

# Network Applications

# Transmission control protocol (TCP)

- **Connections.** Two computers set up a connection to exchange data. The systems synchronize with one another to manage packet flow and adapt to congestion in the network.
- **Full duplex operation .**A TCP connection is a pair of virtual circuits (one in each direction). Only the two end systems can use the connections.
- **Error cheking.** A checksum techinque is used to verify that packets are not corrupted.
- **Sequencing.** Packets are numbered so that the destination can reoder packets and determine if a packet is missing.
- **Aknoledgements.** Upon receipt of one or more packets, the receiver returns a acknowledgement (called an “ACK”) to the sender indicating that it received the packets. If the packets are not ACKed, the sender may retransmit the packets (or tetminate the connection ifit thinks the receiver ha crashed).

- **Flow control.** If the sender is overflowing the receiver by transmitting too quickly, the receiver drops packets. Failed ACKs alert the sender to slow down or to stop sending.
- **Packet recovery services.** The receiver can request retransmission of a packet. Also, if the packet receipt is not ACKed, the sender will resend the packets.

## User Datagram Protocol (UDP)

- No connection required.
- Since UDP does not return ACKs, the receiver cannot signal that packets have been successfully delivered. Lost packets are not retransmitted.
- UDP does not insert sequence numbers. The packets are expected to arrive as a continuous stream or they are dropped.
- ACK, which are used in TCP to control packet flow, are not returned.
- Without ACKs the network cannot signal congestion to the sender.

## **Session protocol**

Session protocol specifies how to establish a communication session with a remote system. Specifications for security details such as security using passwords belong in this protocol level.

## **Presentation protocol**

Presentation protocol specifies how to represent data. Such protocols are needed because different brand of computers use different internal representations for integers, and characters. Presentation protocols are needed to translate from the representation on a computer to the representation on another.

## **Application protocol**

Application protocol specifies how one particular application uses a network. The protocol specifies the details of how an application program on one machine makes a request (e.g., how to specify the name of the desired file) and how the application on another machine responds

- The TCP protocol controls that the transmission between two end points is correct. Does not establish which end point must initiate the communication. (*peer to peer communication*)
- This job is performed by the applications that utilizes TCP for the data transmission by using a specific protocol named **application protocol**.
- The application protocols are based on the *client-server model*. A end point begins the communication (client) and the other (server) receives the messages.
- **Specific port numbers** are assigned to the more common application protocols. For instance, to a server web process (**HTTP protocol**) the port number 80 is assigned..
- To a mail server (**SMTP protocol**) the port number 25 is assigned
- A set of well known standard Internet application protocols is reported in the *RFC 1700*. When a programmer creates a new application, a new port number must be assigned to it.

## Service identification

- TCP uses a 16-bit integer values (proto) to identify services, and assign a unique port number to each service.
- A server specify the protocol port for the service it offers, and then wait passively for communication.
- A client specifies the protocol port number of the desired when sending a request. TCP software on the server's uses the protocol port number in a incoming message to determine which server should receive the request.

- **Applications protocols**

- different kinds of exchanged messages ( request messages, answer messages)
- syntax of exchanged messages (fields and their use)
- semantic of the fields (rules to determine when and how a process sends messages or answers to messages)



## **Application protocols**

SMTP (Simple Mail Transfer Protocol)

FTP (File transfer Protocol)

HTTP (Hypertext Transfer Protocol )

SNMP (Simple Network Management Protocol)

They use the TCP protocol for sending messages.

## WWW application

The application is constituted by

- Browser (Microsoft Internet Explorer, Firefox, Safari, Opera,..)
- Web server (Apache, IIS Internet Information System,...)
- Application protocol HTTP (Hypertext Transfer Protocol )

## CLIENT WEB

the browser:

- Receives and executes the user comandns
- Sends a request to the web server in order to receive a document.
- Interprets different kinds of encoding formats (HTML, GIF, JPEG; MP3..) and presents the content of the document to the user.

## WEB SERVER

- The web server attends to the management, retrieval and sending of the documents required by the clients.
- When the user requests a document by using a link or by defining the address of the document, the client sends a request HTTP to a particular server with the indication of the required document.
- The web server interpretes the request, finds the required document and send it to the client by using **HTTP** and indicating the document type.

- When a user invokes a browser, the user must specify an initial page to view. Each page managed by the server is uniquely identified through a syntactic form that incorporates all the information needed to specify a remote item.
- The syntactic form encodes the information in a character string known as a *URL (Uniform Resource Locator)* whose general form is :

protocol://computer-name:port/document-name

*protocol* is the name of the protocol used to access the document, *computer-name* is the domain name of the computer on which the document resides, *port* is an optional protocol port number, and *document name* is the name of the document on the specified computer

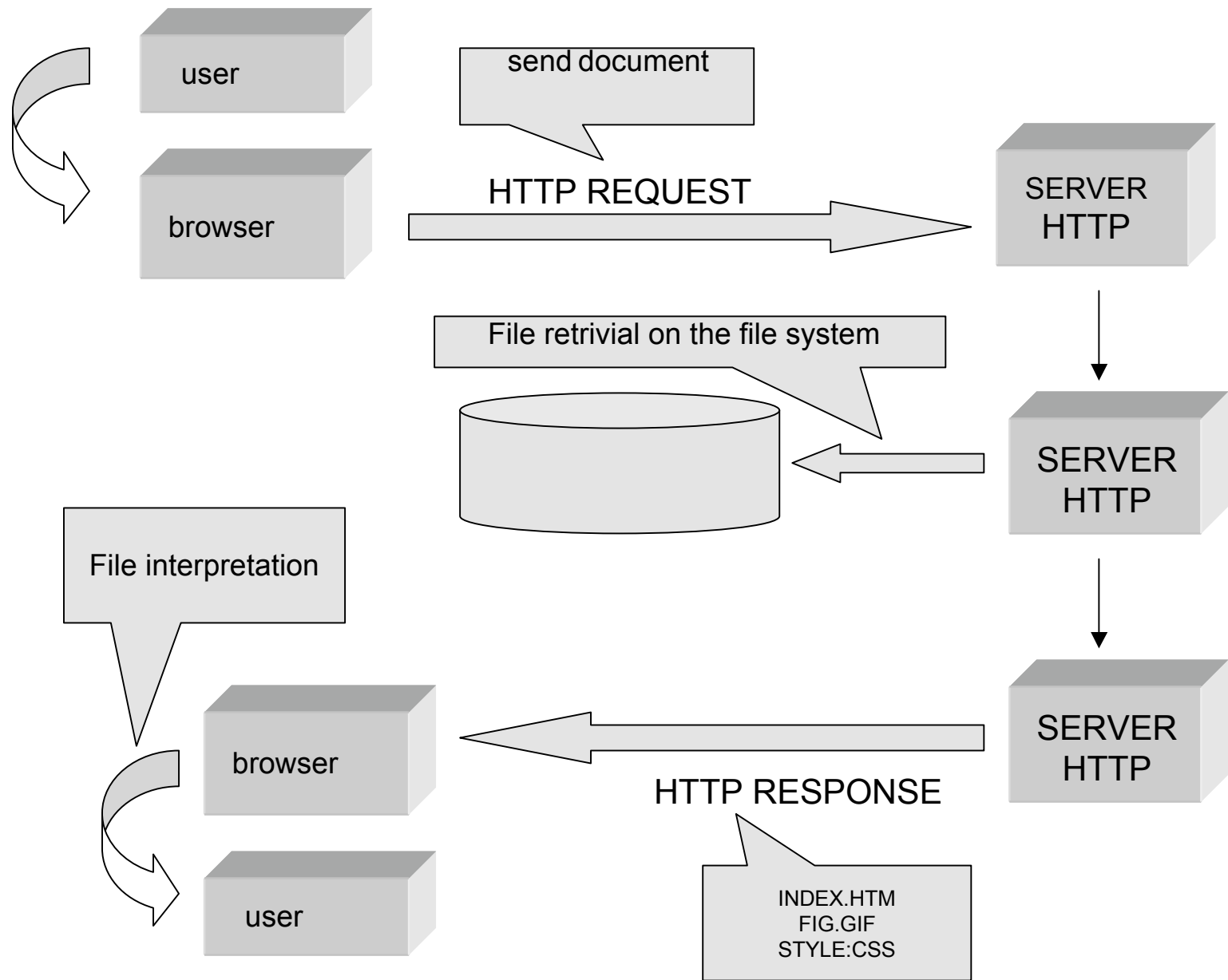
The protocol used is **HTTP (*HyperText Transfer Protocol*)**.

Esempio:

[http: // www.unibo.it/Portale/Master/default.htm](http://www.unibo.it/Portale/Master/default.htm)

- [www.unibo.it](http://www.unibo.it): the name of the computer (home page)  
*master* : the name of the required document

- HTTP: application protocol



- If the page is a simple document it is stored **as a sequence of characters** .
- To allow that , when reproduced on the video of the client, the document is represented as desired (subdivision on paragraphs, title and subtitles, particular highlighting of some parts etc..), it is necessary to describe such structure through the commands of *a mark up language*.
- The standard in this field is **HTML** (*HyperText Markup Language*). The standard allows an author to gives general guidelines for display and to specify the content of a page.
- HTML is **a mark-up language** because it does not include detailed formatting instructions and allows a browser to choose details. Consequently, two browsers may display an HTML document differently.



- For example, although HTML contains extension that allow an author to specify the size of text, the font to be used, or the width of a line, most authors choose instead to specify only a level of importance **as a number from 1 through 6**.
- The browser chooses a font and display size appropriate for each level.
- Similarly, HTML does not specifies exactly how a browser marks an item as **selectable**- . Some browsers **underline** selectable items, others display selectable items in a **different color** and some do both.
- Each browser must contain an HTML interpreter to display documents. Input to an HTML interpreter consists of a document that conforms to the HTML syntax; output consists of a formatted version of the document on the user display

# HTTP (Hypertext Transfer Protocol)

- The World Wide Web is built on top of the internet and uses the Internet protocol. HTTP is the client server application level protocol specifically designed to support hypermedia information systems.
- Web browsers use HTTP to connect with Web servers and access information on those servers.
- The protocol sets up a connection between web browser and then manages the exchange of information. HTTP defines the structure of the exchanged messages and how they are exchanged.
- 
- TCP is used as transport protocol

- TCP guarantees a reliable transfer of HTTP messages sent to the server.
- The basic protocol is **stateless**, meaning that the client and the server **do not store information about one another during the session**.
- The client connects with a server, the server transfers the requested information and the **connection is closed**. The server doesn't need to know anything about the client. It simply gives the requested information.
- If the client requires twice the same informations in a few seconds the server send it again.
- In HTTP version 1.0, every object on a web page (graphic, images, for example) required that a **new connection be established to transfer it**.
- HTTP 1.1 adds *persistence*, wich makes client/server connections more efficient. Persistence allows the client and the server **to mantain a connection** (not to be confused with state) until all objects on the page are tranferred.

## **non permanent connection / HTTP(1.0)**

The requested page is constituted of a HTML page and 10 JPEG images, all stored in the same server.

*www. someSchool.edu/somedepartment/home.index*

1)The HTTP client establish a TCP connection with the *www. someSchool.edu. Server*. The **port. 80** is used on wich the HTTP server will receive the requests of the HTTP client.

2) The HTTP client sends a HTTP request message to the server **by using the socket associated** to the connection established in 1). The request message requires the name of the requested file (*/somedepartment/home.index*).

3) The HTTP server receives the message **through the socket associated** to the connection established in 1), finds the file */somedepartment/home.index*, inserts it in a HTTP message and sends it to the client through the socket.

4) The HTTP server requires to TCP the conclusion of the connection.

5)The HTTP client receives the response message. The TCP connection is then closed. The message defines that the enclosed message is a HTML file.

The client extracts the file from the message, and finds the links to the 10 JPEG objects.

6)For each of the JPEG objects the steps 1,2,3,4 are repeated.

**Each TCP connection transports exactly a request message and an answer message. In the previous example 11 TCP connections are established.**

It is possible to establish more connections simultaneously (5-10, default) and each of them is relative to a request-answer transaction.

## Permanent Connection (HTTP/1.1)

- The server maintains TCP connection after the answer has been sent. The following requests and responses can be sent **on the same connection**.
- In the previous example the web page (HTML file and 10 images ) can be sent on the same **TCP permanent connection**.
- The HTTP server closes the connection if it is not used after a defined time interval (time out ).

## HTTP messages format

HTTP supports four basic operations that a browser can specify when making a request:

- **GET** requests a specified item from the server. The server returns a heading that contains status information followed by a blank line followed by the item
- **HEAD** requests status information about an item. The server returns the status without returning a copy of the item itself.
- **POST** sends data to the server. The server appends the data to a specified item (i.e, a item is appended to a list of messages.
- **PUT** sends data to the server. The server uses the data to replace a specified item

*Example*

```
GET/somedir/page.html HTTP/ 1.1      (request line).  
Host:www.someSchool.edu  
Connection: close                    (header line)  
User-Agent : Mozilla/4.0  
Accept-language :It
```

The first line is the *request line*

The second line is the name of the server

The third line indicates that the connection is not permanent.

The fourth line indicates the name of the browser (Mozilla is a Netscape browser).

The fifth line indicates that the user wishes an Italian version of the document (if it is not indicated, the version sent is the default version)



## *Response message*

HTTP/1.1 200 OK  
Connection : close  
Date: Thu, 06 January 2004 12:00:15  
Server: Apache /1.3.0 (Unix)  
Last- Modified: Mon, 22 Jun 2003 9:23:24  
Content-Length: 6821  
Content-Type :text/html

(data, data, data,...)

*The first line* contains a **status code** that tells the browser whether the server handled the request (404 indicates that the requested item cannot be found; 200 indicates that all is OK).

*Connection: close* indicate to the client that the connection will be closed at the end of the message sending.

*Last- Modified* indicates the date of the last modification of the object

*Content-Length* is the byte dimension of the object

*Content-Type* indicates that the object is HTML text

## Web caching

- **Browsers use a cache to improve document access.** The browser places a copy of each item it retrieves in a cache in the local disk. When a user selects an item the browser checks the disk cache before retrieving a fresh copy.
- Keeping items in a cache can improve performance dramatically. A browser can read an item from disk without waiting for network delays. Cache is especially important for large pages or for users who have slow network connections.
- Despite the large improvement in speed, retaining items in a cache for extended period of time is not always desirable.
  - A cache can take vast amount of disk space (large images)
  - Improvements in performance are only helpful if a user decides to view an item again.
- To help users control how a browser handles the cache, most browsers allow the user to adjust the cache policy.

**The user can set a time limit for caching, and the browser removes items from the cache after the time limit expires.**

**If an item is required after a time interval the copy in the cache may to be different in comparison with that present in the server.**

The browser can control the validity of the document present in the cache by sending to the server a message like this:

If-Modified-since: <date in the cache>.

The web server sends the message:

HTTP/1.0 304 Not Modified

Date:...

Server:...

(body of the object empty)

## Web caching (proxy server)

The user requests are sent to a server (**proxy server**) . In a disk of the server are stored copies of objects recently required .

Ex: <http://www.someschool.edu/campus.gif>

- The browser **creates a TCP connection with the proxy server** and sends a HTTP request for the object.
- The proxy server controls if a locally stored copy is present. If there is, it is sent into a HTTP response message to the user browser.
- Otherwise a **TCP connection with the server is created** and a HTTP request is sent to the proxy. The server sends the object to **the proxy** by using a HTTP response message
- The proxy stores a copy in the local memory and sends a copy to the browser.

## **Cooperative caching**

More proxies located in different Internet sites can cooperate in order to improve the performance.

Ex.: local cache which is connectet to a national cache

**ICP (Internet Caching Protocol)**. Application protocol used for the cooperation of different caches.