CIS 457 - Data Communications

Nathan Bowman

Images taken from Kurose and Ross book

Web Caching

Web cache sits between client and destination web server and satisfies requests on behalf of server

Also called **proxy server**

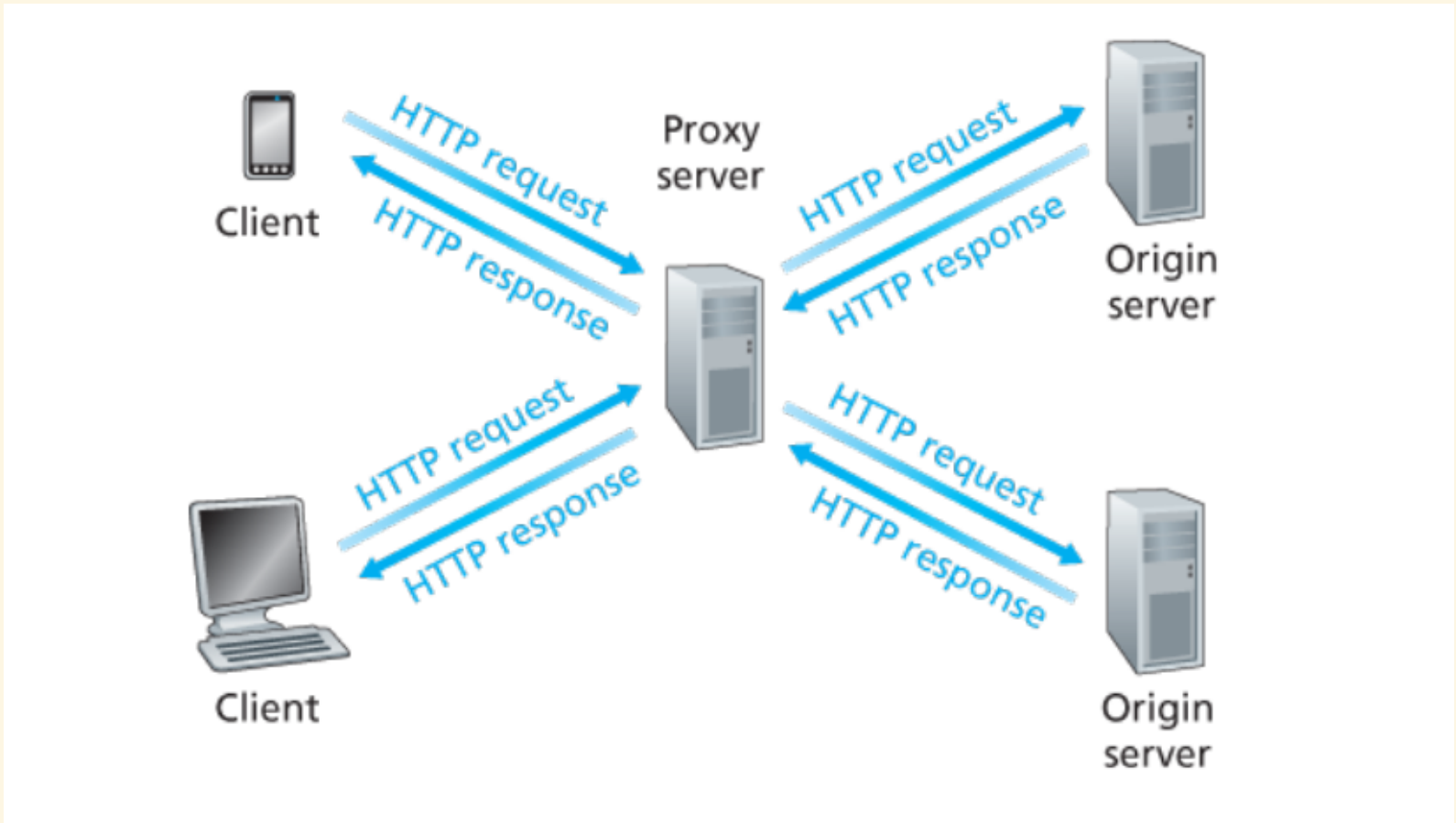
Used by ISPs to speed up web requests and save bandwidth costs

Web cache keeps track of recently requested objects in its local storage

Browser is configured so that when users request web pages, requests go to web cache instead

If cache has seen same request recently, it can respond immediately without needing to send message to destination web server

Otherwise, cache sends request to actual web server, forwards response from web server on to user and also stores response in case it is needed soon


Client
HTTP request
HTTP response
Proxy server
HTTP request
HTTP response
Origin server
Client
HTTP request
HTTP response
HTTP request
HTTP response
Origin server

Steps to requesting web page when cache is used

1. Connect to web cache via TCP and send HTTP request
2. If cache has local copy of object, returns it in HTTP response
3. Otherwise, cache opens TCP connection to origin server and sends HTTP request. Origin server sends requested object to cache in HTTP response
4. Cache stores copy of object locally and sends object in HTTP response message to client browser

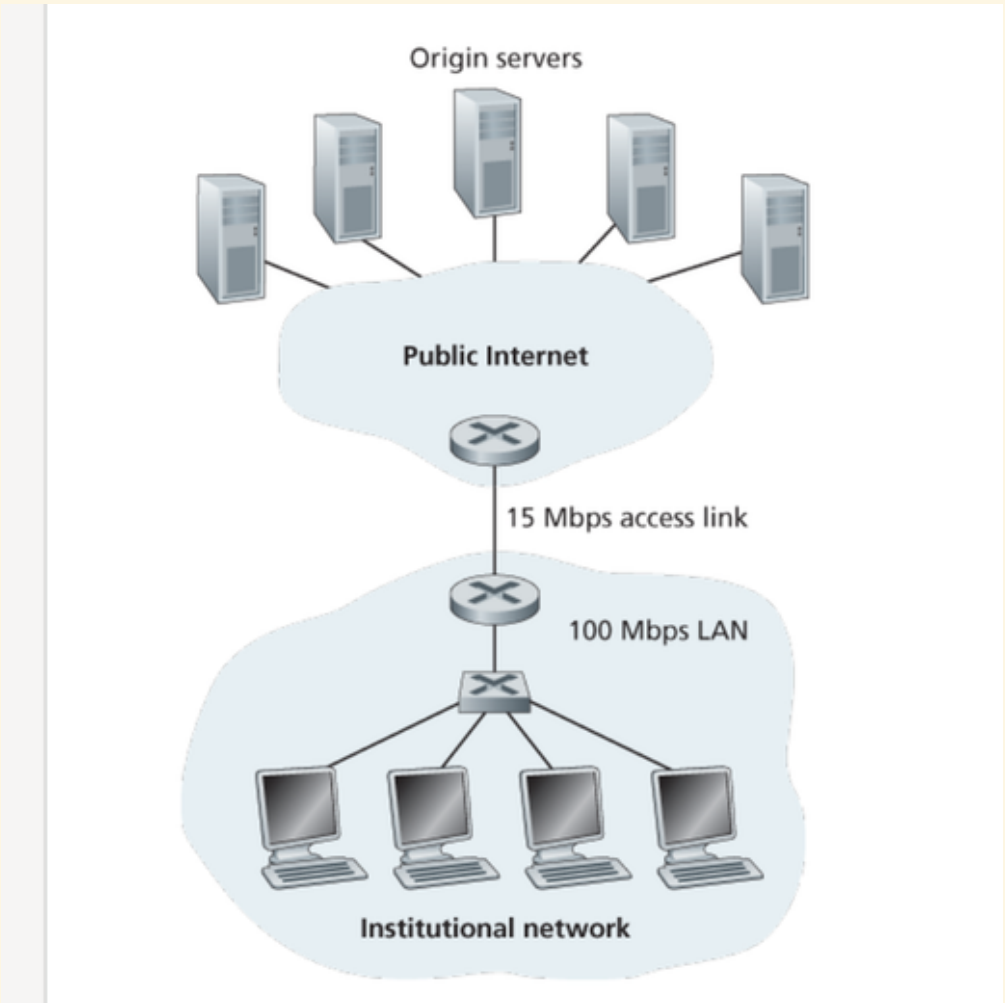
Note that cache acts as both server (to end user) and client (to web server)

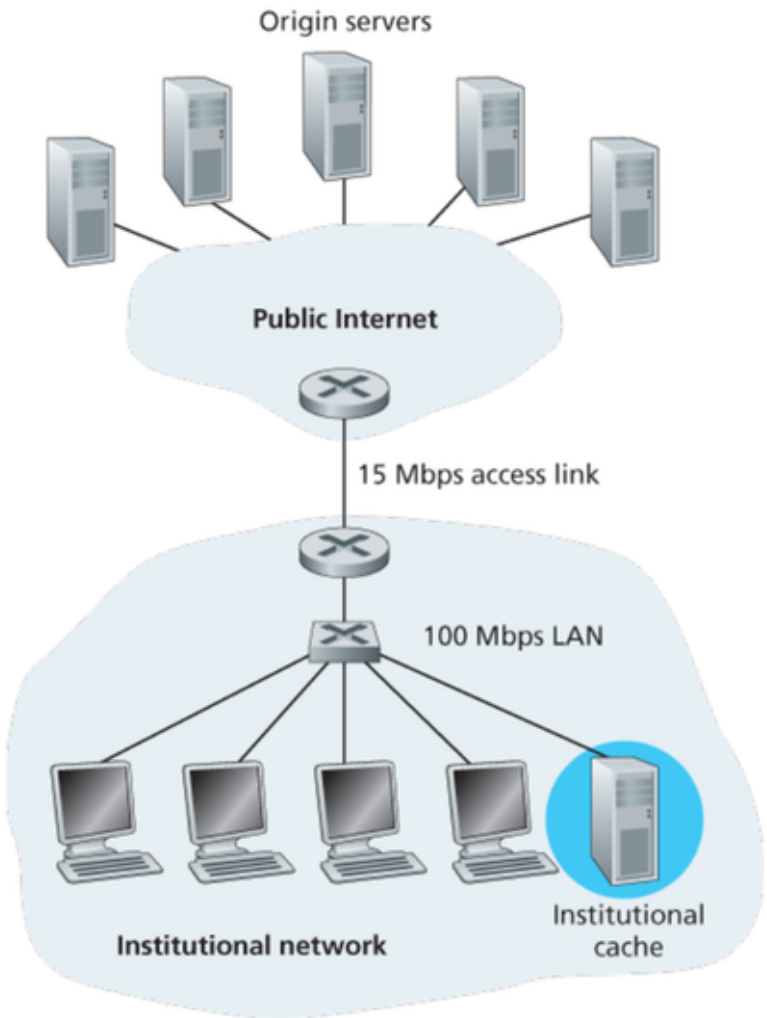
Why does ISP bother installing cache?

Recall that internet is network of networks, and ISP (such as a university) has to pay to connect its network to other networks

The more traffic over that connection, the higher the cost, so there is large interest in minimizing traffic that leaves local network

Web cache is installed on local network, so HTTP requests satisfied by web cache are "free" from perspective of ISP because they do not use bandwith connecting to other ISPs

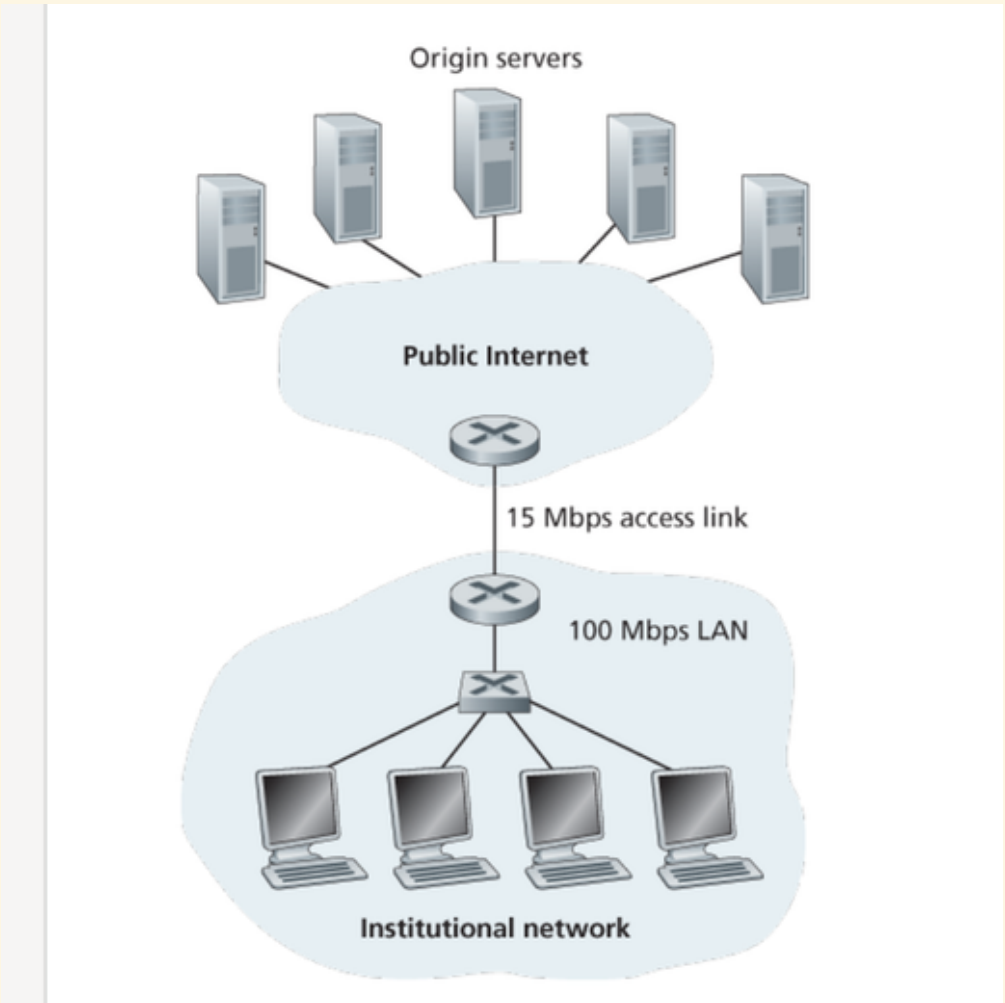

Origin servers
Public Internet
15 Mbps access link
100 Mbps LAN
Institutional network


Origin servers
Public Internet
15 Mbps access link
100 Mbps LAN
Institutional network
Institutional cache

Adding web cache can also greatly improve response time for end user

Time for HTTP request to get from end host to server and back can be broken down into three parts:

- time spent in local network ("LAN delay")
- time transfering between last router of ISP network and first router of public internet ("access delay")
- time spent in the rest of the internet ("internet delay")


Origin servers
Public Internet
15 Mbps access link
100 Mbps LAN
Institutional network

We can approximate speed up from web cache by using rough numbers to approximate delay

- Bandwidth on local network is 100 Mbps
- Bandwidth from local network to public internet is 15 Mbps
- Delay to get response from web server after request reaches public internet is 2 seconds

Assume average request is for object of size 1 Mbit and that 15 requests are generated per second

To figure out queueing delays, we must consider traffic intensity

On LAN: 15 * 1 Mbit / 100Mbps = 0.15

Across access link: 15 * 1 Mbit / 15 Mbps = 1

Traffic intensity of 0.15 on LAN generally results in tens of milliseconds of delay, so we disregard LAN

Access link will see queueing delays grow without bound

Average response time could be several minutes or more, which is unreasonable

How to fix it?

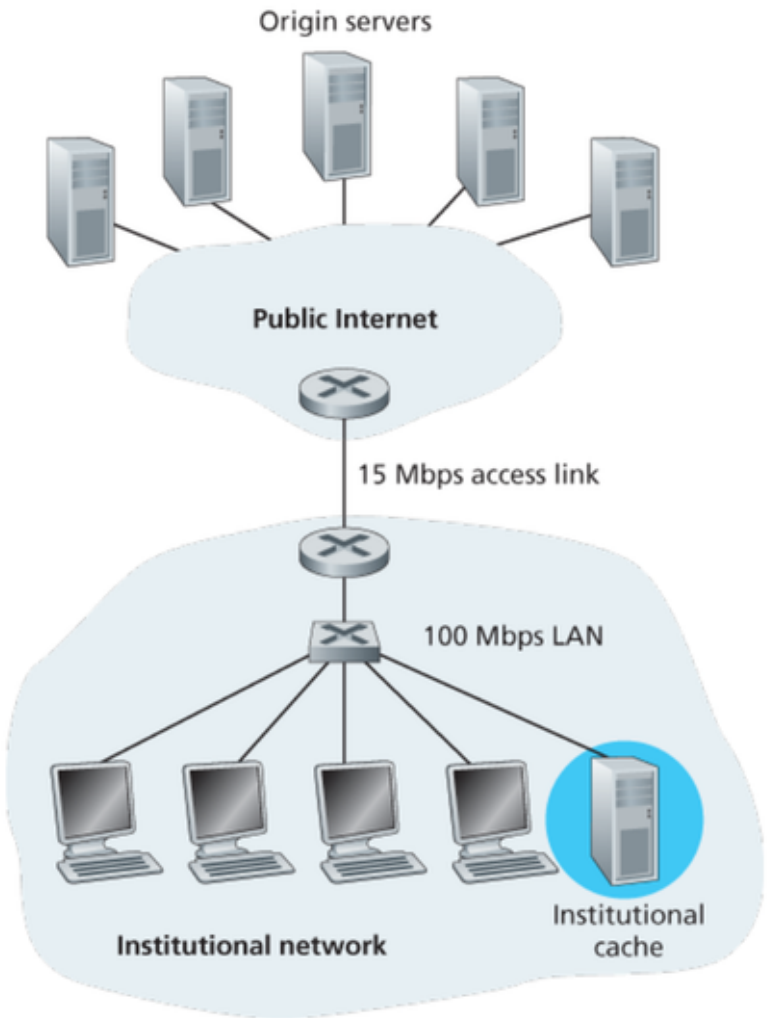
One way is to beef up access link

With access link at 100Mbps, intensity drops to 0.15, which results in negligible delays

Total response time consists of just internet delay: 2 seconds

However, this was an expensive fix

Better way is to use web cache inside institutional network


Origin servers
Public Internet
15 Mbps access link
100 Mbps LAN
Institutional network
Institutional cache

Hit rates fo web caches are typically 0.2 - 0.7 (20% - 70%)

Assume hit rate of 0.4

In that case, 40% of requests are satisfied almost immediately by cache: assume cache hit takes 10ms

What about remaining 60%?

Because traffic is reduced to 60%, traffic intensity on access link drops to 0.6.

Generally results in small delay: assume 10 milliseconds

Average delay:

$0.4 * 0.01 + 0.6 * 2.01$

= 1.2 seconds (approximately)

Cache solution actually much faster than (expensive) bandwidth solution