COMPUTER NETWORKS

Sivaraman Eswaran Ph.D.
Department of Computer Science and Engineering

COMPUTER NETWORKS

Application Layer

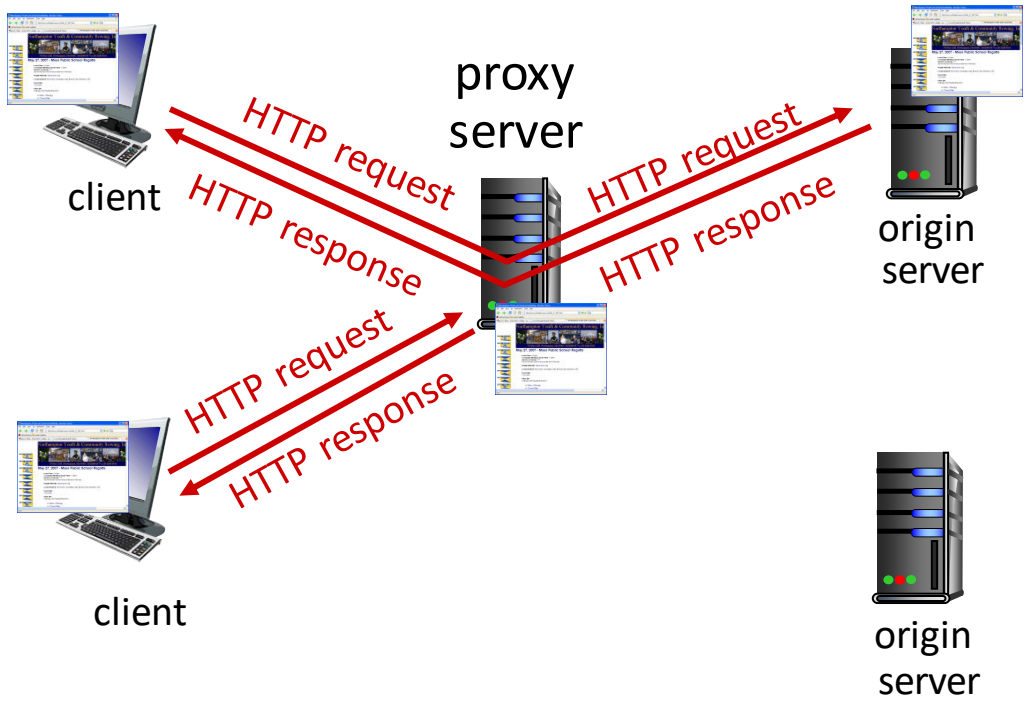
Sivaraman Eswaran Ph.D.
Department of Computer Science and Engineering

Unit – 2 Application Layer

Web Caches (Proxy Servers)

Goal: satisfy client request without involving origin server

- user configures browser to point to a *Web cache*
- browser sends all HTTP requests to cache
  - *if* object in cache: cache returns object to client
  - *else* cache requests object from origin server, caches received object, then returns object to client



Web Caches (Proxy Servers)

- Web 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 (speed)
  - cache is closer to client
- reduce traffic on an institution's access link (saves bandwidth)
- internet is dense with caches
  - enables "poor" content providers to more effectively deliver content
- privacy – surf the internet anonymously
- activity logging

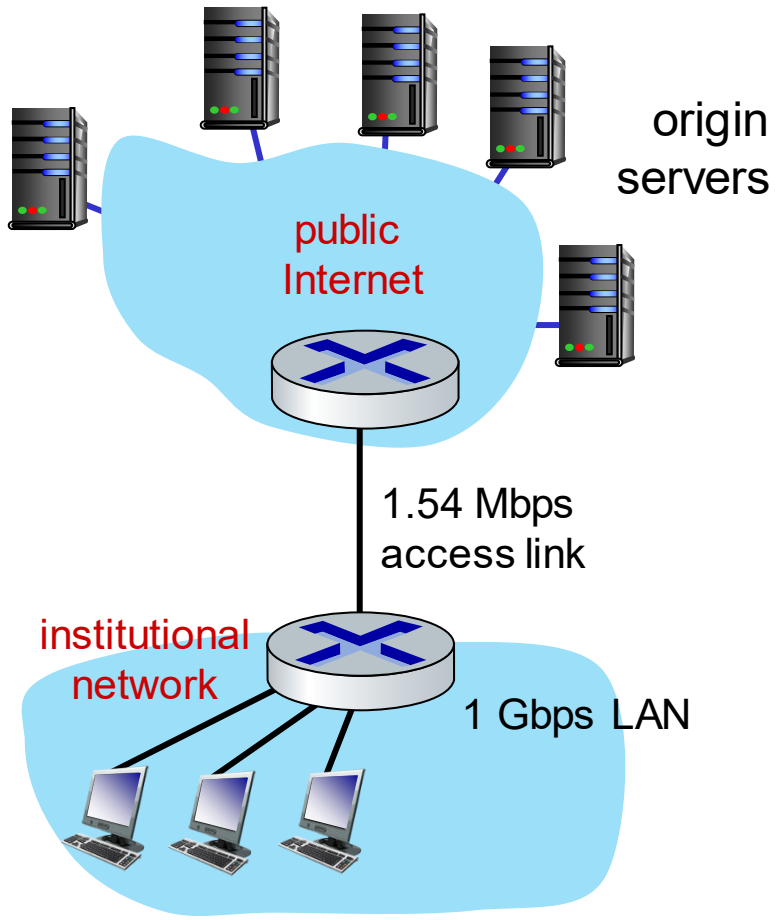
Caching example

(15 req/sec) * (100 Kbits/req)/(1.54 Mbps) = 0.974

(15 req/sec) * (100 Kbits/req)/(1 Gbps) = 0.0015

Scenario:

- access link rate: 1.54 Mbps
- RTT from institutional router to server: 2 sec
- Web object size: 100K bits
- Average request rate from browsers to origin servers: 15/sec
  - average data rate to browsers: 1.50 Mbps



Performance:

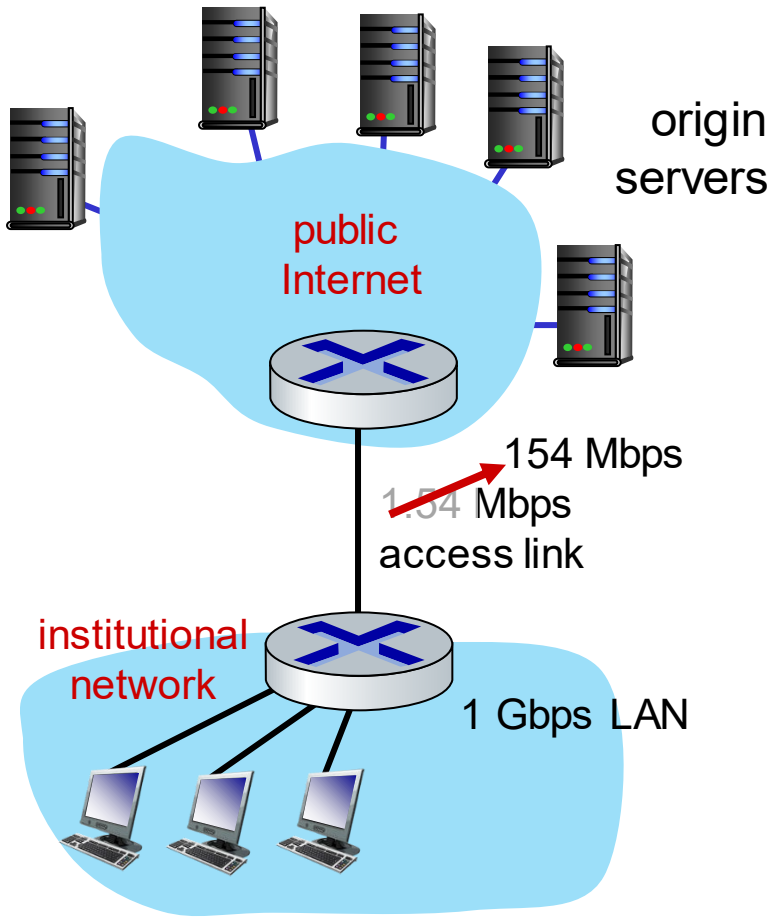
- LAN utilization: .0015
- access link utilization = .97
- end-end delay = Internet delay + access link delay + LAN delay
  = 2 sec + minutes + usecs

problem: large delays at high utilization!

Caching example: buy a faster access link

Scenario:

- access link rate: 1.54 Mbps → 154 Mbps
- RTT from institutional router to server: 2 sec
- Web object size: 100K bits
- Avg request rate from browsers to origin servers: 15/sec
  - avg data rate to browsers: 1.50 Mbps



Performance:

- LAN utilization: .0015
- access link utilization = .97 → .0097
- end-end delay = Internet delay + access link delay + LAN delay
  = 2 sec + minutes + usecs → msecs

Cost: faster access link (expensive!)

Caching example: install a web cache

Scenario:

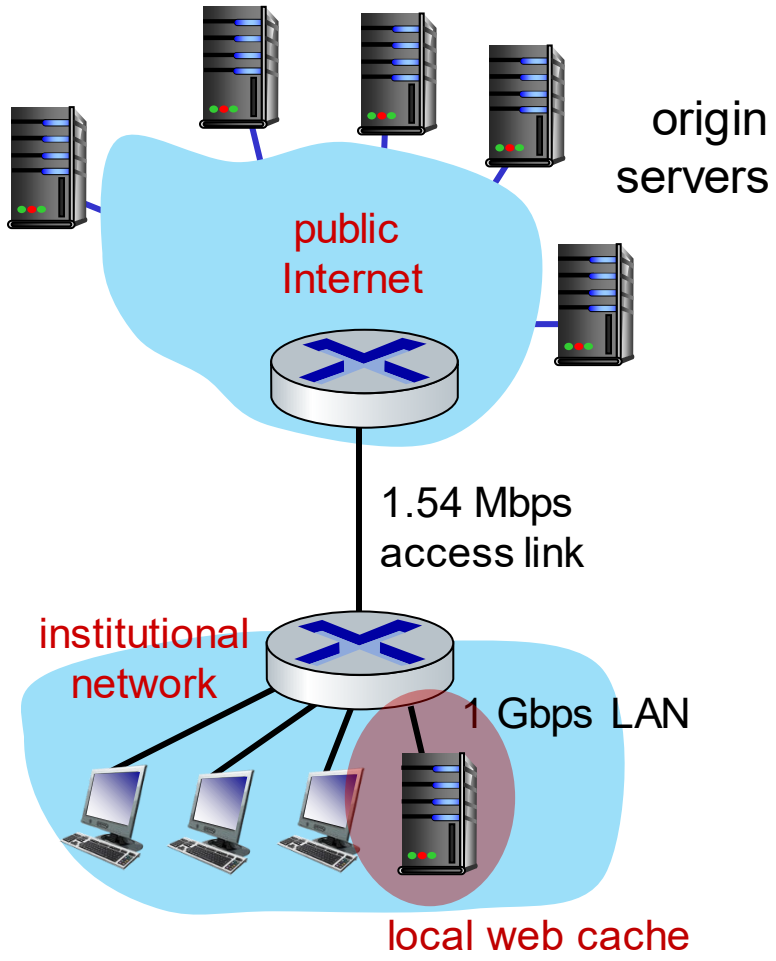
- access link rate: 1.54 Mbps
- RTT from institutional router to server: 2 sec
- Web object size: 100K bits
- Avg request rate from browsers to origin servers: 15/sec
  - avg data rate to browsers: 1.50 Mbps

Performance:

How to compute link utilization, delay?

- LAN utilization: .?
- access link utilization = ?
- average end-end delay = ?

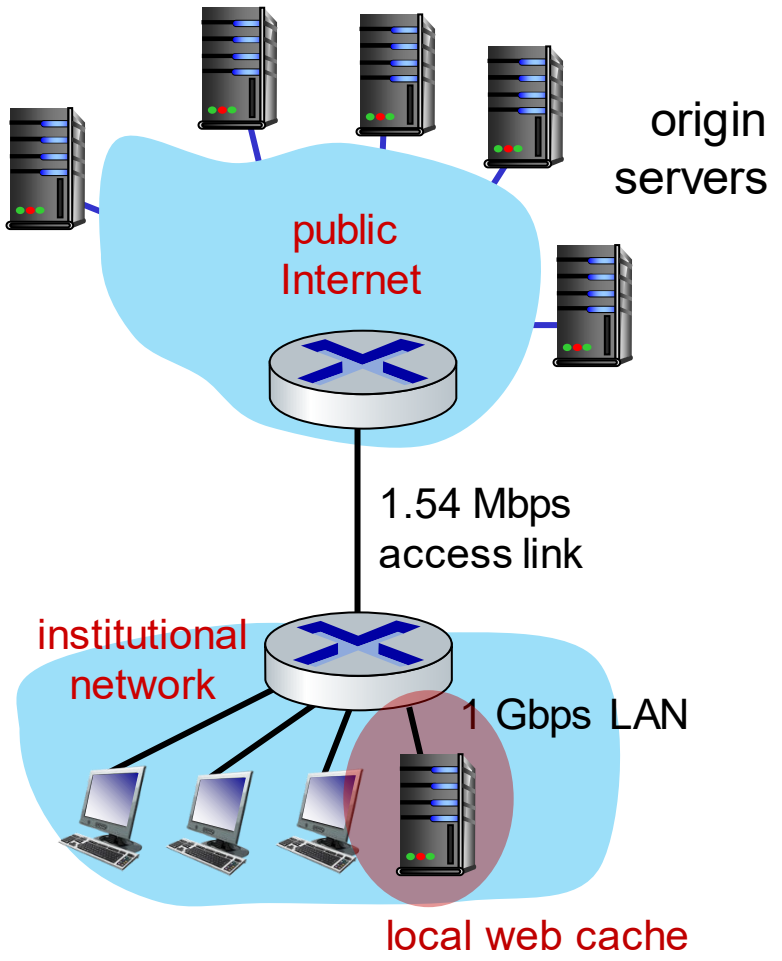
Cost: web cache (cheap!)



Caching example: install a web cache

Calculating access link utilization, end-end delay with cache:

- suppose cache hit rate is 0.4: 40% requests satisfied at cache, 60% requests satisfied at origin
- access link: 60% of requests use access link
- data rate to browsers over access link
  = 0.6 * 1.50 Mbps = .9 Mbps
- utilization = 0.9/1.54 = .58
- average end-end delay
  = 0.6 * (delay from origin servers)
  + 0.4 * (delay when satisfied at cache)
  = 0.6 (2.01) + 0.4 (~msecs) = ~ 1.2 secs



lower average end-end delay than with 154 Mbps link (and cheaper too!)

Conditional Get

Goal: don't send object if cache has up-to-date cached version

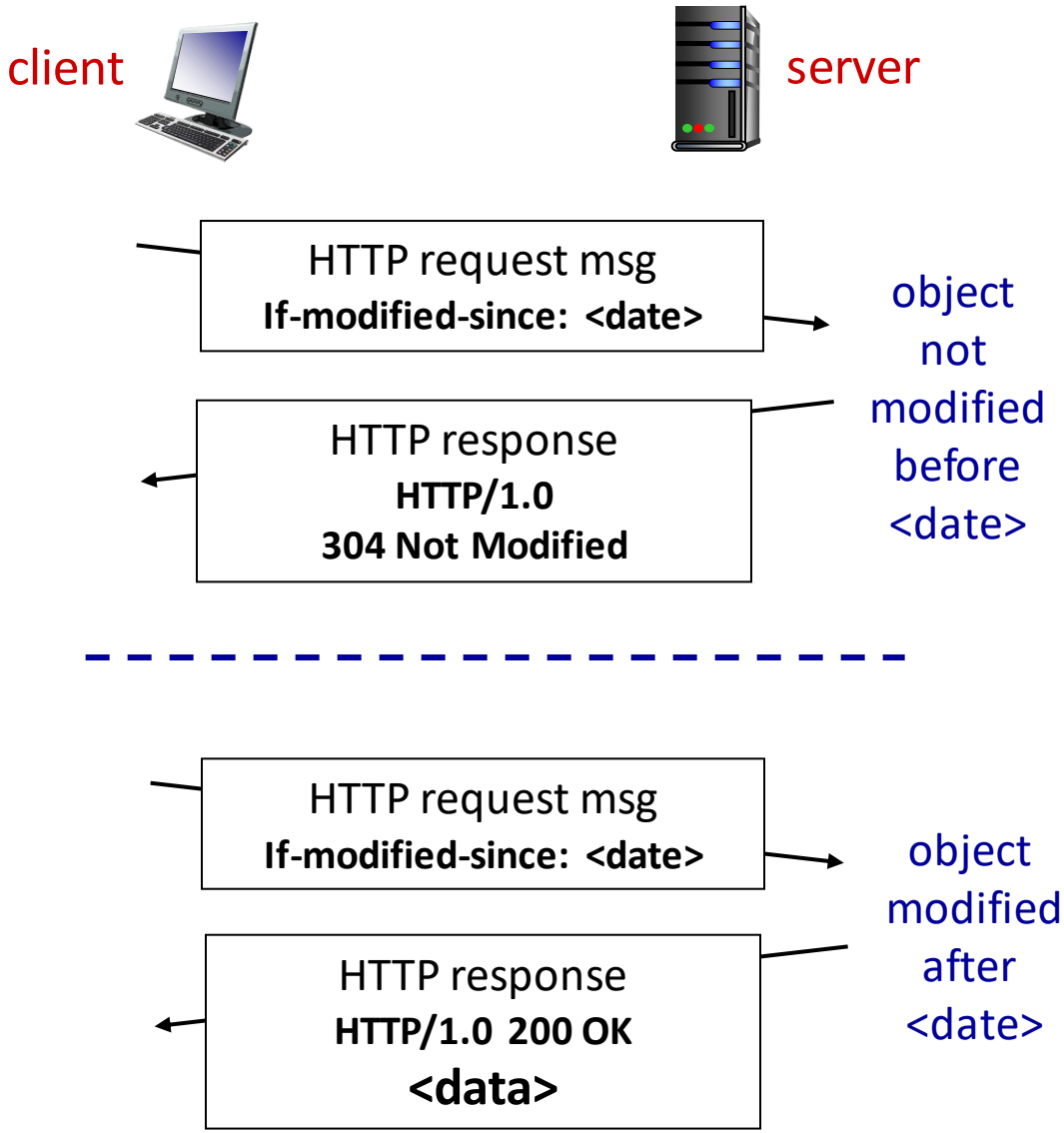
- no object transmission delay
- lower link utilization

- *cache:* specify date of cached copy in HTTP request

  If-modified-since: <date>

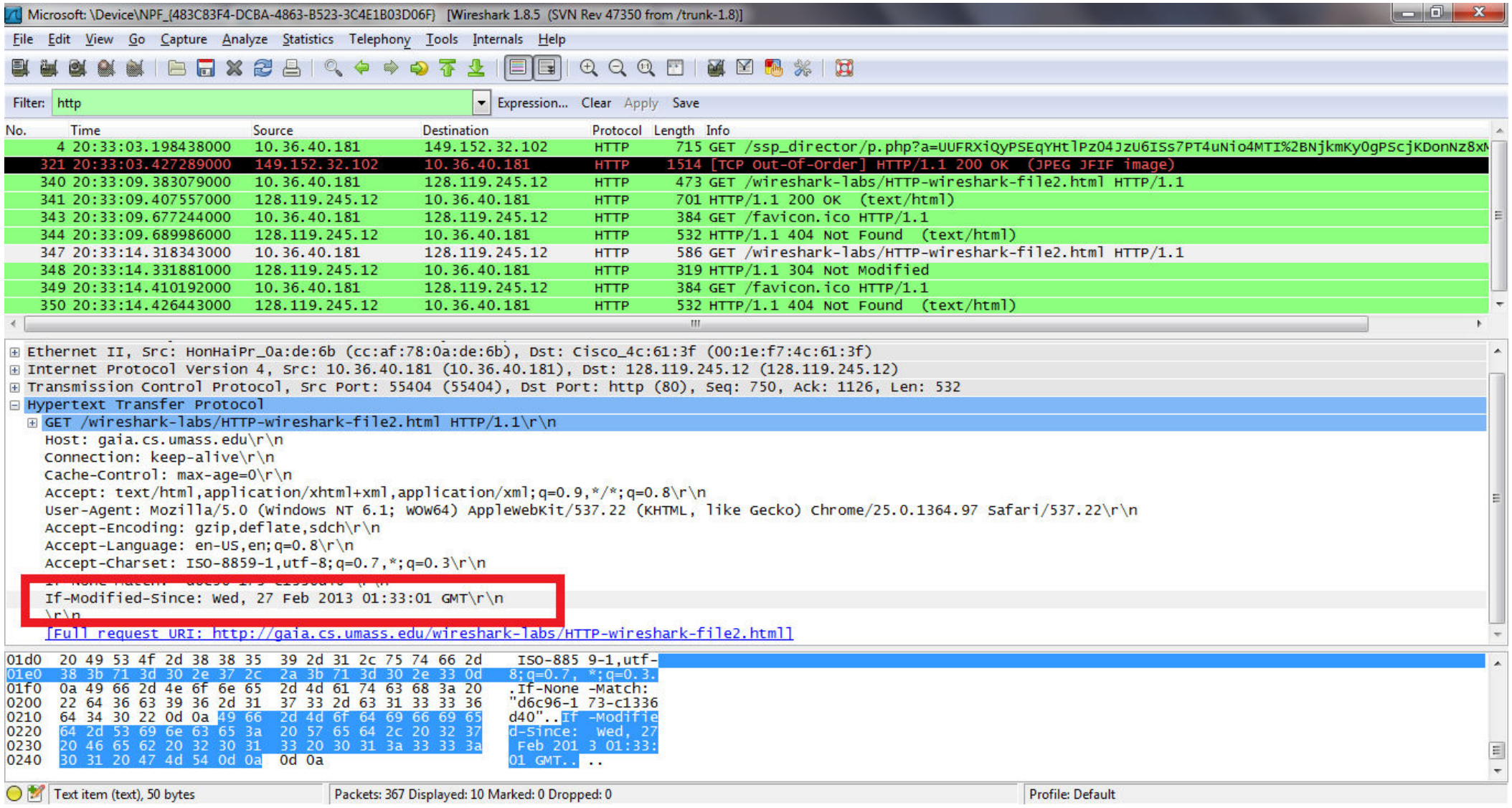
- *server:* response contains no object if cached copy is up-to-date:

  HTTP/1.0 304 Not Modified



COMPUTER NETWORKS

Conditional Get (more)



THANK YOU

Sivaraman Eswaran Ph.D.
Department of Computer Science and Engineering

sivaramane@pes.edu

+91 80 6666 3333 Extn 834