CS3200: Computer Networks

Lecture 31

IIT Palakkad

30 Oct, 2019

E-mail

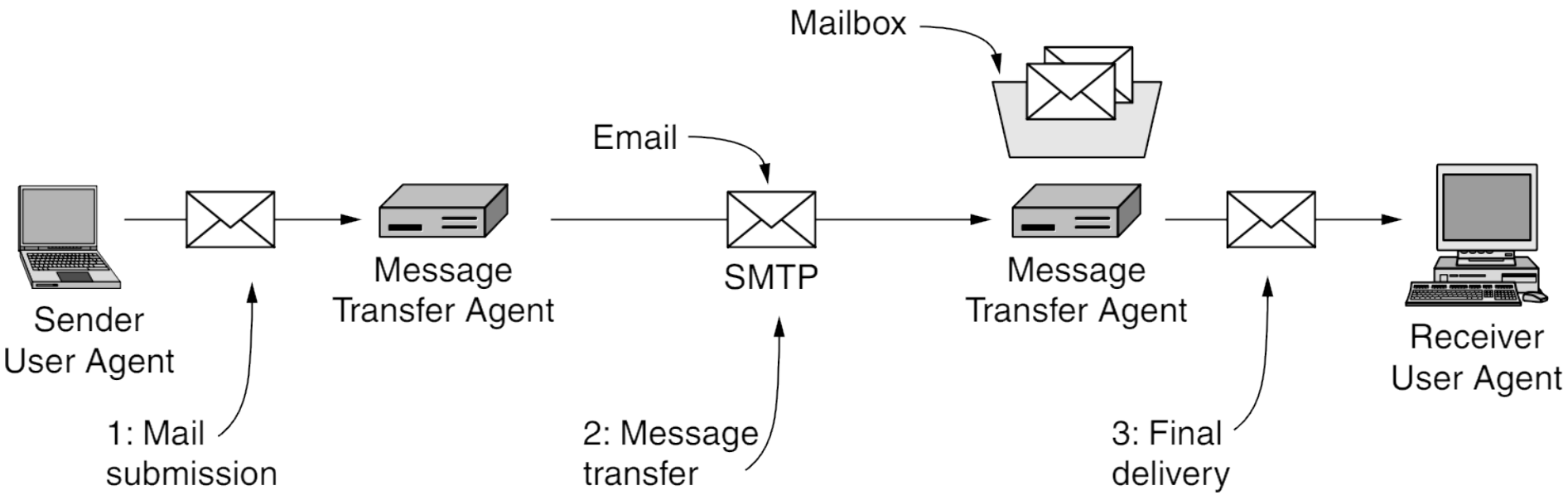
- Electronic mail, or more commonly email, has been around for over three decades. Faster and cheaper than paper mail, email has been a popular application since the early days of the Internet.

- The first email systems simply consisted of file transfer protocols, with the convention that the first line of each message (i.e., file) contained the recipient's address. As time went on, email diverged from file transfer and many features were added, such as the ability to send one message to a list of recipients.

- Multimedia capabilities became important in the 1990s to send messages with images and other non-text material.

Architecture

Consists of two kinds of subsystems: the **user agents**, which allow people to read and send email, and the **message transfer agents**, which move the messages from the source to the destination.



Read Section 7.2.2 of Tanenbaum.

Simple Mail Transfer Protocol (SMTP)

- The simplest way to move messages is to establish a transport connection from the source machine to the destination machine and then just transfer the message.
- Within the Internet, email is delivered by having the sending computer establish a TCP connection to port 25 of the receiving computer.
- Listening to this port is a mail server that speaks SMTP (Simple Mail Transfer Protocol). This server accepts incoming connections, subject to some security checks, and accepts messages for delivery.
- If a message cannot be delivered, an error report containing the first part of the undeliverable message is returned to the sender.

The World Wide Web (WWW)

- It is an architectural framework for accessing linked content spread out over millions of machines all over the Internet.
- From the users' point of view, the Web consists of a vast, worldwide collection of content in the form of **Web pages**, often just called **pages** for short.
- Pages are generally viewed with a program called a browser. Firefox, Internet Explorer, and Chrome are examples of popular browsers.
- A piece of text, icon, image, and so on associated with another page is called a **hyperlink**.

WWW — Client Side

Say you want to explore `http://www.cs.washington.edu/index.html`

- The browser determines the URL
- The browser asks DNS for the IP address of the server *www.cs.washington.edu*.
- DNS replies with 128.208.3.88.
- The browser makes a TCP connection to 128.208.3.88 on port 80, the well-known port for the HTTP protocol.
- It sends over an HTTP request asking for the page */index.html*.
- The server sends the page as an HTTP response, for example, by sending the file */index.html*, and the browser displays it.
- The TCP connections are released if there are no other requests to the same servers for a short period.

WWW — Server Side

- Accept a TCP connection from a client (a browser).
- Get the path to the page, which is the name of the file requested.
- Get the file (from disk).
- Send the contents of the file to the client.
- Release the TCP connection
- One problem with the simple design is that accessing files is often the bottleneck. Disk reads are very slow compared to program execution, and the same files may be read repeatedly from disk using operating system calls.
- Another problem is that only one request is processed at a time. The file may be large, and other requests will be blocked while it is transferred.

WWW — Server Side

- One obvious improvement (used by all Web servers) is to maintain a cache in memory of the n most recently read files or a certain number of gigabytes of content.
- Before going to disk to get a file, the server checks the cache. If the file is there, it can be served directly from memory, thus eliminating the disk access.
- To tackle the problem of serving a single request at a time, one strategy is to make the server **multithreaded**. In one design, the server consists of a front-end module that accepts all incoming requests and k processing modules.

Cookies

- How can we return different pages to different users depending on what they have already done with the server?
- This problem is solved with an often critized mechanism called **cookies**.
- The cookie is a rather small, named string (of at most 4 KB) that the server can associate with a browser.
- A cookie may contain up to five fields.

Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=297793521	15-10-10 17:00	Yes
jills-store.com	/	Cart=1-00501;1-07031;2-13721	11-1-11 14:22	No
aportal.com	/	Prefs=Stk:CSCO+ORCL;Spt:Jets	31-12-20 23:59	No
sneaky.com	/	UserID=4627239101	31-12-19 23:59	No

Web tracking — A Controversial use of Cookie

- Sneaky Ads, contacts major Web sites and places ads for its clients' products on their pages. It gives them a URL to add to each page e.g., `http://www.sneaky.com/382674902342.gif`
- When a user first visits a page, P, containing such an ad, the browser fetches the image file at www.sneaky.com, so it sends a request there for the image. A GIF file containing an ad is returned, along with a cookie containing a unique user ID, 4627239101.
- Sneaky records the fact that the user with this ID visited page P. This is easy to do since the path requested (382674902342.gif) is referenced only on page P.
- When the user visits another Web page containing any of Sneaky's ads, say, http://www.sneaky.com/19365491923.gif on the page and requests that file. Since it already has a cookie from the domain sneaky.com, the browser includes Sneaky's cookie containing the user's ID. Sneaky now knows a second page the user has visited.

The HyperText Transfer Protocol (HTTP)

- HTTP is a simple request-response protocol that runs over TCP.
- It specifies what messages clients may send to servers and what responses they get back in return.
- The request and response headers are given in ASCII, just like in SMTP.
- Each request consists of one or more lines of ASCII text, with the first word on the first line being the name of the method requested.

Method	Description
GET	Read a Web page
HEAD	Read a Web page's header
POST	Append to a Web page
PUT	Store a Web page
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Connect through a proxy
OPTIONS	Query options for a page

The HyperText Transfer Protocol (HTTP)

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

HTTP Caching