

Chương 7

Application Layer

Tài liệu : Forouzan , Data Communication and Networking

Các ứng dụng mạng

Network Applications

- Là những chương trình chạy trên những đầu cuối khác nhau, giao tiếp với nhau thông qua network
- Một số ứng dụng mạng :
 - text email, remote access to computers, file transfers, and newsgroups (1970s and 1980s)
 - Web surfing, search, and electronic commerce (mid-1990s)
 - Instant messaging and P2P file sharing
 - voice-over-IP (VoIP) and video conferencing over IP (Skype) ; user-generated video distribution (YouTube); and movies on demand such as Netflix (Since 2000,)
 - các ứng dụng mạng xã hội như Facebook và Twitter

Kiến trúc của ứng dụng mạng (1)

- Hai loại kiến trúc cho ứng dụng mạng :
 - Client – server
 - Peer – to – peer (P2P)
- **client-server architecture**
 - Một chương trình – gọi là server : cung cấp các dịch vụ , đáp ứng yêu cầu từ những chương trình khác _ gọi là client
 - Server Apps chạy trên host có một địa chỉ IP cố định, luôn ở trạng thái ON sẵn sàng phục vụ
 - Các client gửi yêu cầu tới server , và nhận kết quả từ server. Các client không giao tiếp với nhau
 - Ví dụ : Web application

Kiến trúc của ứng dụng mạng (2)

- **P2P architecture**

- Các bên tham gia là người dùng cuối (cá nhân, văn phòng, trường học ,...) chia sẻ dữ liệu với nhau mà không cần qua 1 server.
- P2P có khả năng tự mở rộng ; không yêu cầu hạ tầng máy chủ
- Vd: dịch vụ chia sẻ file BitTorrent, Internet Telephone Skype, IPTV (PPstream)

- **Kiến trúc lai giữa client-server và P2P**

Giao tiếp giữa các ứng dụng mạng (1)

- Các tiến trình giao tiếp với nhau bằng cách trao đổi các messages qua môi trường network
 - Vai trò giữa 2 tiến trình : client process (khởi tạo kết nối), server process (chờ và phục vụ)
 - Một client browser process trao đổi messages với một Web server process
 - Trong hệ thống P2P file-sharing, tiến trình cần download file (client) sẽ trao đổi với tiến trình upload file (server)

Giao tiếp giữa các ứng dụng mạng (2)

- Quá trình truyền Messages của sending process tới receiving process sẽ đi từ tầng Applications qua các tầng bên dưới theo mô hình TCP/IP , và ngược lại ở đầu nhận.
 - Process sử dụng dịch vụ được cung cấp bởi tầng transport : giao thức chuyển vận TCP hay UDP ; Kích thước segment có thể chuyển trong một giới hạn ; ...
 - Process sẽ yêu cầu dịch vụ từ tầng transport bằng sử dụng socket – một API

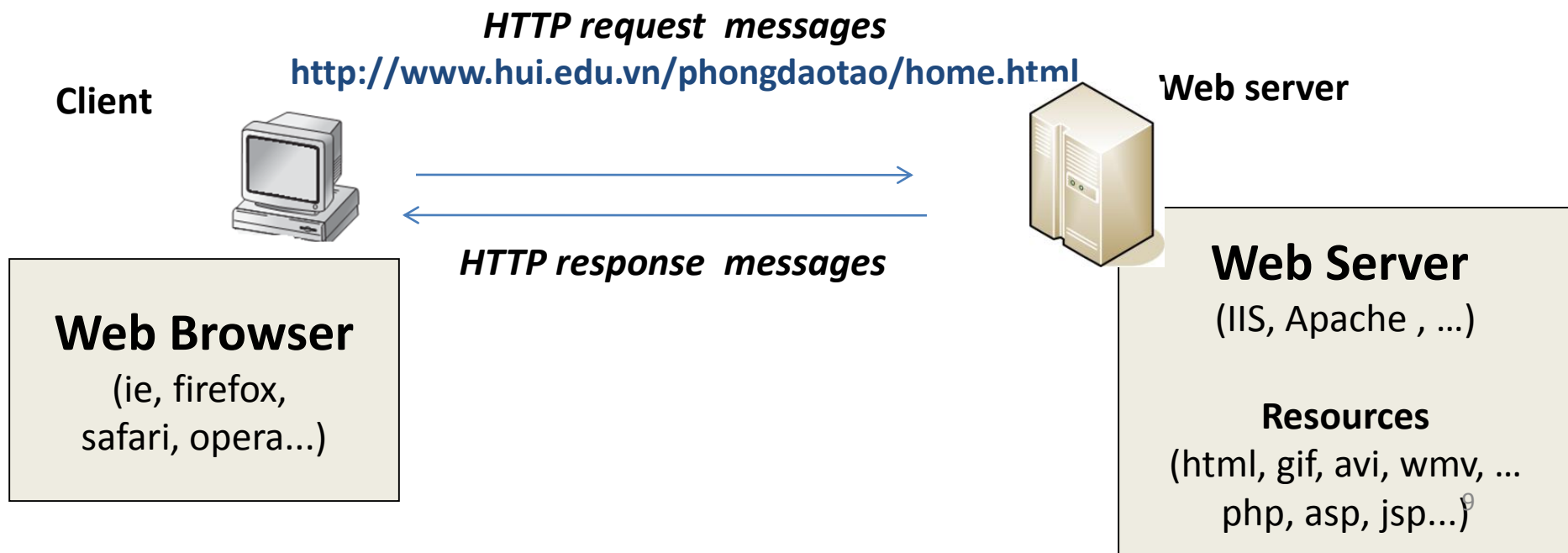
Các giao thức tầng Application

- Giao tiếp giữa các ứng dụng được xây dựng dựa trên các giao thức tầng Application
- Giao thức tầng Application chỉ định :
 - Các loại messages trao đổi giữa client và server
 - Cú pháp của mỗi loại messages : các field , kích thước và ý nghĩa thông tin trong field
 - Các quy tắc hướng dẫn cách gửi một message và trả lời một message

- Liệt kê tên một số giao thức tầng Application ?

HTTP (1)

- Mô hình hoạt động của ứng dụng Web
 - Client-server
 - Web server sử dụng port 80 , cung cấp Web pages (định vị bằng 1 URL)



Web and HTTP

First, a review...

- ❖ *web page* consists of *objects*
- ❖ object can be HTML file, JPEG image, Java applet, audio file,...
- ❖ web page consists of *base HTML-file* which includes *several referenced objects*
- ❖ each object is addressable by a *URL*, e.g.,

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

protocol

host name

path name

HTTP (2)

- The Hyper Text Transfer Protocol (HTTP)
- Định nghĩa cách thức giao tiếp giữa client HTTP program và server HTTP program
 - Phiên bản HTTP 1.1 : Định nghĩa trong RFC 2616 (năm 1999) ; Đã được sửa đổi và thay thế vào 2006 : RFC 723X ; Dự kiến phiên bản mới HTTP 2.0
 - Sử dụng dịch vụ TCP
 - Là giao thức không trạng thái (stateless protocol)
 - Sử dụng persistent connections hoặc non-persistent connections

HTTP (3)

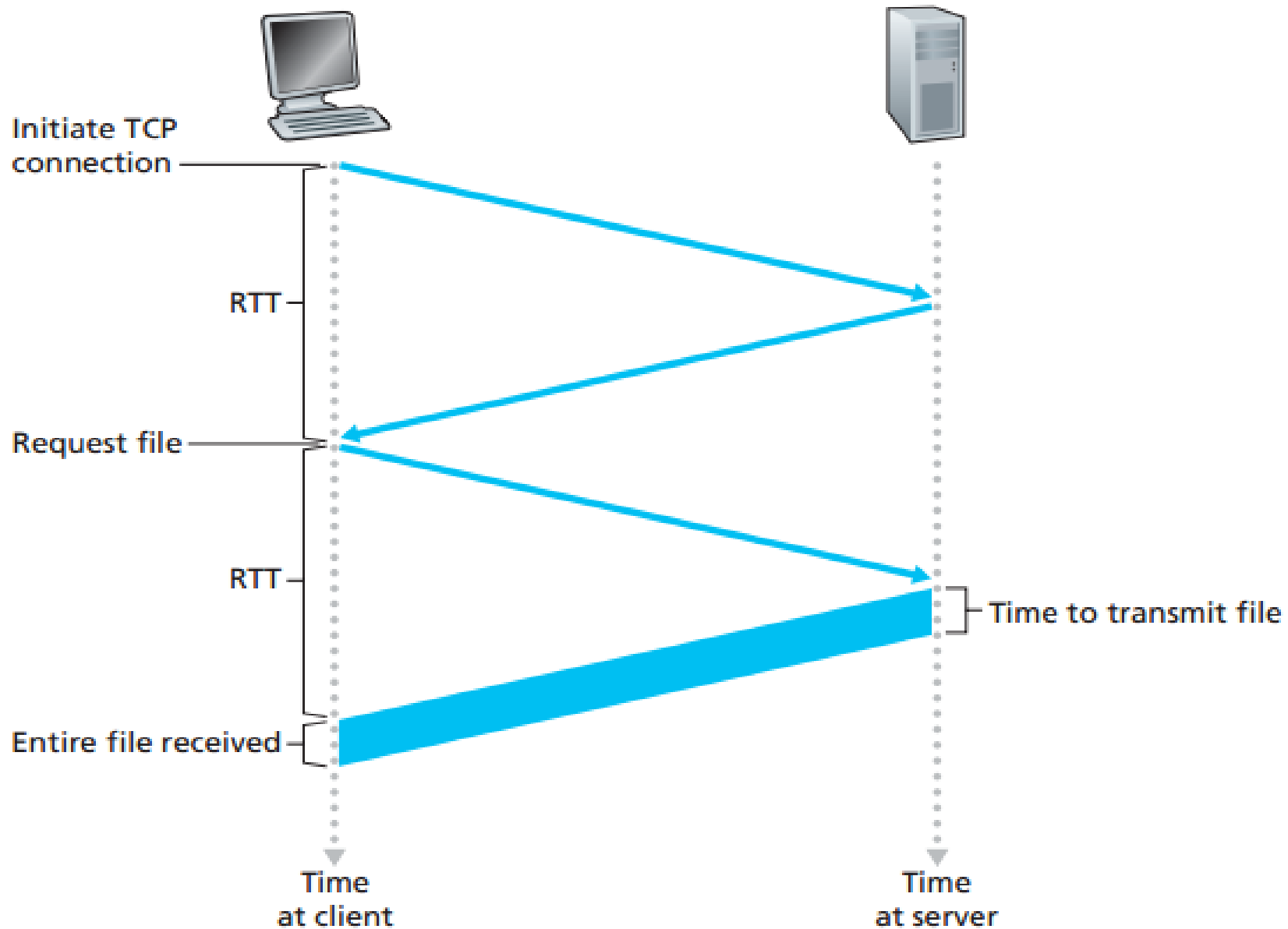
- Các bước trao đổi message giữa server và client (Non-Persistent Connections) :
 1. HTTP client process khởi tạo một kết nối TCP tới Web server (port 80)
 2. Client gửi một HTTP request message tới server . Request chứa thông tin về địa chỉ của trang web html (dạng URL) – giả định trang web chứa text và 2 JPEG object
 3. Server nhận được request message, tìm trang Web theo yêu cầu , đóng gói nó trong một HTTP response message và gửi tới client

HTTP (4)

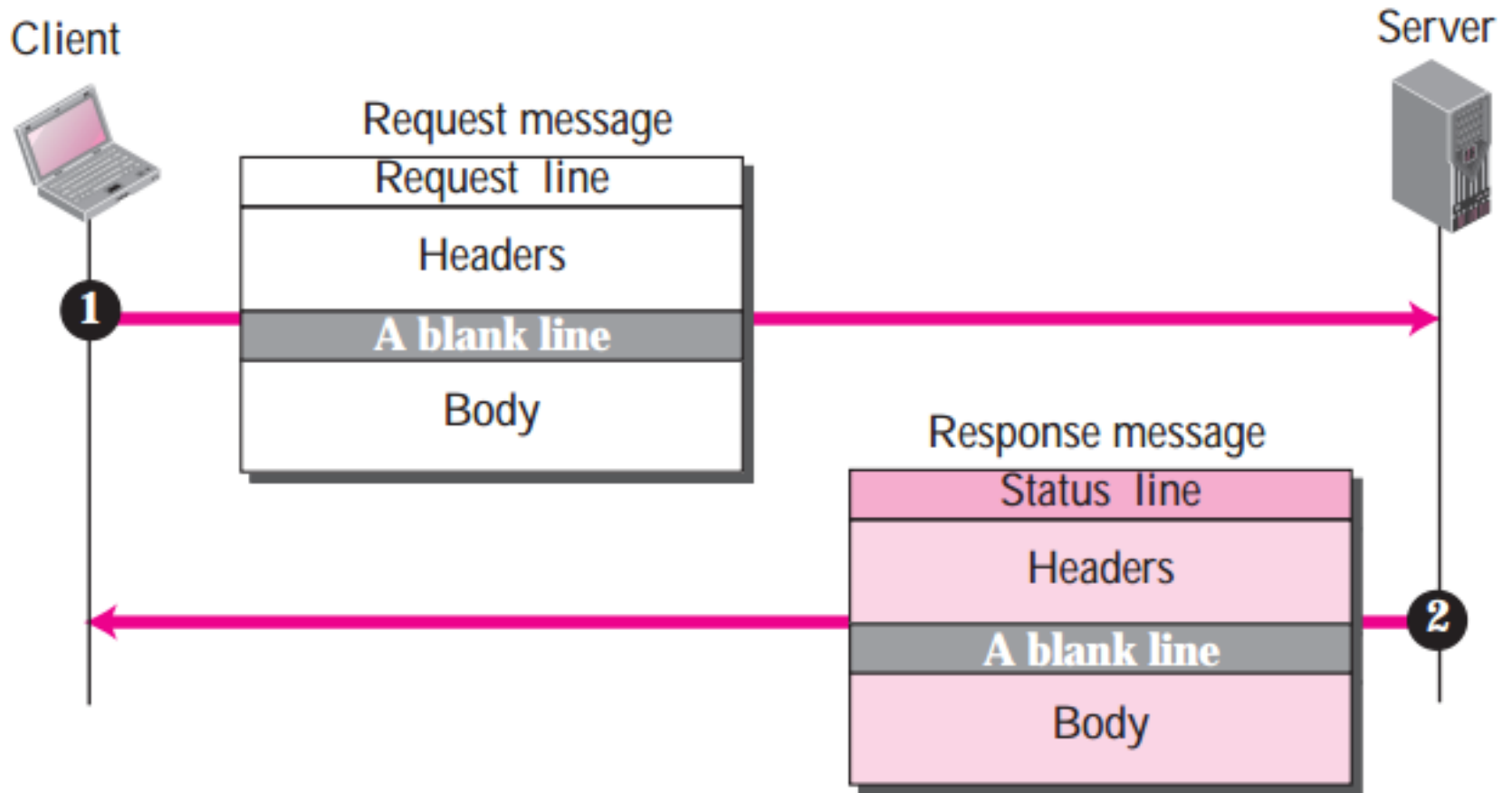
- Các bước trao đổi message giữa server và client (Non-Persistent Connections) :
 4. Server yêu cầu TCP đóng kết nối
 5. Client nhận response message. Kết nối TCP kết thúc. Client đọc file HTML nhận được, tìm tham chiếu tới 2 JPEG object.
 6. Lặp lại 4 bước đầu với mỗi một JPEG object
- => Mỗi kết nối mất 2 RTT

HTTP (5)

- Các bước trao đổi message giữa server và client (Persistent Connections) :
 - Server vẫn duy trì kết nối sau khi gửi response message. Các request và response messages sau đó được thực hiện trong cùng một kết nối
 - Server sẽ đóng một kết nối khi không sử dụng trong một khoảng thời gian qui định.
 - HTTP mặc định sử dụng persistent connections với pipelining.



HTTP transaction



HTTP (6)

- HTTP Message Format :
 - HTTP Request Message

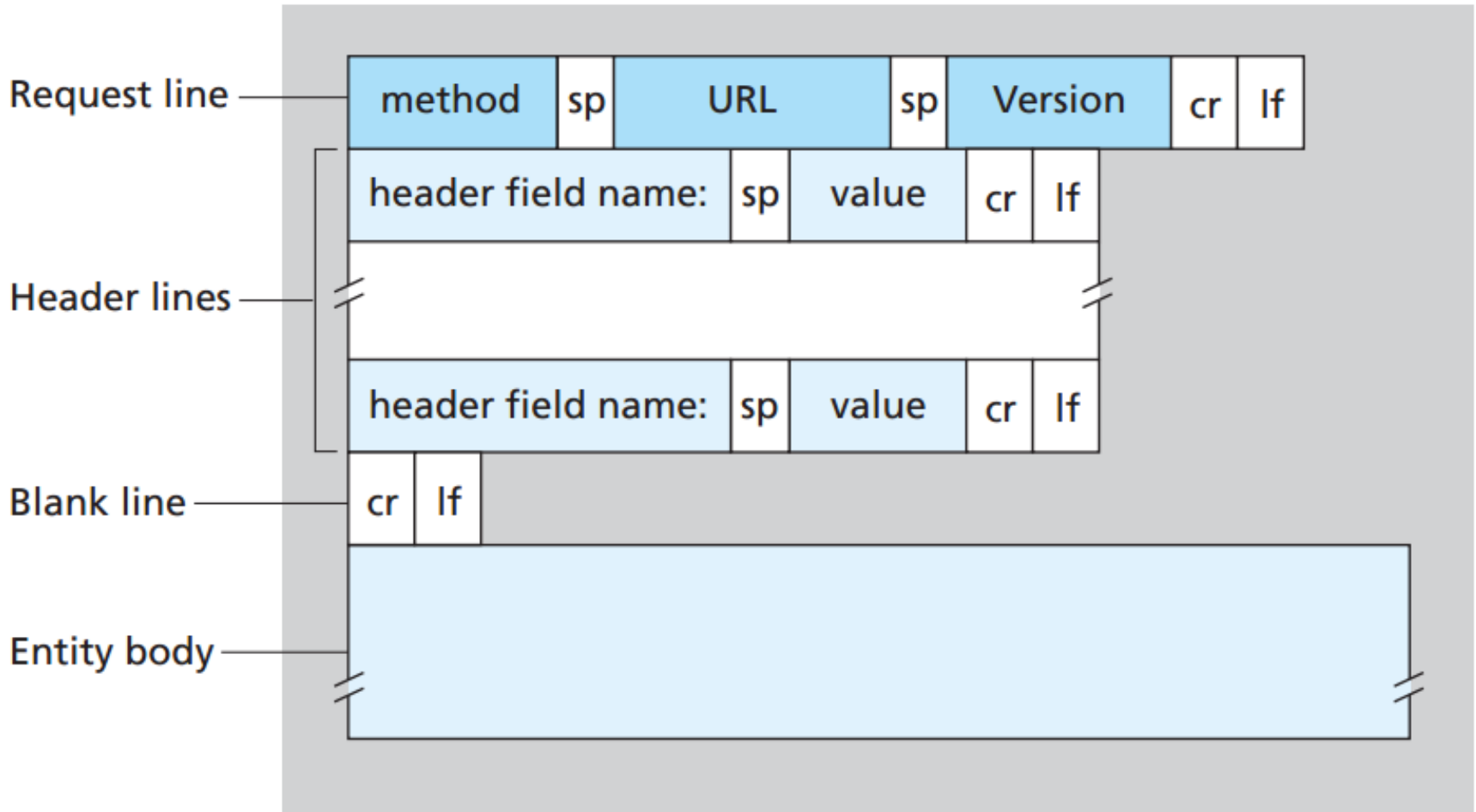
GET /somedir/page.html HTTP/1.1

Host: www.someschool.edu

Connection: close

User-agent: Mozilla/5.0

Accept-language: fr



General format of an HTTP request message

Table 22.1 *Methods*

<i>Method</i>	<i>Action</i>
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
DELETE	Remove the Web page
OPTIONS	Enquires about available options

Table 22.2 *Request Header Names*

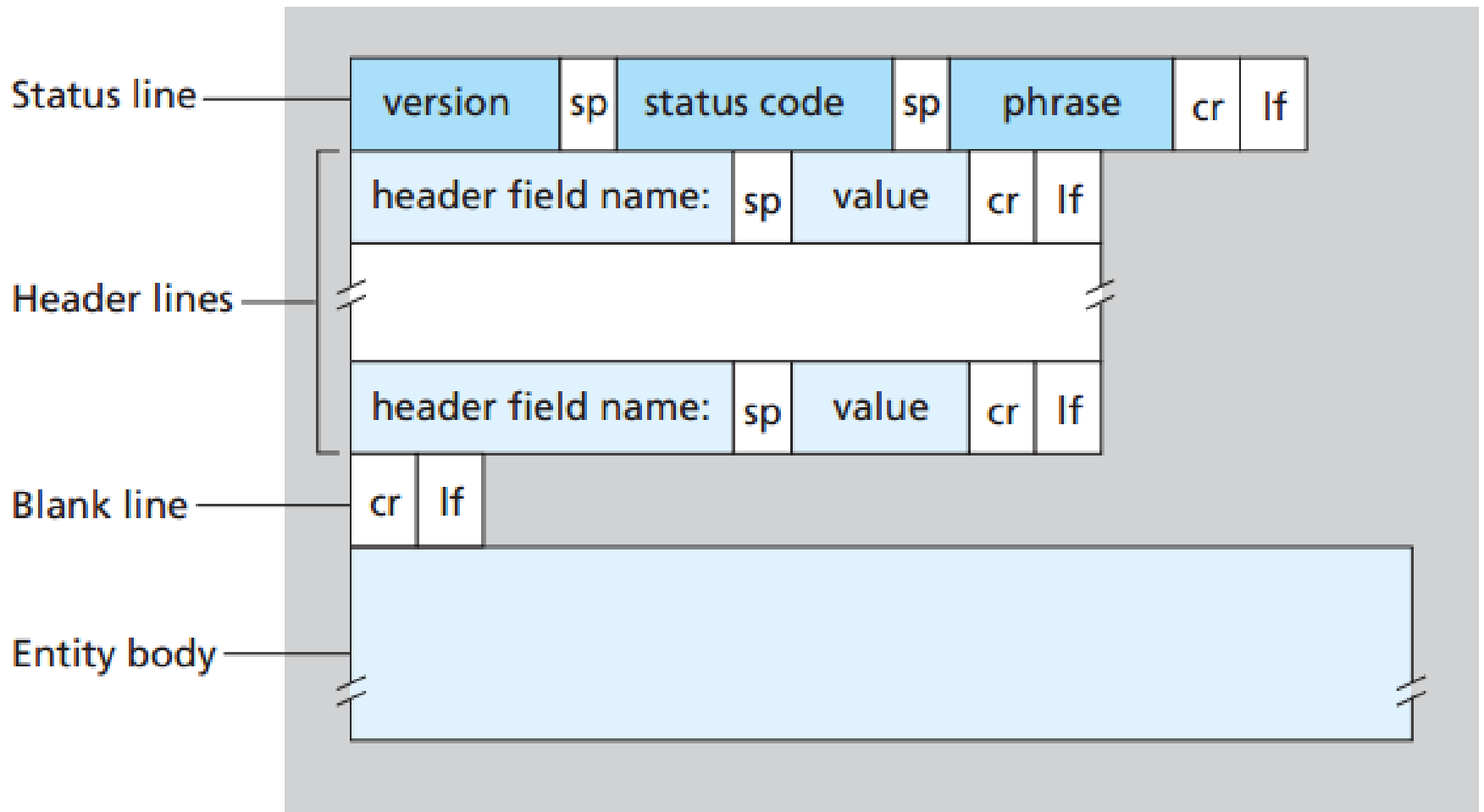
<i>Header</i>	<i>Description</i>
User-agent	Identifies the client program
Accept	Shows the media format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
Host	Shows the host and port number of the client
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Cookie	Returns the cookie to the server
If-Modified-Since	Returns the cookie to the server

Body In Request Message The body can be present in a request message. Usually, it contains the comment to be sent.

- HTTP Message Format :
 - HTTP Response Message

Demo ???

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
( data ..... )
```



General format of an HTTP response message

Table 22.3 *Status Codes and Status Phrases*

<i>Status Code</i>	<i>Status Phrase</i>	<i>Description</i>
Informational		
100	Continue	The initial part of the request received, continue.
101	Switching	The server is complying to switch protocols.
Success		
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon.
204	No content	There is no content in the body.
Redirection		
301	Moved permanently	The requested URL is no longer used by the server.
302	Moved temporarily	The requested URL has moved temporarily.
304	Not modified	The document has not modified.
Client Error		
400	Bad request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in this URL.
406	Not acceptable	The format requested is not acceptable.
Server Error		
500	Internal server error	There is an error, such as a crash, at the server site.
501	Not implemented	The action requested cannot be performed.
503	Service unavailable	The service is temporarily unavailable.

Table 22.4 *Response Header Names*

<i>Header</i>	<i>Description</i>
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme
Content-Language	Specifies the language
Content-Length	Shows the length of the document
Content-Type	Specifies the media type
Location	To ask the client to send the request to another site
Accept-Ranges	The server will accept the requested byte-ranges
Last-modified	Gives the date and time of the last change

Body The body contains the document to be sent from the server to the client. The body is present unless the response is an error message.