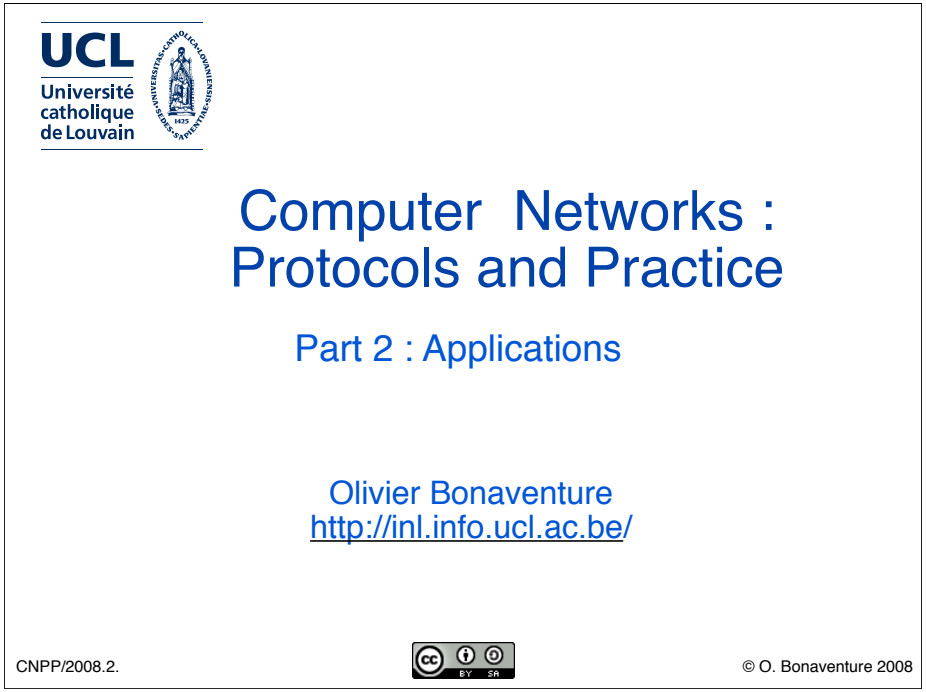
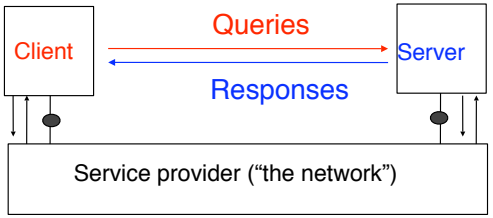


The Application Layer

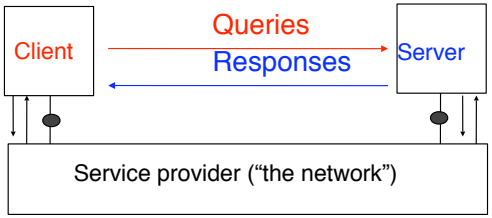
- Contents
  - The client-server model
  - Name to address resolution
  - email
  - world wide web
  - peer-to-peer applications

The client server model



- Client
  - interacts with server through transport layer
  - sends queries or commands
- Server
  - Answers the queries received from clients
  - Executes the commands from clients
  - Many clients can use the same server
- Example : email, www, ...

The client server model (2)

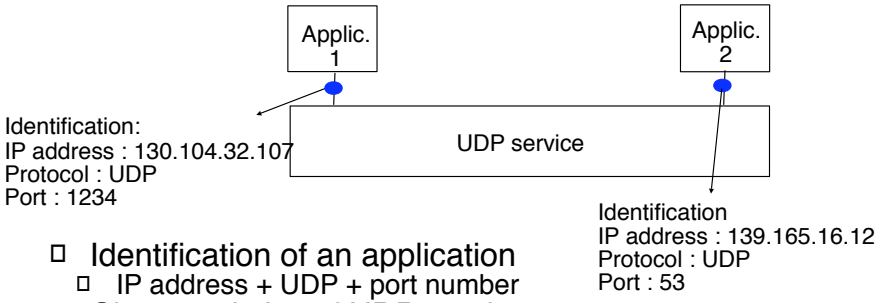


- Client and servers interact with service provider
- Both the client and the server must speak the same language
  - Application-level protocol : set of syntactical and semantical rules that define the messages exchanged between the client and the server and their ordering

Transport service on the Internet

- On the Internet, applications can use two different transport services
  - The service provided by the User Datagram Protocol (UDP)
    - unreliable connectionless service with error detection
  - The service provided by the Transmission Control Protocol (TCP)
    - reliable bytestream connection-oriented service

UDP service



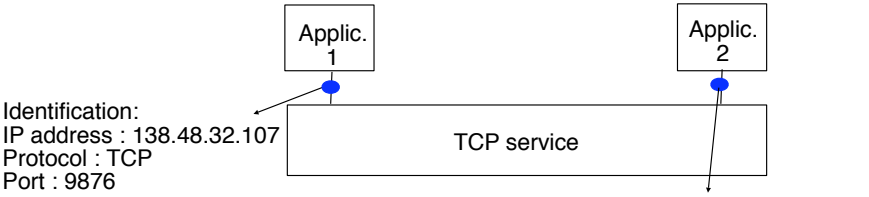
- Identification of an application
  - IP address + UDP + port number
- Characteristics of UDP service
  - connectionless
  - unreliable
    - messages can be lost
    - transmission errors can be detected but not recovered
    - sequence is not preserved
  - Maximum size of messages : almost 64 Kbytes

UDP is defined in

J. Postel, User Datagram Protocol. RFC768, August 1980

It will be described in more details later

TCP service



- Identification of an application
  - IP address + UDP + port number
- Characteristics of TCP service
  - connection-oriented
  - bidirectional
  - reliable
  - byte stream
  - connection release
    - abrupt if initiated by service provider
    - graceful or abrupt if initiated by user

TCP is defined in

J. Postel, Transmission Control Protocol, RFC793, September 1981

It will be described in more details later

Internet applications

- Contents
  - The client-server model
  - Name to address resolution
  - email
  - world wide web
  - peer-to-peer applications

Names and addresses

- Address of a server
  - IP Address of the host on which the server is running port number (TCP or UDP)
    - usually well known port number
- Drawback
  - Difficult to remember an IP address for a human
- Idea
  - Replace IP address by a hostname
  - Easier for humans
    - but IP address is necessary to contact server
  - How to translate a hostname in an IP address ?

Usually, a server always listens on the same port. A list of all reserved TPC ports may be found in : http://www.iana.org/assignments/port-numbers

Names and addresses (2)

- `hosts.txt` file
  - contains the name-address table
  - must be updated regularly

```
#
# Internet host table
#
127.0.0.1       localhost
138.48.32.99    babbage
138.48.32.100   leibniz
138.48.32.1     routeur
138.48.32.92    corneille
138.48.32.107   backus
138.48.20.152   arzach
138.48.32.137   almin01
138.48.32.170   duke
```

  - cannot be used in a large network

Hostnames

- Requirement
  - Host names should be unique
- How to achieve this in a scalable manner ?
  - Introduce hierarchy
  - Each hostname is composed of two parts
    - domain name (globally unique)
    - hostname (unique within a given domain)
  - How to uniquely distribute domain names ?
    - Introduce hierarchy
    - A small number of top-level domain names
    - Inside each top-level domain, allocate uniquely second level domain names
    - Inside each seconde-level domain, allocate uniquely either third-level domain names or host names,
    - ...

The Domain Name System was proposed in

P. Mockapetris and K. Dunlap. Development of the domain name system. In Proc. SIGCOMM'88, August, 1988. available from http://www.acm.org/sigcomm/ccr/archive/1995/jan95/ccr-9501-mockapet.html

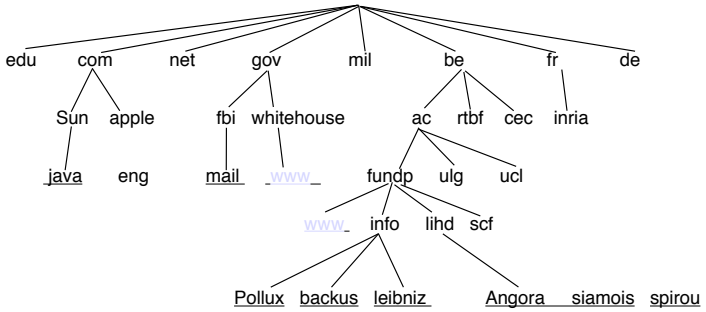
A more detailed description of the DNS protocols may be found in :

P. V. Mockapetris. Domain names - concepts and facilities. Request for Comments 1034, Internet Engineering Task Force, November 1987.

M. K. Stahl. Domain administrators guide. Request for Comments 1032, Internet Engineering Task Force, November 1987.

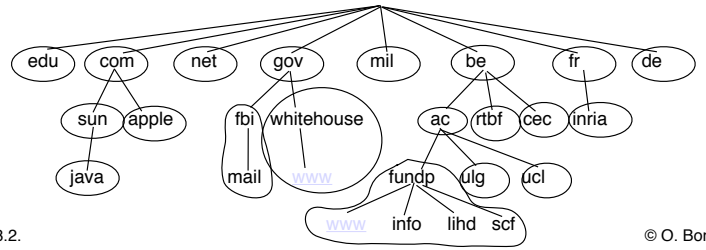
Host names and domain names

- Tree of all host names



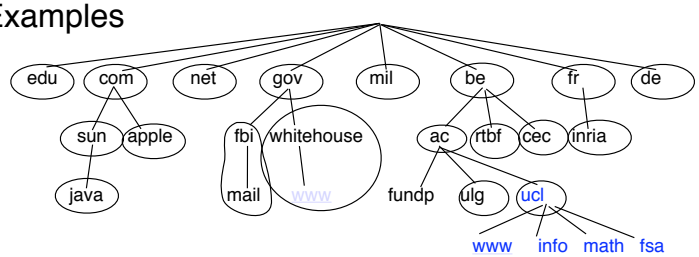
How to translate names into addresses ?

- How to efficiently translate a host name ?
  - By using a centralised database
    - there are more than 1 billion host names today
  - By using a distributed database
    - DNS : Domain Name System
    - relies on the hierarchy of domain names
    - there is one server responsible for each domain and this server must be queried to translate host names inside this domain



How to translate names into addresses ?

- Domain Name Service (DNS)
  - Each DNS server is responsible for a domain and knows
    - The IP addresses of all host names in this domain
    - The IP addresses of the DNS servers responsible for subdomains
  - Examples



    - java.sun.com
    - www.ucl.ac.be

There are many DNS server implementations available, the most widely used one is

http://www.isc.org/products/BIND

Many documents explain how to configure a DNS server, e.g.
http://www.tldp.org/HOWTO/DNS-HOWTO.html

DNS resolver

- To be able to translate name to addresses, a DNS implementation needs
  - to know actual list of IP addresses of root servers
  - to implement the DNS protocol and traverse the domain names hierarchy
  - Difficult to do this on all endhosts

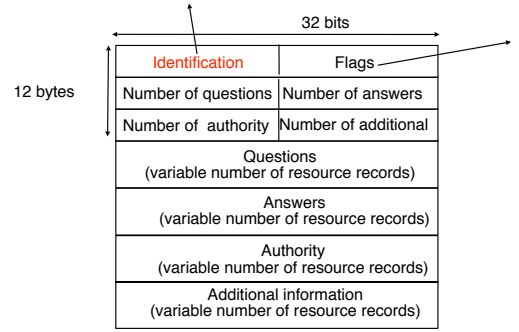
- Solution
  - DNS resolver
    - one resolver for a set of endhosts
    - maintains up-to-date list of IP addresses of root servers
    - implements DNS protocol
  - endhosts
    - only need to be able to send DNS requests to resolver
    - must know IP address of closest DNS resolvers

DNS : optimisations

- Reduce risk of failures
  - several root-servers
  - server DNS servers authoritative for each domain
  - each endhost can send queries to multiple resolvers
- Improved performance
  - avoid sending several times the same query
  - cache memory on DNS resolvers containing
    - recent name-addresses translations
    - addresses of DNS servers recently contacted
- DNS protocol
  - usually runs over UDP
  - sometimes is also used over TCP

DNS : message format

Each DNS request contains a number that will be returned in the response by the server to allow the client to match the request.



- Question/Response
- Recursive question or not
- Authoritative answer or not
- Possible error

DNS : resource records

- Each DNS messages is composed of resource records (RR) encoded as TLV
  - < Name, Value, Type, TTL>
  - Types de RR
    - A (Address)
      - Name is a hostname and Value an **IPv4 address**
    - AAAA (Address)
      - Name is a hostname and Value an **IPv6 address**
    - NS (NameServer)
      - Name is a domain name and Value is the hostname of the DNS server responsible for this domain
    - MX (Mail Exchange)
      - Name is a domain name and Value is the name of the SMTP server that must be contacted to send emails to this domain
    - Type CNAME
      - Alias

Lifetime of the RR in server's cache

The RR MX were proposed in

C. Partridge. Mail routing and the domain system. Request for Comments 974, Internet Engineering Task Force, January 1986.

A complete list of DNS RR may be found at
http://www.its.uq.edu.au/tn-0011

Internet applications

- Contents
  - The client-server model
  - Name to address resolution
  - email
  - world wide web
  - peer-to-peer applications

Internet applications

- Contents
  - The client-server model
  - Name to address resolution
  - email
  - world wide web
  - peer-to-peer applications

FTP : File Transfer Protocol

- Protocol from the old days
  - allows a client to send/retrieve files from a server
- Problems solved by FTP
  - User authentication
    - username, password
  - Filesystem traversal
    - browse directories on server and locate files
  - file transfer
    - to or from the server

CNPP/2008.2.

FTP is defined in

J. Postel and J. K. Reynolds. File transfer protocol. Request for Comments 959, Internet Engineering Task Force, October 1985.

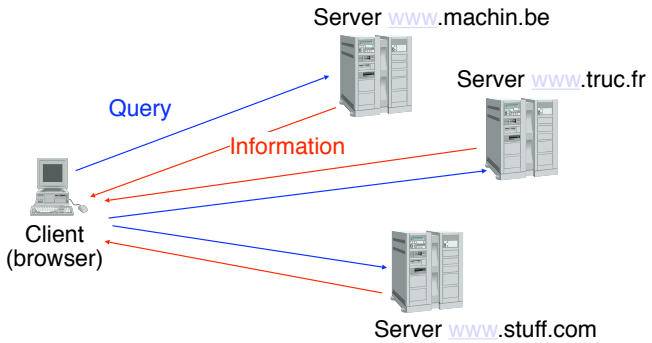
There are many ftp clients and servers
WU-ftpd http://www.wu-ftpd.org/

FTP : main commands

- Main commands
  - USER <user>
    - username, ftp for anonymous access
  - PASS <pass>
    - allows user to send password associated to username
  - SYST
    - information about type of server
  - CWD <path>
    - directory traversal
  - STOR <file>
    - save file in the current directory on server
  - RETR <file>
    - retrieve file from current directory on server
  - PORT <B1,B2,B3,B4,B5,B6>
    - use TCP connection on port B5*256+B6 on B1.B2.B3.B4

World Wide Web

- Goals
  - Allow browsers to browse hypertext documents stored on multiple servers



CNPP/2008.2. © O. Bonaventure 2008

The most detailed textbook on HTTP and related protocols

B. Krishnamurthy and J. Rexford. Web protocols and practice : HTTP/1.1, networking protocols, caching and traffic measurement. Addison Wesley, 2001.

The web technologies and protocols are standardised by the W3C :
http://www.w3c.org

World Wide Web (2)

- The five key elements of www

1. An addressing scheme that allows to identify any document stored on a server
   - URL : Uniform Resource Locator
2. An hypertext language that allows to easily write documents with hypertext links
   - HTML : HyperText Markup Language
3. An efficient and lightweight application-level protocol to exchange documents
   - HTTP : HyperText Transfer Protocol
4. Servers
5. Clients (browsers)

URLs are defined in

T. Berners-Lee, L. Masinter, M. McCahill, Uniform Resource Locators (URL) , December 1994 , RFC 1738

Voir également :

http://www.w3.org/Addressing

Pour plus d'informations sur HTML et son évolution, consultez :

http://www.w3.org/MarkUp

De nombreux navigateurs existent. Parmi les serveurs http, un des plus courants est :
apache : http://www.apache.org

Uniform Resource Locator (URL)

- Uniform Resource Locator (URL)
  - generic syntax : <protocol>://<document>
    - protocol used to retrieve document from server
      - http is the most common one but others are frequently used
    - document indicates the server and the location of the document
    - *<user>:<password>*@<server>*:<port>*/<path>
      - <user> : optional username
      - <password> : optional password
      - <machine> : hostname or IP address of the server that hosts the document
      - <port> : optional port number
      - <path> : document location on server
    - examples
      - http://www.info.ucl.ac.be
      - http://alice:secret@inl.info.ucl.ac.be:80/index.html

HTML

- HyperText Markup Langage
  - Language used to encode documents on the web
  - Keywords
    - <HTML>...</HTML>
    - <HEAD>...</HEAD>
    - <BODY>...</BODY>
    - <TITLE>..</TITLE>
    - <B>...</B>
    - <I>...</I>
    - <H1>...</H1>
    - <P>
    - <HR>
    - <UL>...</UL>
    - <OL>...</OL>
    - <IMG SRC="URL">
    - <A HREF="URL">*text anchor*</A>

HTML (2)

- Example

```
<HTML>

<HEAD>
<TITLE>HTML test page</TITLE>
</HEAD>

<BODY>
<IMG SRC="http://www.images.be/logo.gif">
<H1>Web servers from UCL UCL<P></H1>
<HR>
<UL>
 <LI><A HREF="http://www.uclouvain.be">UCL</A>
 <LI><A HREF="http://www.info.ucl.ac.be">CSE Dept.</A>
 <LI><A HREF="http://www.math.ucl.ac.be">Math</A>
</UL>
</BODY>

</HTML>
```

Header

Body

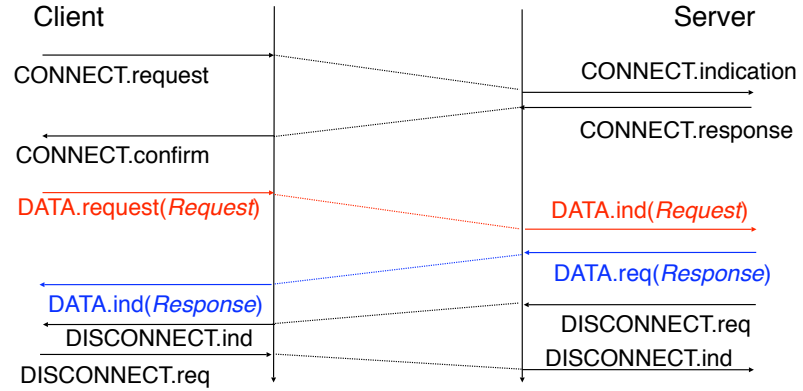
Image on remote server

First level title

External hypertext link

Information transfer www

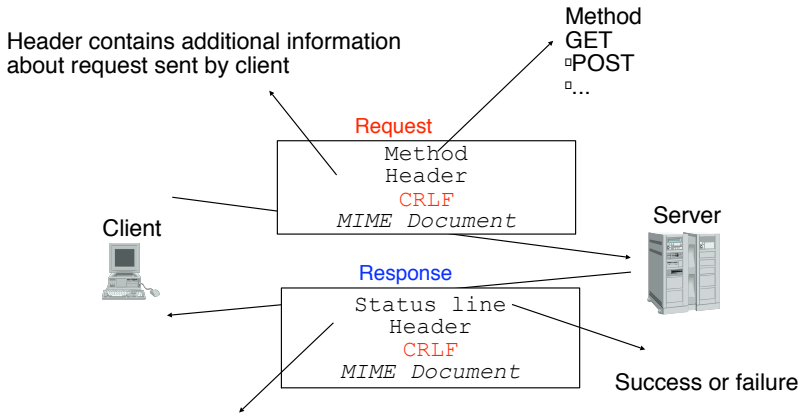
- HTTP 1.0 - non-persistent connection
  - Principle
    - relies on TPC service (default port : 80)
    - Client sends request, server sends reply



HTTP 1.0 is defined in :

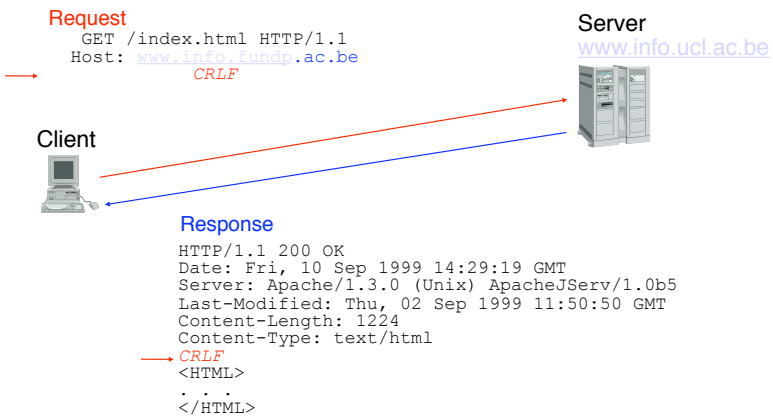
T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext transfer protocol -- HTTP/1.0. Request for Comments 1945, Internet Engineering Task Force, May 1996.

HTTP



- HTTP is a stateless protocol, server does not maintain any state from one request to another
  - POP, FTP, SMTP are examples of stateful protocols in contrast

HTTP : Example



HTTP : Methods

- Methods
  - GET
    - method used to request a "document" stored on server
      - GET <document> HTTP/1.0
    - example
      - `GET /index.html HTTP/1.0`
  - POST
    - method used to send a "document" to a server
      - document is part of the request and encoded as a MIME document

HTTP : Request headers

- Request headers
  - Allow to add information about the client or the request
  - `Host: <name>`
    - Name of the server where the document is stored
  - `Authorization`
    - allows to perform access control
  - `If-Modified-Since: <date>`
    - server will only send the requested document if the document is more recent than date
  - `Referer: <url>`
    - Information, indicates the URL visited by the client before this request
  - `User-Agent: <agent>`
    - information, indicates the browser used on the client

HTTP : Status line

- Status liine
  - Format : `Version_HTTP Code Comment`
  - Success/Failure
    - `1xx` : For information (unused)
    - `2xx` : Success
      - Example : `HTTP/1.0 200 OK`
    - `3xx` : Redirection
      - Request could not be handled on local server and should be sent to another server
      - Example :
        - `HTTP/1.0 301 Moved permanently`
        - attached MIME document will contain URL of document
    - `4xx` : Client-side error
      - examples
        - syntax error, unreachable URL, unauthorised, ...
    - `5xx` : Server-side error
      - examples :
        - internal error, method not implemented on server, ...

HTTP : Response headers

- Header
  - Optional information about the server, the response or the document attached to the response
  - Date
    - date of the document attached to response
    - example : `Date: Wed, 05 Sep 2001 13:27:34 GMT`
  - Server
    - Name and version of http server used
    - example :
      `Server: Apache/1.3.20 (Unix)ApacheJServ/1.1.2 PHP/4.0.6`
  - Content-*
    - MIME header of the attached document
    - example :
      `Content-Length: 5891`
      `Content-Type: text/html`

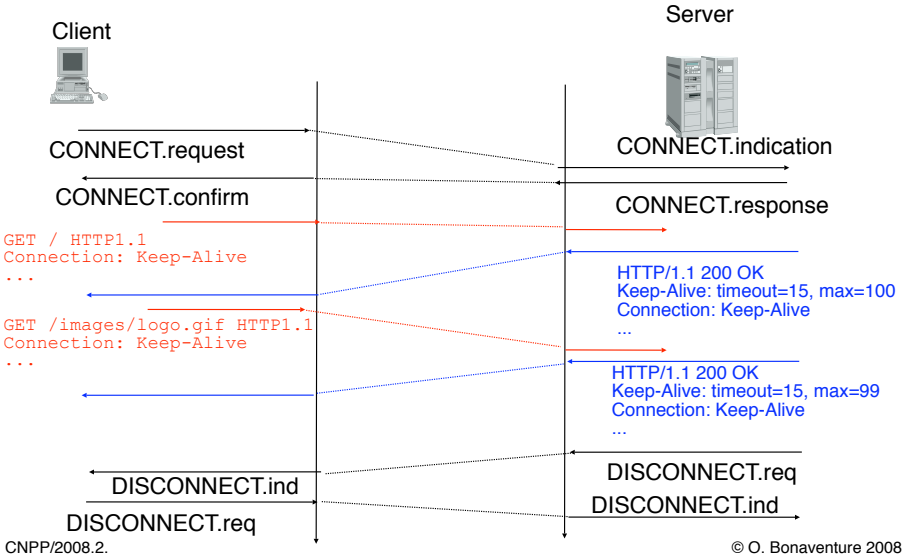
HTTP 1.1

- HTTP 1.0
  - a single TCP connection used to transmit a single document (html file, image,...)
    - the establishment and release of the TCP connection induce a significant overhead, in particular for small pages
- HTTP 1.1
  - uses a single persistent TCP connection
    - This TPC connection can be used for several requests and the corresponding responses
    - the cost of establishing and releasing the TCP connection is amortised over multiple requests
    - Although HTTP 1.1 uses a single TCP connection for multiple requests, HTTP 1.1 remains stateless

HTTP 1.1 is defined in :

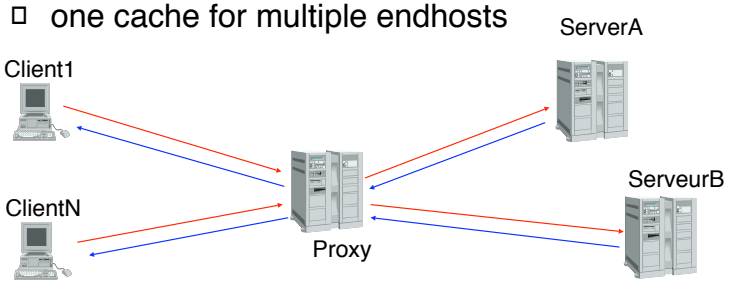
R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol -- HTTP/1.1. Request for Comments 2616, Internet Engineering Task Force, June 1999.

HTTP 1.1 : Persistent connection



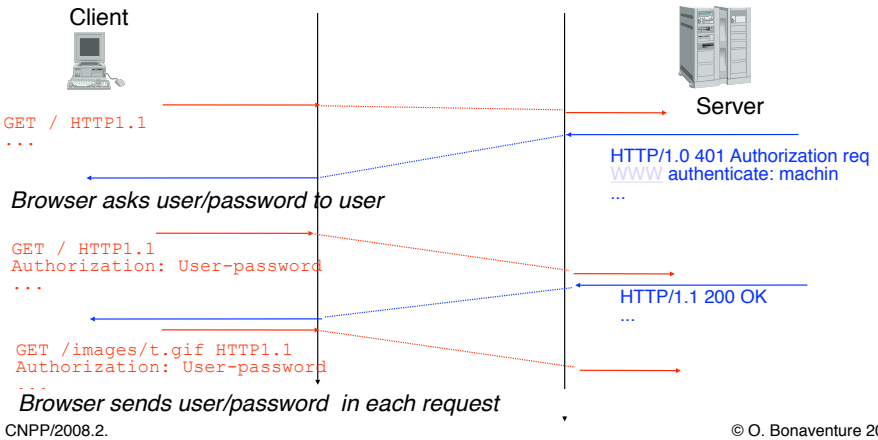
Improving performance

- Observation
  - Many pages are requested multiple times or from close endhosts
- Solution
  - local cache on each client
    - `if-modified-since` header helps
  - one cache for multiple endhosts



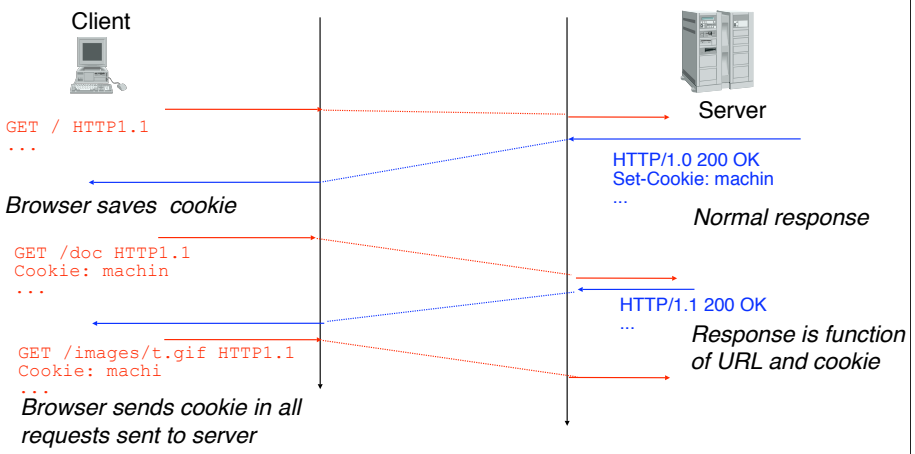
HTTP Authentication

- Example



HTTP Cookies

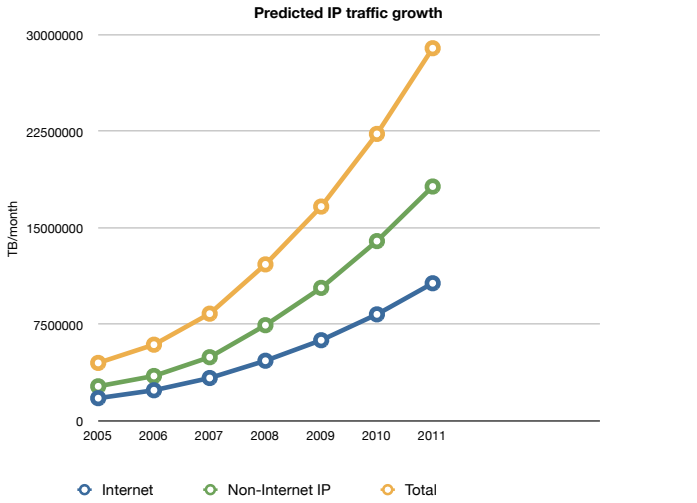
- Example



Internet applications

- Contents
  - The client-server model
  - Name to address resolution
  - email
  - world wide web
  - peer-to-peer applications

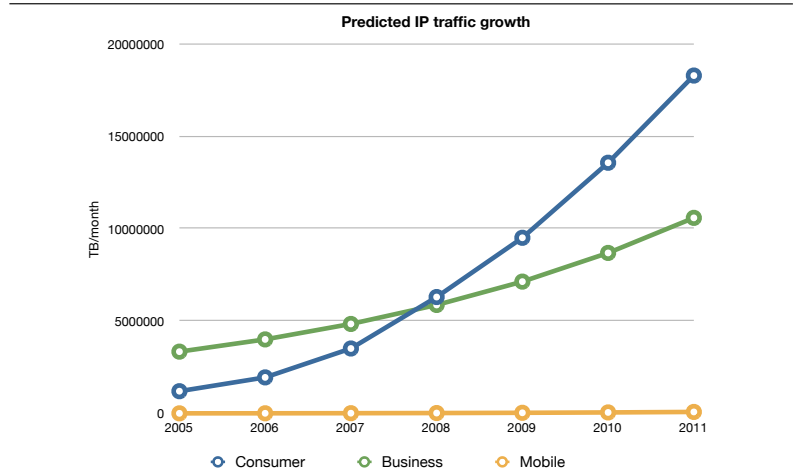
Predicted IP traffic growth



CNPP/2008.2.

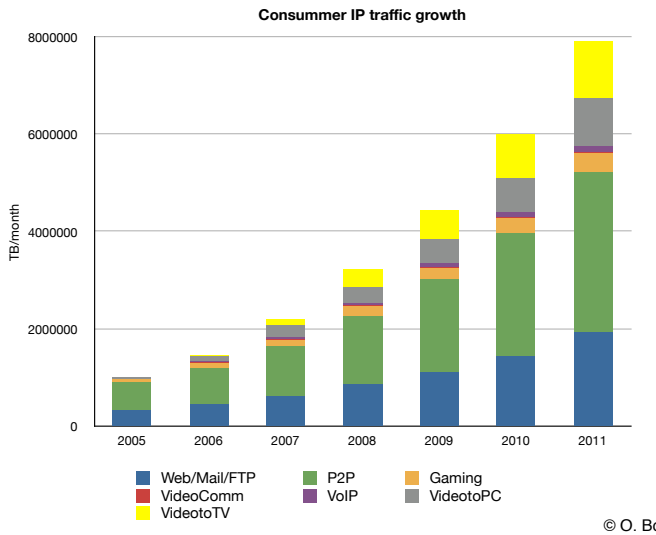
Source for this data Global IP traffic forecast and methodology, 2006-2011, Cisco 2007 white paper, [online version] http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/net_implementation_white_paper0900aecd806a81aa.pdf

Predicted IP traffic growth



CNPP/2008.2.

Predicted IP traffic growth (3)



Peer-to-peer file sharing

- Evolution of file sharing on the Internet
- Servers using `ftp` protocol
  - A single server that serves all files on disk
  - A set of mirror serves serving the same content
- The innovation introduced by Napster
  - How to distribute many files from many nodes ?
    - Keep the files on their source nodes
    - Central Napster server stores description and URL of each shared file
    - Users willing to obtain a file consult central server to obtain file URL and then download file from their respective source nodes
      - server remains simple and can index large number of files
      - server does not directly participate in file transfer

Peer-to-peer file sharing (2)

- Limitations of the Napster approach
  - a single server indexes all files
  - if a source node fails, then the ongoing file transfers must be restarted
    - completely or partially depending on the file transfer protocol begin used for the transfer
  - performance of file transfer is function of performance of the corresponding source node
    - if source node is connected via ADSL, performance will be severely limited
- How to improve ?
  - Divide the file in blocks
  - Each block can be served by multiple nodes
    - provides redundancy
  - Download from several nodes at the same time
    - one TCP connection may be slow and others faster

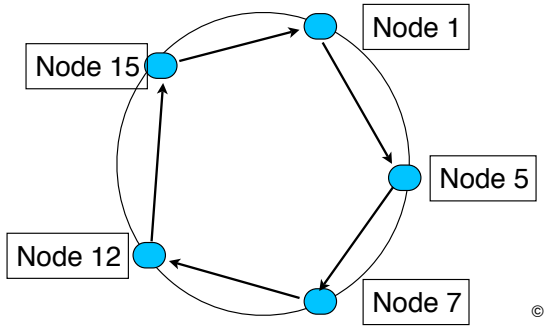
Today, Napster does not work anymore as explained due to copyright violations reasons.

One of the most efficient file transfer protocol used today is Bittorrent. Bittorrent also divides files in blocks and allows files to be downloaded from several nodes at the same time. This provides good redundancy in case of node/link failures, but also allows an efficient utilisation of the available link bandwidth by using uncongested paths (the node with the highest bandwidth will automatically serve blocks faster than a congested node). A Bittorrent node will not necessarily receive blocks in sequence. Furthermore, to ensure that all Bittorrent users contribute to the system, Bittorrent implementations apply the tit-for-tat principle which implies that once a node has received a block, it must serve this block to other nodes before being allowed to download new blocks.

Additional information about the Bittorrent protocol may be found in

Distributed Hash Table based P2P

- How to scale file sharing to a very large number *n* of nodes ?
- Principle of the solution
  - Use a hash function such as SHA-1
  - Each node has one identifier, id=hash(IPaddress) and a pointer to its successor on the Chord ring

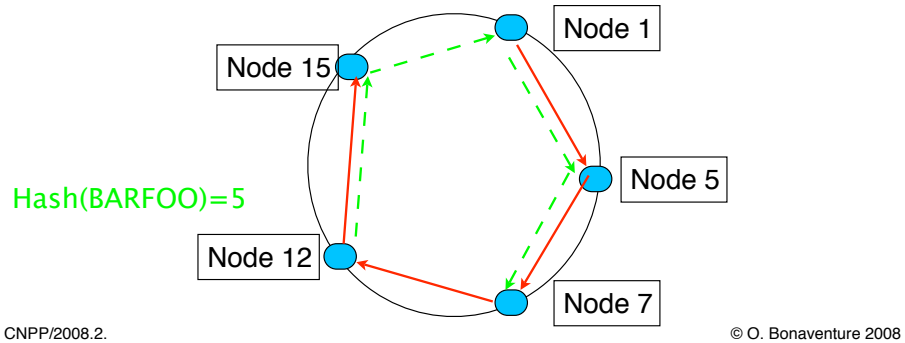


Several Distributed Hash Tables have been proposed in the literature. One of the first and most influential ones is Chord

Ion Stoica, Rober t Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan , " Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications" ACM SIGCOMM 2001

How to store files ?

- Principle
  - File FOOBAR is stored on the node whose id is the successor of hash("FOOBAR") on the ring
  - A node on the ring uses its successors to find the responsible node for a given file
    - Examples

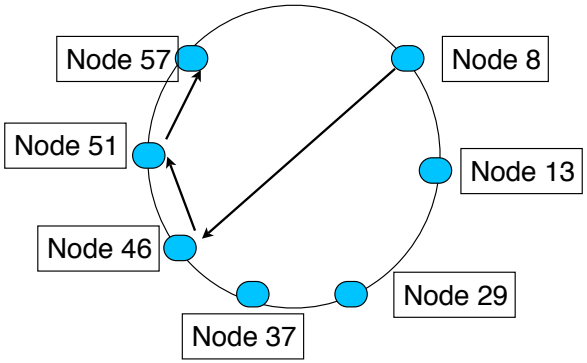


How to find files faster ?

- Performance of file lookup
  - O(n)
    - worst case is to follow linked list of n nodes
- How to improve ?
  - Allow nodes to know addition pointers to other nodes on the Chord ring to speedup lookup
    - m = number of bits in the key/node identifiers
    - Each node maintains routing table of m entries
    - The ith entry in the table at node n contains the identity of the first node, s, that succeeds n by at least 2^i–1 on the identifier circle, i.e., s = successor(n + 2^i–1)
      - arithmetic modulo 2^m is used
    - A finger table entry includes both the Chord identifier and the IP address (and port number) of relevant node.

Scalable lookup with Chord

- Example
  - m=8



N8+1	N13
N8+2	N13
N8+4	N13
N8+8	N29
N8+16	N29
N8+32	N46

N46+1	N51
N46+2	N51
N46+4	N51
N46+8	N57
N46+16	N57
N46+32	N8

- How to find key 53 from node 8 ?

Summary

- Client-server model
- UDP and TCP services
- Application-level protocols
  - DNS
    - relies on UDP, stateless
  - SMTP, POP, FTP
    - rely on TPC, statefull
  - HTTP
    - relies on TCP, stateless
- Peer-to-peer applications