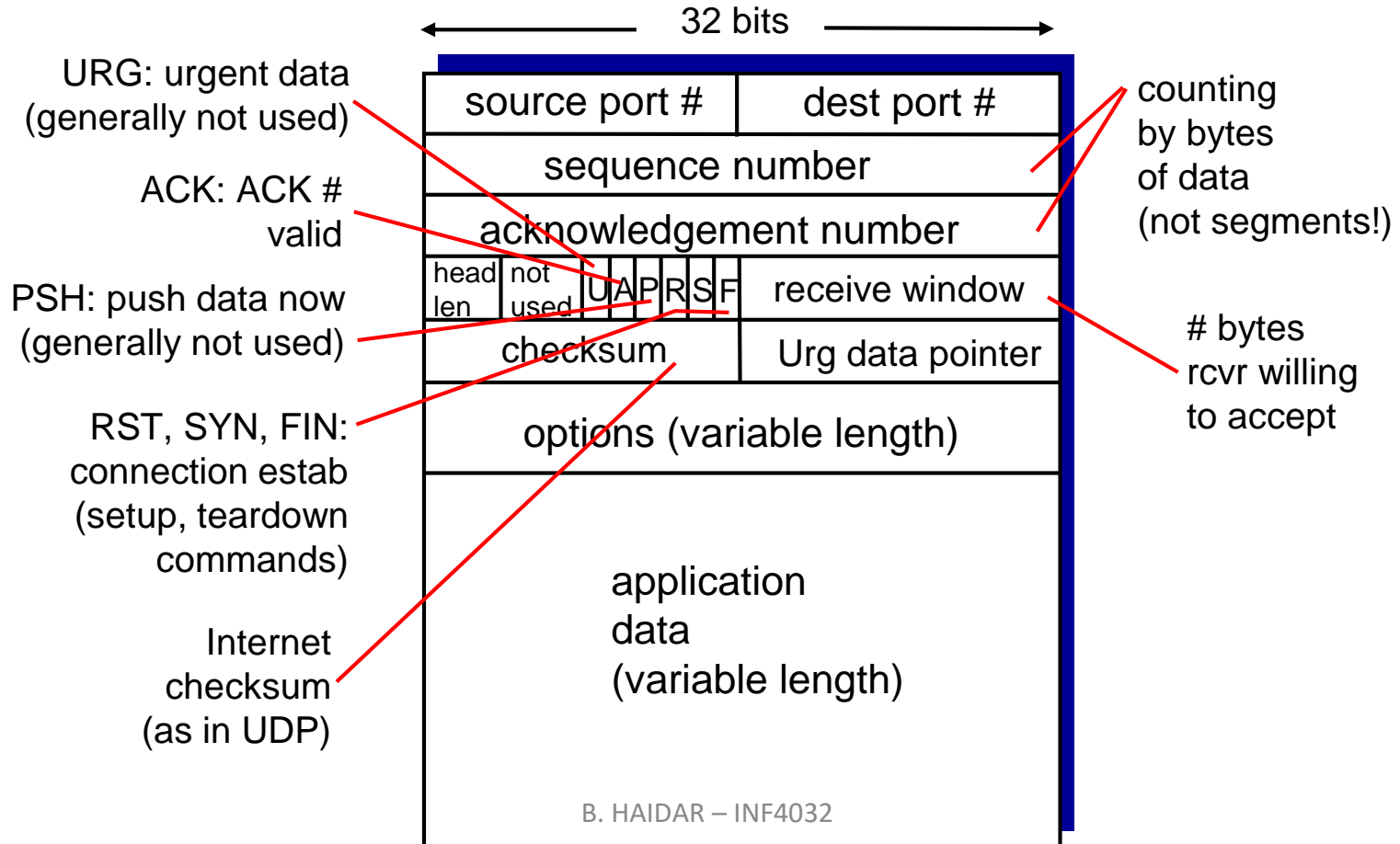


UDP segment format

TCP: Overview RFCs: 793,1122,1323, 2018, 2581

- données en duplex intégral :
 - flux de données bidirectionnel dans la même connexion
 - MSS : taille maximale des segments
- Connexion orientée:
 - établissement de liaison (échange de msgs de contrôle) dans son état émetteur, récepteur avant l'échange de données
- débit contrôlé, control de flux :
 - l'expéditeur ne submergera pas le destinataire
- point à point:
 - un expéditeur, un destinataire
- flux d'octets fiable et ordonnée :
 - pas de « limites de message »
- en pipeline :
 - contrôle de congestion et de flux - taille de la fenêtre

Structure des segments TCP



Accepter d'établir une connexion TCP 3-way handshake

client state

LISTEN

SYNSENT

ESTAB
B

choose init seq num, x
send TCP SYN msg

SYNbit=1, Seq=x

SYNbit=1, Seq=y
ACKbit=1; ACKnum=x+1

received SYNACK(x)
indicates server is live;
send ACK for SYNACK;
this segment may contain
client-to-server data

ACKbit=1, ACKnum=y+1

choose init seq num, y
send TCP SYNACK
msg, acking SYN

received ACK(y)
indicates client is live

server state

LISTEN

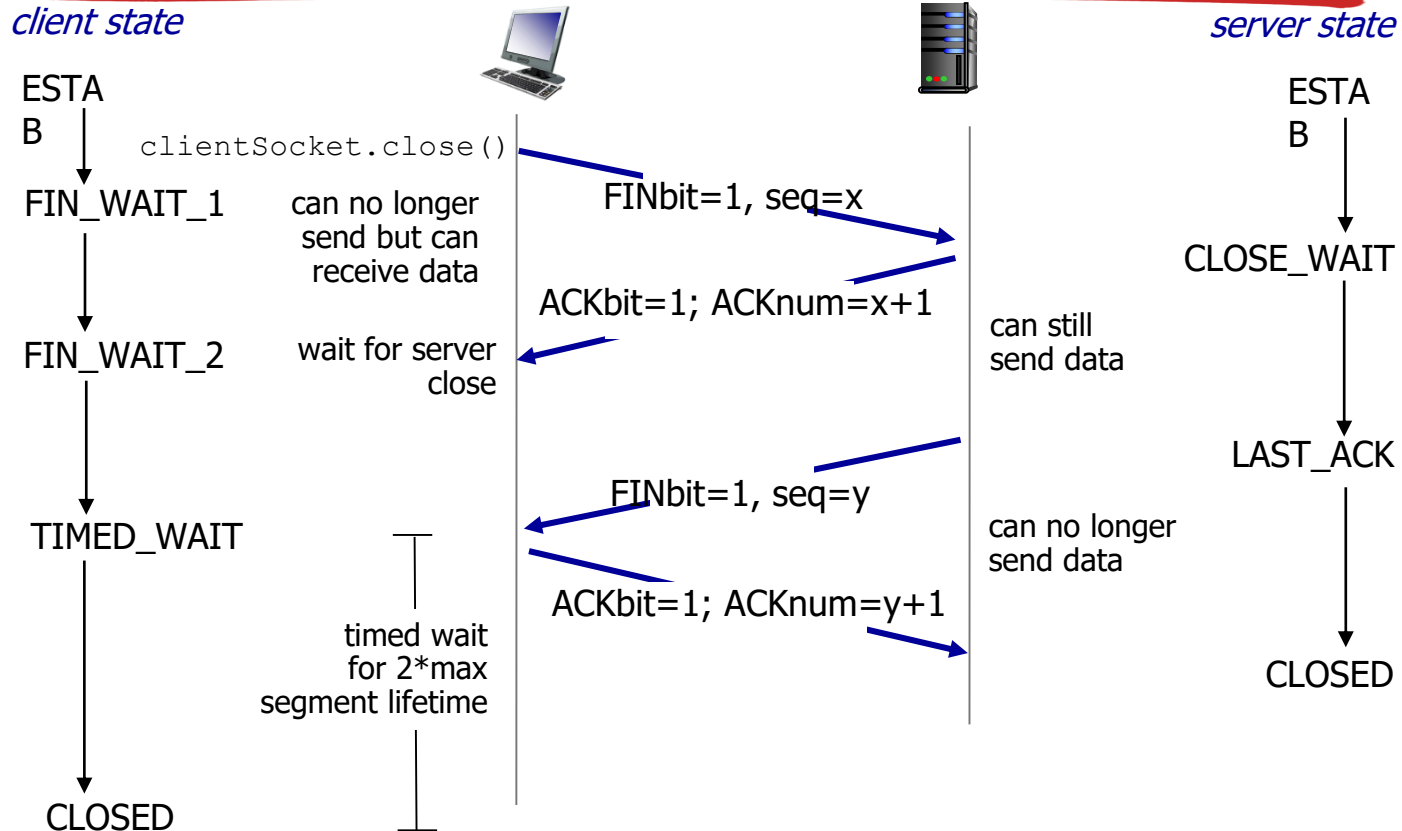
SYN RCVD

ESTAB
B

TCP : fermeture d'une connexion

- le client, le serveur ferment chacun leur côté de connexion
 - envoyer le segment TCP avec le bit FIN = 1
- répondre à FIN reçu avec ACK
 - à la réception de FIN, ACK peut être combiné avec son propre FIN
- les échanges FIN simultanés peuvent être gérés

TCP : fermeture d'une connexion



Introduction à la couche Application

HTTP – DNS

Web et HTTP

- la page Web se compose d'objets
- l'objet peut être un fichier HTML, une image JPEG, une applet Java, un fichier audio,...
- la page Web se compose d'un fichier HTML de base qui comprend plusieurs objets référencés
- chaque objet est adressable par une URL, par exemple,

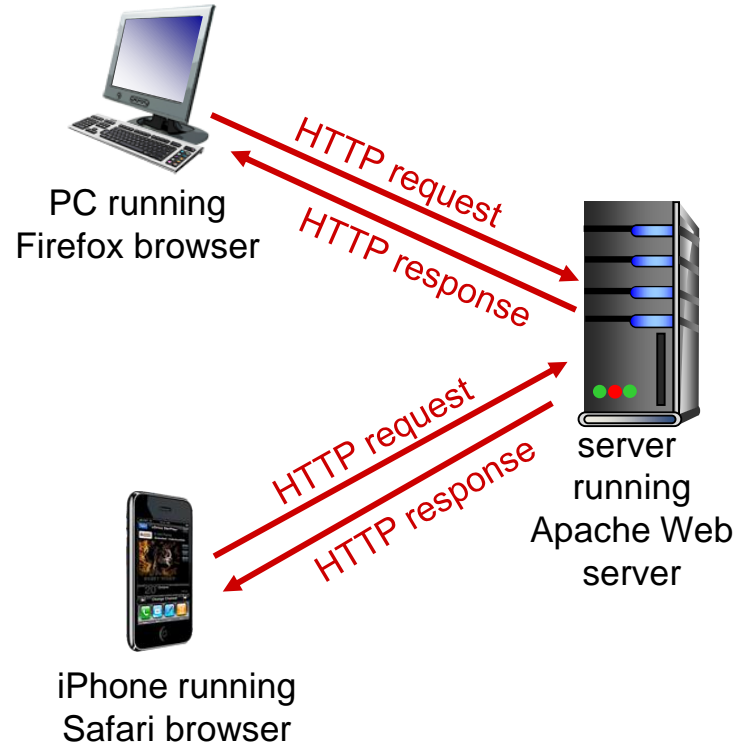
`www.someschool.edu/someDept/pic.gif`

host name

path name

Présentation HTTP

- HTTP : protocole de transfert hypertexte
- Protocole de couche application du Web
- modèle client/serveur
 - client : navigateur qui demande, reçoit, (en utilisant le protocole HTTP) et « affiche » des objets Web
 - serveur : le serveur Web envoie (en utilisant le protocole HTTP) des objets en réponse aux requêtes



HTTP overview (continued)

- utilise TCP :
- le client initie la connexion TCP (crée un socket) au serveur, port 80
- le serveur accepte la connexion TCP du client
- Messages HTTP (messages de protocole de couche application) échangés entre le navigateur (client HTTP) et le serveur Web (serveur HTTP)
- Connexion TCP fermée
- HTTP est « sans état »
- le serveur ne conserve aucune information sur les demandes passées des clients

aside

les protocoles qui maintiennent
« l'état » sont complexes !
l'histoire passée (état) doit être
maintenue
si le serveur/client tombe en panne,
leurs points de vue sur
« l'état » peuvent être
incohérents.

Message de requête HTTP

- deux types de messages HTTP : **requête, réponse**

- **Message de requête HTTP :**

— ASCII (human-readable format)

request line
(GET, POST,
HEAD commands)

header
lines

carriage return,
line feed at start
of line indicates
end of header lines

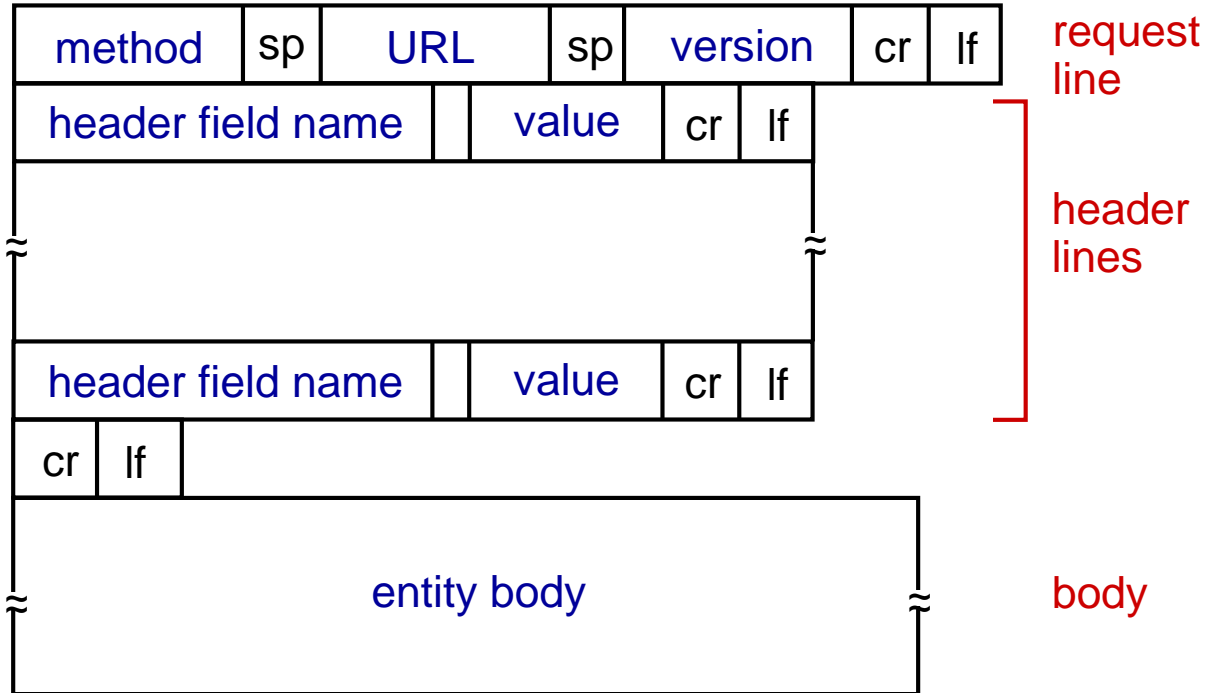
```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

carriage return character

line-feed character

* Check out the online interactive exercises for more
examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Message de requête HTTP : format général



Message de réponse HTTP

status line
(protocol
status code
status phrase)

header
lines

data, e.g.,
requested
HTML file

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02
GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-
1\r\n
\r\n
data data data data data ...
```

* Check out the online interactive exercises for more
examples: http://gaia.cs.umass.edu/kurose_ross/interactive

Application Layer

Codes d'état de réponse HTTP

- le code d'état apparaît sur la 1ère ligne dans le message de réponse serveur-client.
- quelques exemples de codes :
- **200 OK**
 - request succeeded, requested object later in this msg
- **301 Moved Permanently**
 - requested object moved, new location specified later in this msg (Location:)
- **400 Bad Request**
 - request msg not understood by server
- **404 Not Found**
 - requested document not found on this server
- **505 HTTP Version Not Supported**

DNS

DNS: domain name system

- personnes : de nombreux identifiants :
 - SSN, nom, passeport #
- Hôtes Internet, routeurs :
 - Adresse IP (32 bits) - utilisée pour l'adressage des datagrammes
 - "nom", par exemple, www.yahoo.com - utilisé par les humains
- Q : comment mapper entre l'adresse IP et le nom, et vice versa ?

Système de noms de domaines:

- base de données distribuée implémentée dans la hiérarchie de nombreux serveurs de noms
- protocole de la couche application : les hôtes, les serveurs de noms communiquent pour résoudre les noms (traduction d'adresse/de nom)

DNS: services, structure

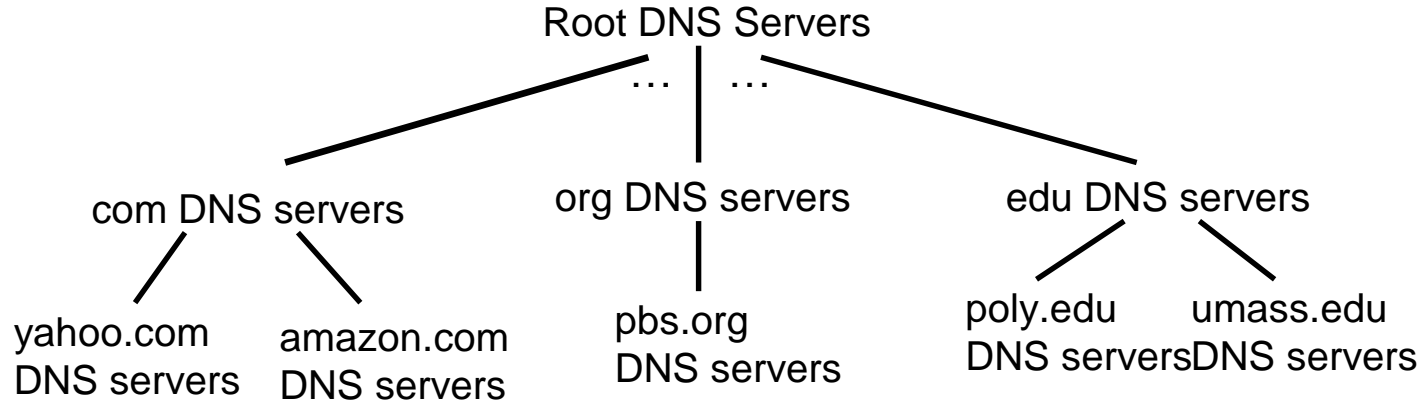
DNS services

- traduction du nom d'hôte en adresse IP
- alias d'hôte
 - canonique, noms d'alias
- alias de serveur de messagerie
- Répartition de la charge
 - serveurs Web répliqués : plusieurs adresses IP correspondent à un nom

pourquoi ne pas centraliser le DNS ?

- point de défaillance unique
- le volume de circulation
- base de données centralisée distante
- maintenance

DNS : une base de données distribuée et hiérarchique



le client veut une IP pour www.amazon.com ; 1ère approximation :

- le client interroge le serveur racine pour trouver le serveur DNS com
- le client interroge le serveur DNS .com pour obtenir le serveur DNS amazon.com
- le client interroge le serveur DNS amazon.com pour obtenir l'adresse IP de www.amazon.com

DNS name resolution example

- l'hôte de cis.poly.edu veut une adresse IP pour gaia.cs.umass.edu

requête itérative :

- le serveur contacté répond avec le nom du serveur à contacter
- "Je ne connais pas ce nom, mais demandez à ce serveur"

