

شبكات الحاسبات

قسم الهندسة الكهربائية
المرحلة الثالثة
م.م. براء سعيد العبيدي

Application Layer

- Network applications are the reason for existence of a computer network—if we couldn't conceive of any useful applications, there wouldn't be any need for networking infrastructure and protocols to support them.
- Internet applications include:
 - World Wide Web, encompassing Web surfing, search, and electronic commerce, instant messaging and P2P file sharing,
 - Voice over IP and video conferencing such as Skype, Facetime, and Google Hangouts.
 - User generated video such as YouTube and movies on demand such as Netflix.
 - Multiplayer online games such as Second Life and World of Warcraft, PubG.
 - Social networking applications—such as Facebook, Instagram, Twitter, and WeChat.
 - Mobile apps

Network Application Architectures

I. Client-server architecture:

- there is an always-on host, called the server, which services requests from many other hosts, called clients.
- a single-server host is incapable of keeping up with all the requests from clients. For example, a popular social-networking site can quickly become overwhelmed if it has only one server handling all of its requests. For this reason, a data center, housing a large number of hosts, is often used to create a powerful virtual server.

II. Peer-to-Peer (P2P) architecture:

- there is minimal (or no) reliance on dedicated servers in data centers. Instead, the application exploits direct communication between pairs of intermittently connected hosts, called peers.

The Interface Between the Process and the Computer Network

- A network application consists of pairs of processes that send messages to each other over a network.
- Any message sent from one process to another must go through the underlying network.
- A process sends messages into, and receives messages from, the network through a software interface called a **socket**.



Application Layer Protocols

Application	Application-Layer Protocols
Electronic mail	SMTP (Simple Mail Transfer Protocol)
Web	HTTP (Hyper Text Transfer Protocol)
File transfer	FTP (File Transfer Protocol)
Streaming multimedia	HTTP
Internet telephony	Skype

HTTP: Hyper Text Transfer Protocol

The Web's application-layer protocol, is at the heart of the Web.

HTTP is implemented in two programs: a client program and a server program.

The client program and server program, executing on different end systems, talk to each other by exchanging HTTP messages.

HTTP is a **stateless protocol**.

Web (http) uses the client-server application architecture

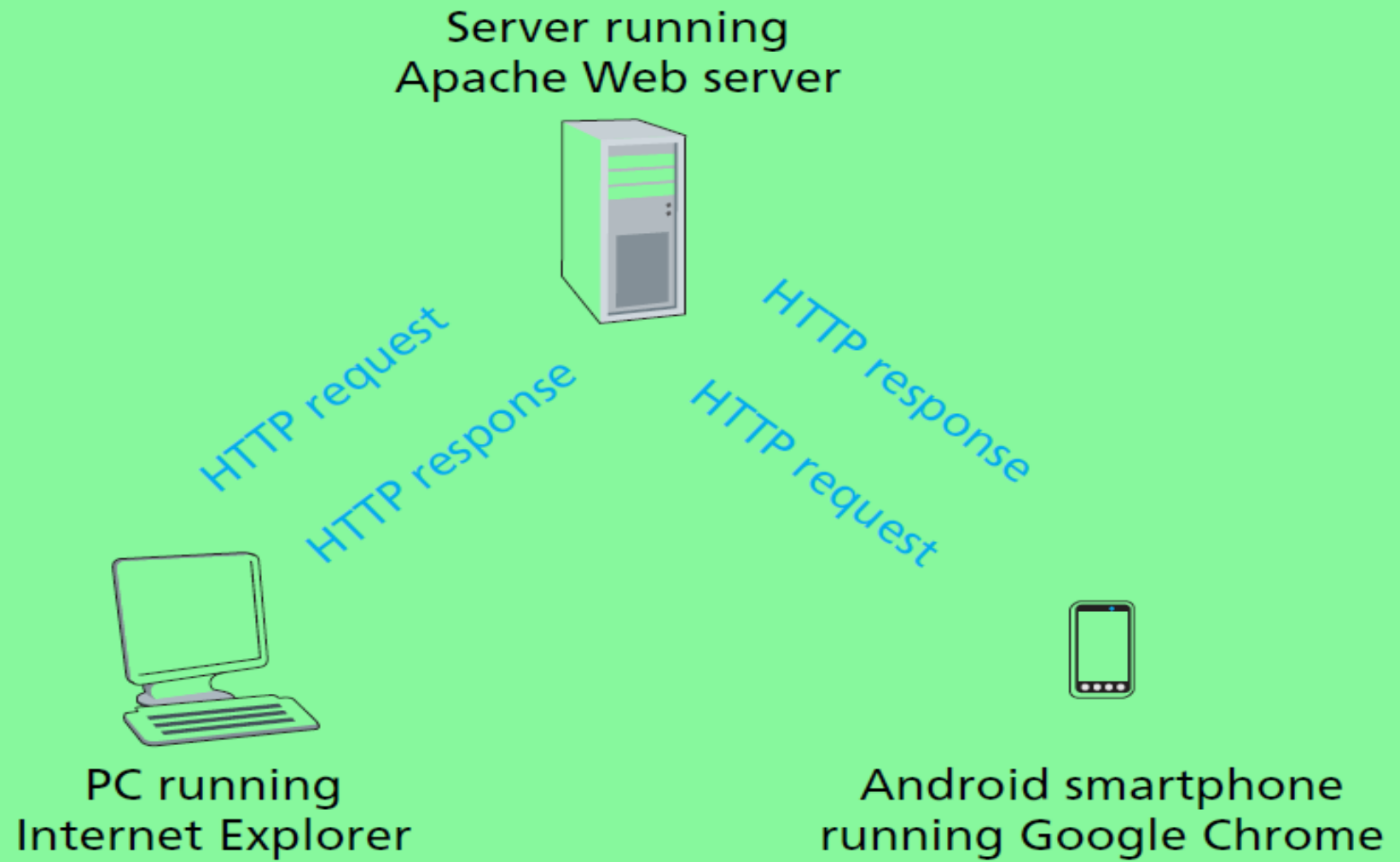


Figure 2.6 ♦ HTTP request-response behavior

Non-Persistent and Persistent Connections

A Web page consists of objects. An object is simply a file—such as an HTML file, a JPEG image, a Java applet, or a video clip—that is addressable by a single URL.

If a Web page contains HTML text and five JPEG images, then the Web page has six objects: the base HTML file plus the five images.

Non-persistent connections: each request/response pair be sent over a *separate* connection.

persistent connections: all the requests and their corresponding responses be sent over the same connection.

An Example of non-persistent connections

Suppose the page consists of a one base HTML file and 10 JPEG images, and that all 11 of these objects reside on the same server (total 11 objects). The URL (Uniform Resource Locator) is:

```
https://www.uoanbar.edu.iq/EngineeringCollege.index
```

1. The HTTP client process initiates a connection to the server `www.somuoanbar.edu` on port number 80, which is the default port number for HTTP. Associated with the connection, there will be a socket at the client and a socket at the server.
2. The HTTP client sends an HTTP request message to the server via its socket that contains the path `/EngineeringCollege`.
3. The HTTP server process receives the request message via its socket, retrieves the object from its storage (RAM or hard disk), encapsulates the object in an HTTP response message, and sends the response message to the client via its socket.
4. The HTTP server process closes the TCP connection.
5. The HTTP client receives the response message.
6. The first four steps are then repeated for each of the referenced JPEG objects.

Round-Trip Time (RTT)

- RTT is the time it takes for a small packet to travel from client to server and then back to the client.

- Example:

A user clicks on a URL (link) requesting an HTML file.

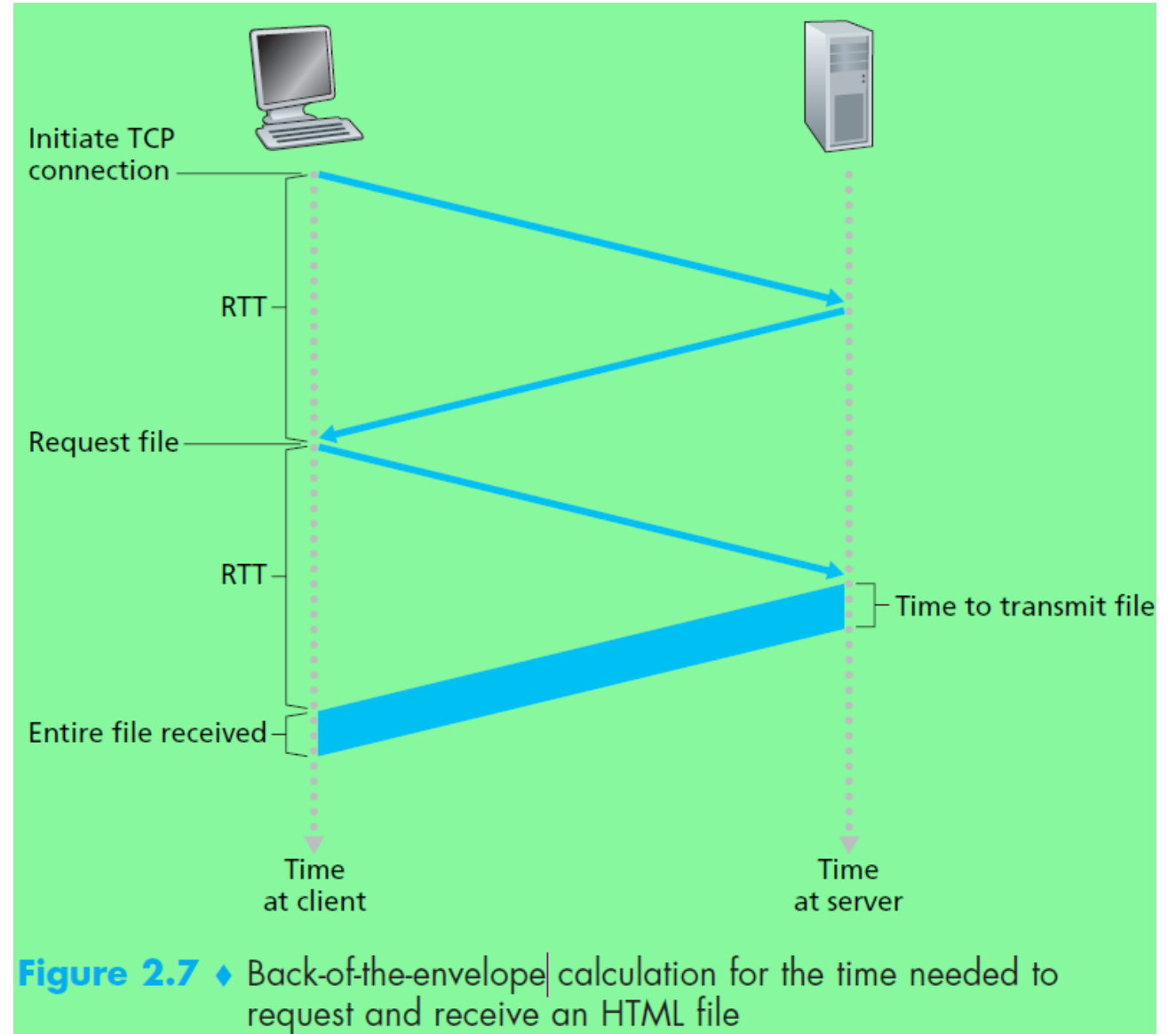


Figure 2.7 ♦ Back-of-the-envelope calculation for the time needed to request and receive an HTML file

HTTP Message Format

