COMPUTER COMMUNICATION NETWORKS

Prajeesha

Department of Electronics and Communication Engineering

COMPUTER COMMUNICATION NETWORKS

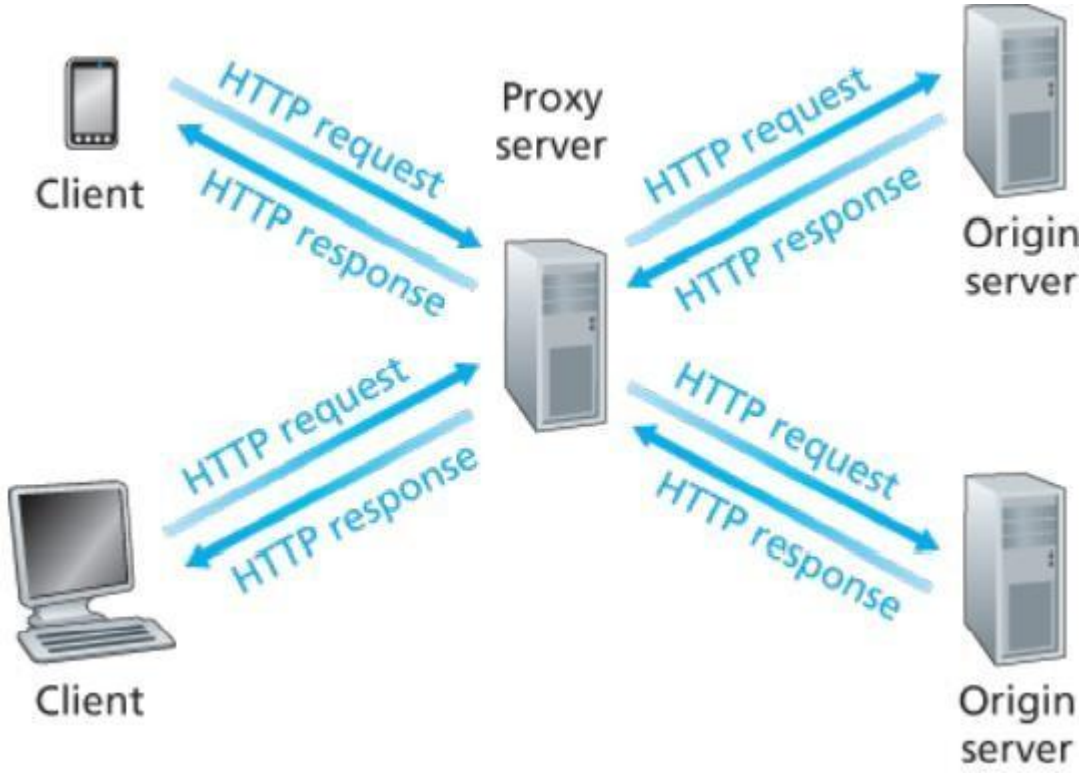
Web Cache & Cookies

Prajeesha
Department of Electronics and Communication Engineering

Web Caches (Proxy Server)

Goal: Satisfy client request without involving origin server

- User sets browser: Web accesses via cache
- Browser sends all HTTP requests to cache
  - Object in cache: cache returns object
  - Else cache requests object from origin server, then returns object to client



Web Caches (Proxy Server)

- Cache acts as both client and server
  - Server for original requesting client
  - Client to origin server
- Typically cache is installed by ISP (university, company, residential ISP)

Why Web caching?

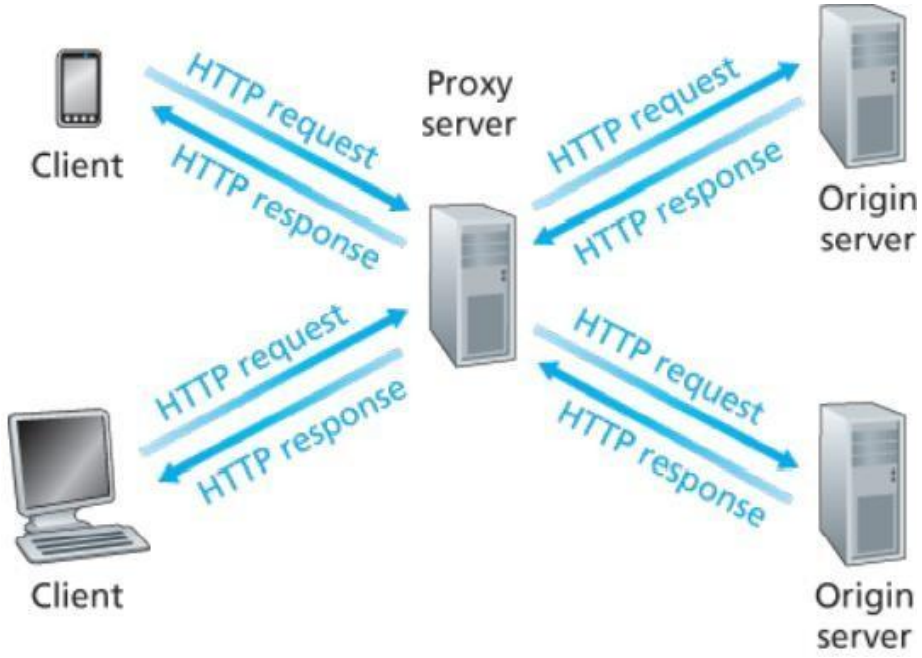
- Reduce response time for client request
- Reduce traffic on an institution's access link
- Internet dense with caches: enables "poor" content providers to effectively deliver content (so too does P2P file sharing)

Web Cache -Example

E.g. Suppose a browser is requesting the object http://www.someschool.edu/campus.gif .

1. The browser establishes a TCP connection to the Web cache and sends an HTTP request for the object to the Web cache.

2. The Web cache checks to see if it has a copy of the object stored locally. If it does, the Web cache returns the object within an HTTP response message to the client browser.

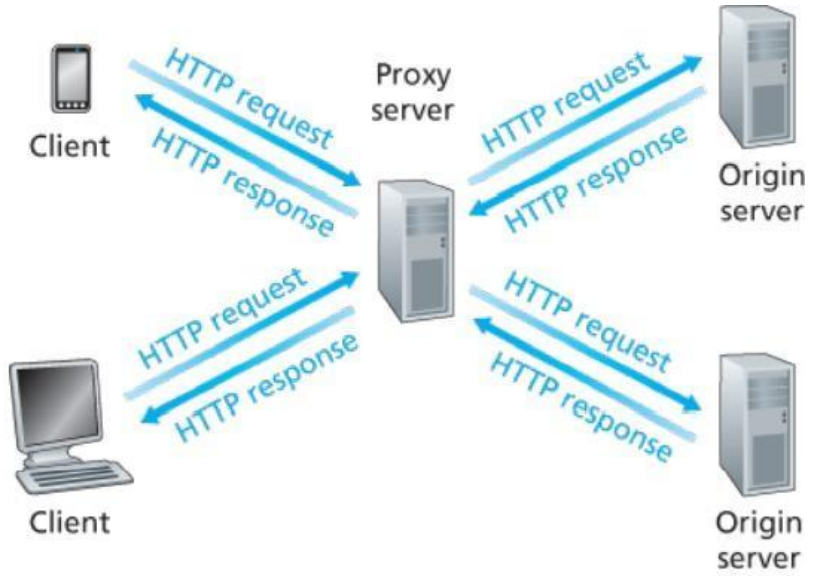


Web Cache -Example

E.g. Suppose a browser is requesting the object http://www.someschool.edu/campus.gif .

3. If the Web cache does not have the object, the Web cache opens a TCP connection to the origin server, that is, to www.someschool.edu . The Web cache then sends an HTTP request for the object into the cache-to-server TCP connection

4. When the Web cache receives the object, it stores a copy in its local storage and sends a copy, within an HTTP response message, to the client browser



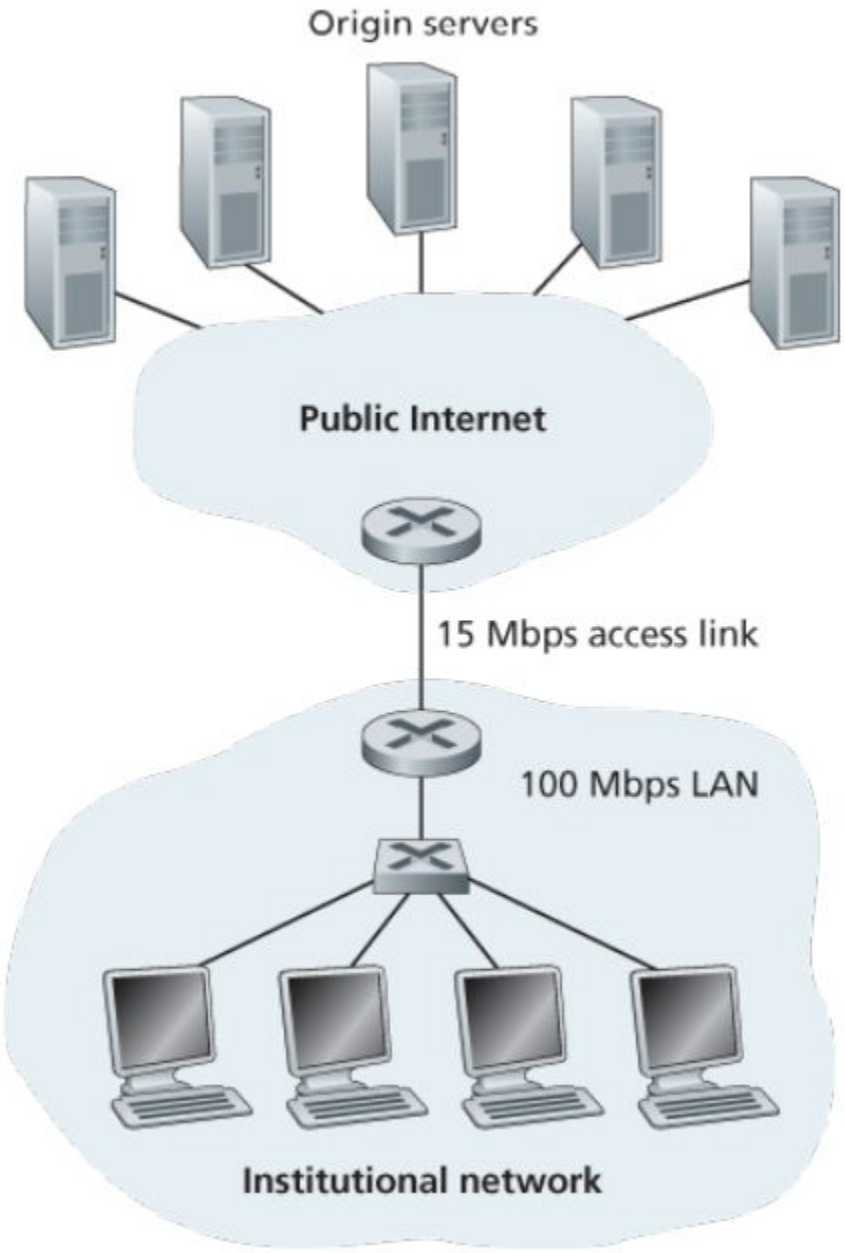
COMPUTER COMMUNICATION NETWORKS

Consider the figure which shows two networks : Institutional network and Public internet

Suppose the average object size is 1 Mbits and that the average request rate from the institution's browsers to the origin servers is 15 requests per second. Suppose that the HTTP request messages are negligibly small.

Also suppose that the amount of time it takes from when the router on the Internet side of the access link forwards an HTTP request until it receives the response is 2 secs on average.

The total response time = LAN delay + access delay + Internet delay

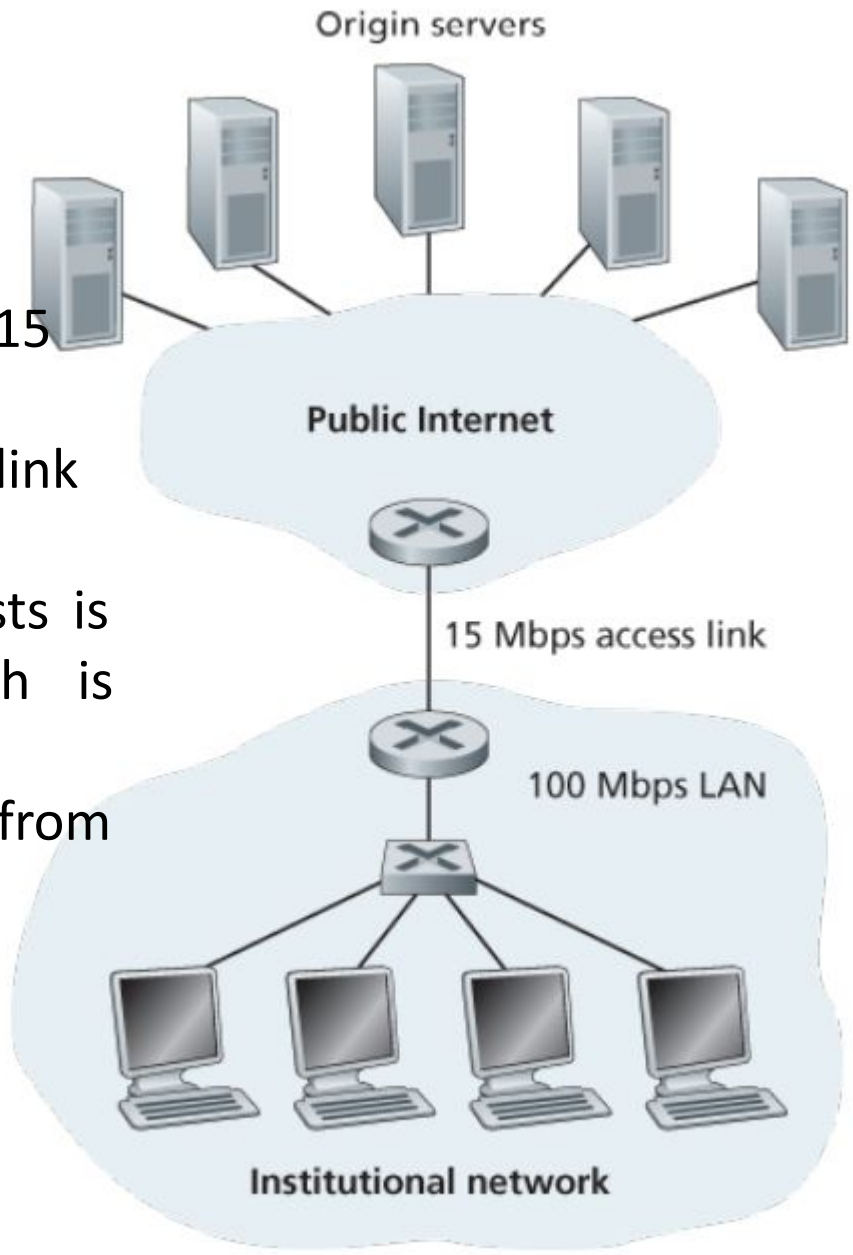


COMPUTER COMMUNICATION NETWORKS

The traffic intensity on the LAN =

(15 requests/sec) · (1 Mbits/request)/(100 Mbps)=0.15

As the traffic intensity approaches 1, the delay on a link becomes very large and grows without bound.
Thus, the average response time to satisfy requests is going to be on the order of minutes which is unacceptable for the institution's users.
One possible solution is to increase the access rate from 15 Mbps to, say, 100 Mbps, a costly proposition.



COMPUTER COMMUNICATION NETWORKS

The alternative solution installing a Web cache in the institutional network

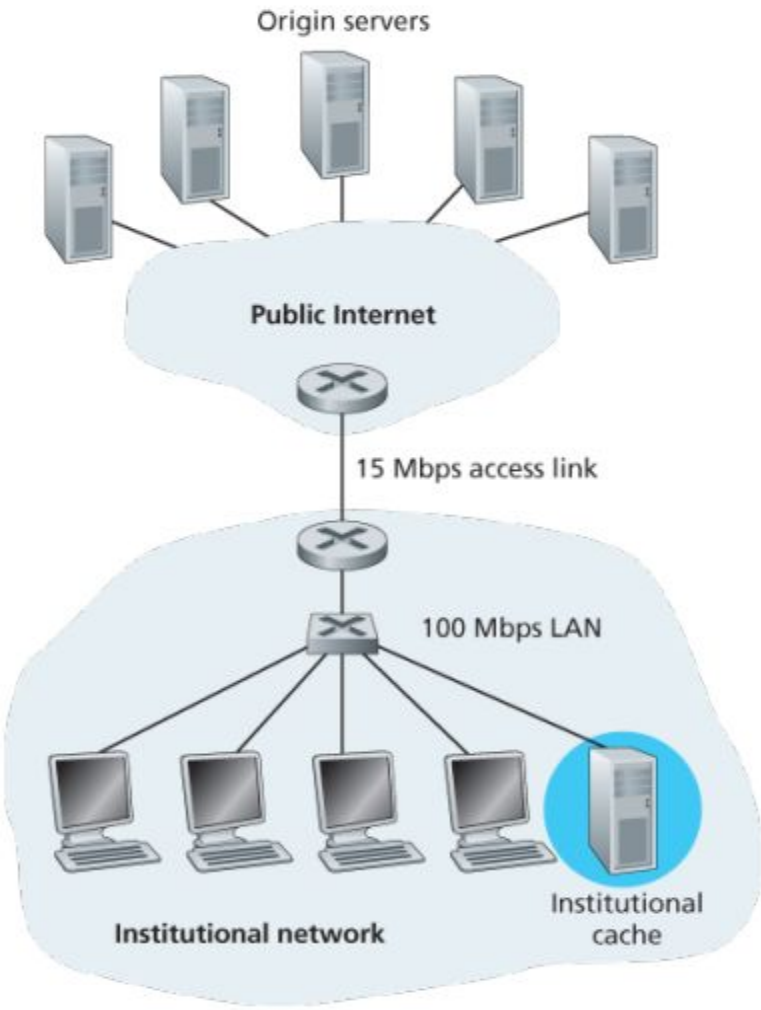
Hit rates (the fraction of requests that are satisfied by a cache) is typically ranging from 0.2 to 0.7.

Suppose that the cache provides a hit rate of 0.4, 40 percent of the requests will be satisfied almost immediately, say, within 10 milliseconds, by the cache.

The traffic intensity on the access link is reduced from 1.0 to 0.6. A traffic intensity less than 0.8 corresponds to a small delay, say, tens of milliseconds, on a 15 Mbps link. This delay is negligible compared with the two-second Internet delay.

Therefore, average delay = $0.4 \cdot (0.01 \text{ seconds}) + 0.6 \cdot (2.01 \text{ seconds}) = 1.2 \text{ secs}$

So the best solution is to purchase and install a Web cache



Adding a cache to the institutional network

User-server state: cookies

Many Web sites use cookies

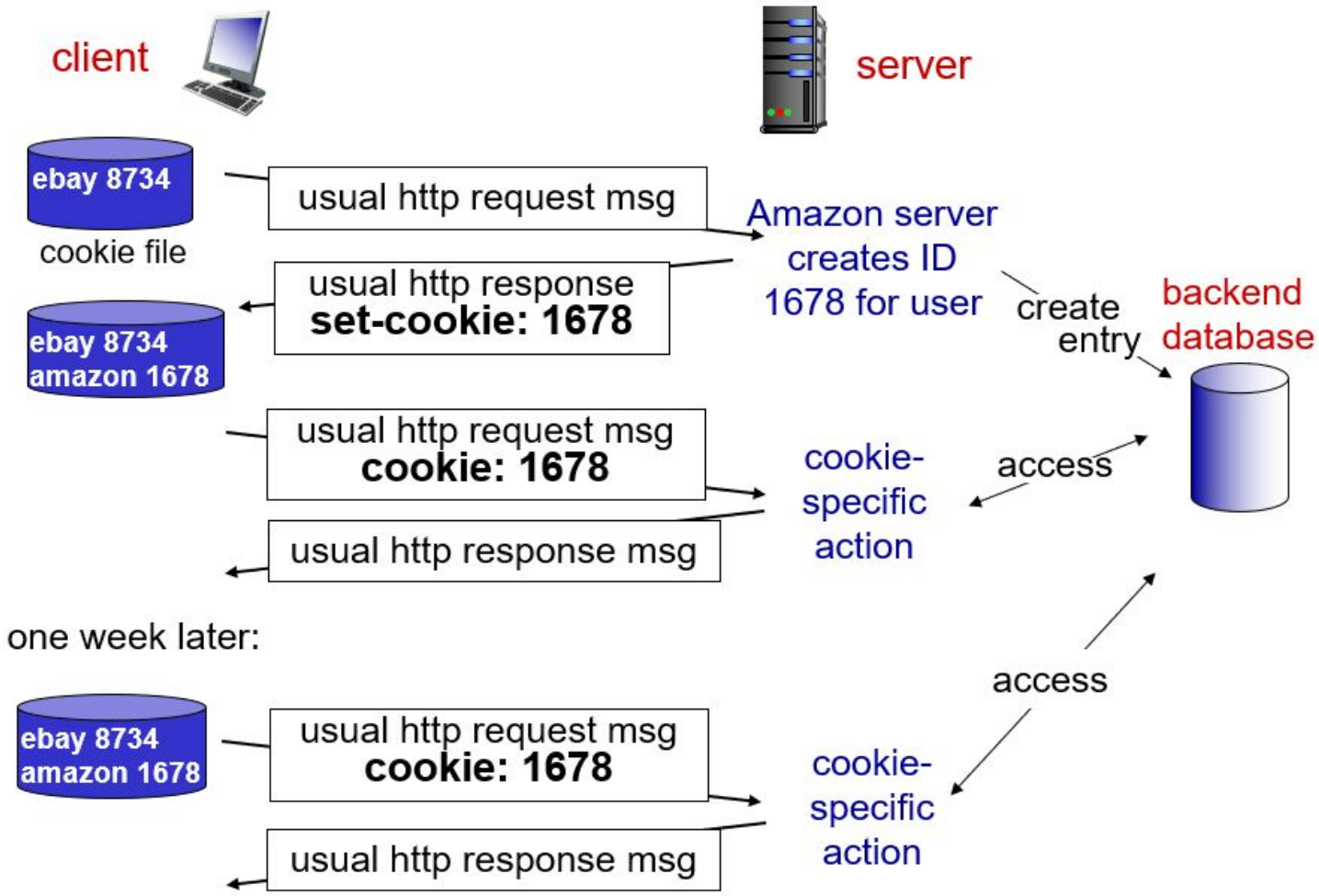
It has four components:

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, managed by user's browser
4) back-end database at Web site

E.g. Susan always access Internet from PC

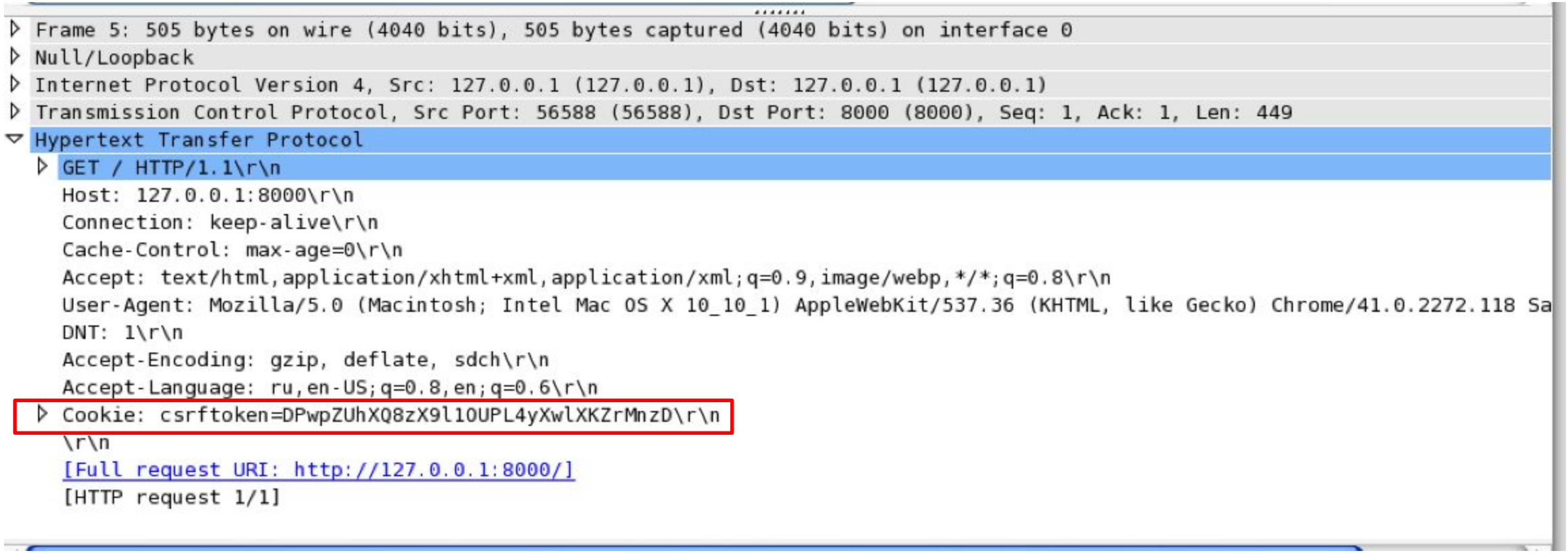
- visits specific e-commerce site for first time
- when initial HTTP requests arrives at site, site creates:
  - ✔ unique ID
  - ✔ entry in backend database for ID

Cookies: keeping “state” (cont.)



Cookies

Cookies: keeping “state” (cont.)



Cookies

What cookies can be used for:

- authorization
- shopping carts
- recommendations
- user session state (Web e-mail)

How to keep "state":

- protocol endpoints: maintain state at sender/receiver over multiple transactions
- cookies: http messages carry state

Cookies and privacy:

- cookies permit sites to learn a lot about you
- you may supply name and e-mail to sites

PES UNIVERSITY ONLINE

THANK YOU

Prajeesha

Department of Electronics and Communication Engineering