

Chapter 2.2 Web and HTTP

24/05/2018 [Th]

2.2.1 Review of Web and HTTP

- *Web pages* are made up of *objects*.
- An **object** can be an *HTML file*, a *JPEG image*, a *Java applet*, an *audio file*, etc...
- A web page consists of a base HTML file, which references several objects, which are addressable by a *URL*.

2.2.2 Overview of HTTP

- **HTTP** stands for *Hypertext Transfer Protocol*.
- It uses a *client/server model*:
 - The **client** is the browser that requests, receives, and displays web objects.
 - The **server** is the web server, which sends objects in response to requests.
- TCP is used by the client to create a socket connection to the server on port 80. The server will accept the TCP connection and then HTTP messages will be exchanged between the client browser and the web server. When it is finished, the connection is closed.
- Note that *HTTP* is "*stateless*", meaning that the server does not keep information about past client requests.
- There are *two types* of HTTP connections:
 - In **non-persistent HTTP**, at most one object can be sent over the TCP connection before the connection is closed. Thus, downloading multiple objects would require multiple connections.
 - * It has a response time of $2RTT + \text{file transmission time}$ (1 RTT to initiate TCP connection, and one for HTTP request).
 - * Browsers often open multiple TCP connections in parallel to fetch referenced objects.
 - In **persistent HTTP**, multiple objects can be sent over a single TCP connection between client and server.
 - * Very low response time (as little as 1 RTT + transmission time) because the connection is left open.
- **Note:** **RTT** is the time for a small packet to travel from the client to the server and back.
- There are two types of HTTP messages: *request*, and *response*.
- See slide 2-28 to 2-31 for the format of request messages and response messages.
- HTTP/2 was released on November 2012. It uses a new method to decrease latency and increase page load speeds in web browsers.
 - In HTTP/2, HTTP messages are broken down into independent frames. They are interleaved and reassembled after being received.
 - Messages are interleaved in parallel, so none are blocked.
 - A single connection can also deliver multiple requests and responses in parallel, which means multiple connections are no longer needed for parallel delivery.

2.2.3 Cookies

- Many websites use cookies for *4 components*:
 1. Cookie header line of HTTP response message.
 2. Cookie header line in next HTTP request message.
 3. Cookie file kept on user's host, which is managed by the user's browser.
 4. Back-end databases of websites.
- As an **example**:
 - If Susan always accesses the internet from PC, and she visits a website for the first time, when the HTTP requests arrives at that website, the website will create a *unique ID* and an *entry in backend database for the ID*.
- Cookies can be used for:
 - authorization
 - shopping carts
 - recommendations
 - user session state

2.2.4 Web Caches

Also known as a "proxy server"

- The goal of a **web cache** is to fulfill the client request *without involving the origin server*.
- Browsers usually accesses the web via a cache. When the browser sends HTTP requests, they are sent to the cache.
 - If the object is in the cache, then the cache *returns the object*.
 - Otherwise, the cache *requests the object* from the origin server, then returns it to the client.
- A cache acts as both a *client* and a *server*.
- The cache is typically installed by the ISP.
- Web caching helps to:
 - Reduce response time for client requests.
 - Reduce traffic on an institution's access link.
- To calculate total delay, do:
$$(1 - \text{cache satisfaction frequency}) * (\text{delay from origin server}) + (\text{cache satisfaction frequency}) * (\text{delay when satisfied by cache})$$
- A **conditional GET** doesn't send objects if cache already has an up-to-date cached version.
 - There will be no object transmission delay.
 - The cache *specifies the date* of the cached copy in the HTTP request.
 - The server response contains *no object* if the cached copy is up-to-date.