The Application Layer

File Transfer/FTP, Web/HTTP and Email

School of Software Engineering
South China University of Technology
Dr. Chunhua Chen
chunhuachen@scut.edu.cn

2019 Spring

Review

Architecture

- Client-Server model
- Peer-to-Peer (P2P) model
- Hybrid model

Services

- Reliability (data loss)
- Throughput
- Timing (delay)

TCP and UDP

Requirements of selected network applications

Application	Data Loss	Throughput	Time-Sensitive
File transfer/download	No loss	Elastic	No
Email	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Streaming stored audio /video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no

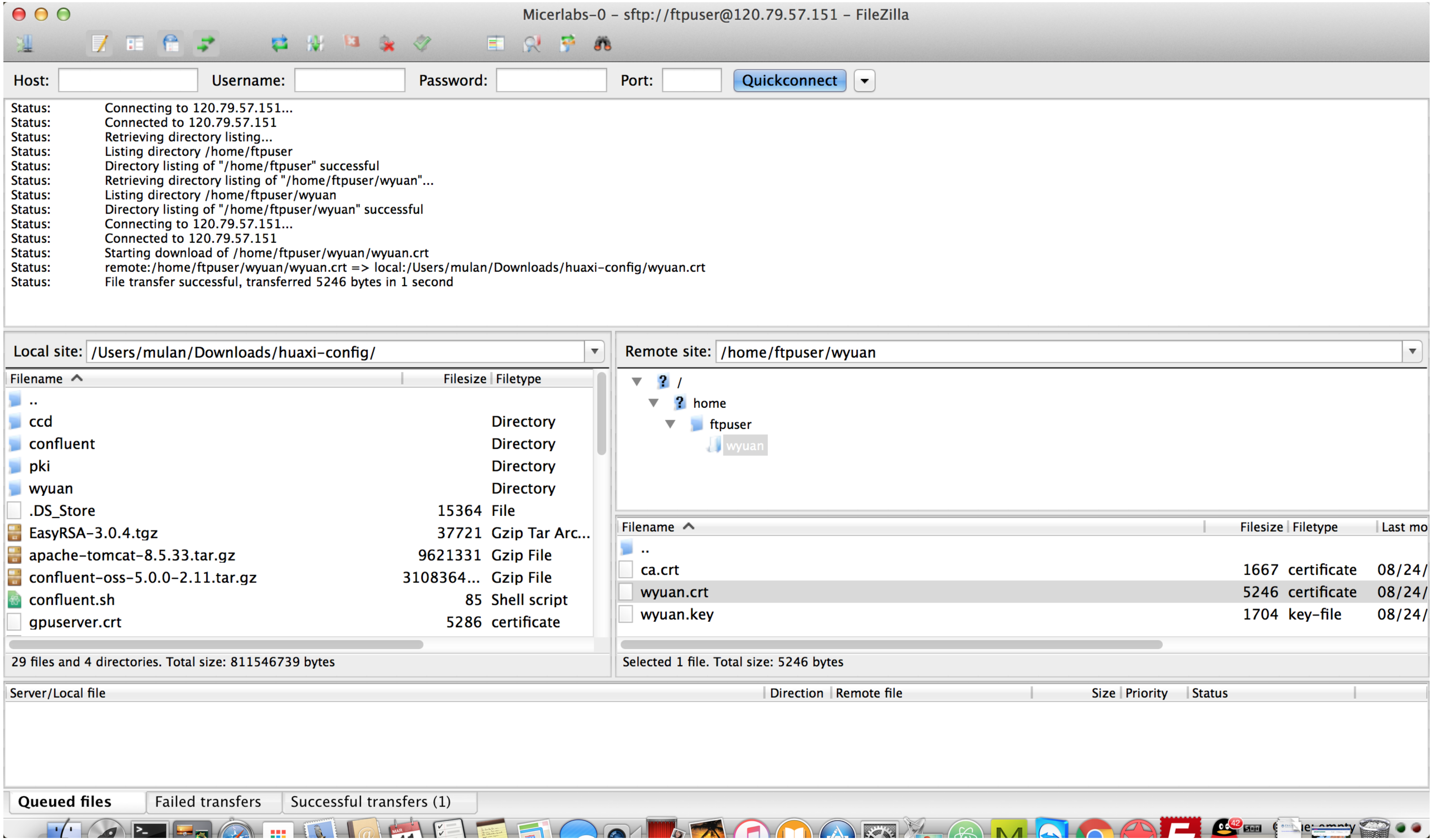
APP protocol

- Types of messages
- Syntax
- Semantics
- actions

```
GET /img/bd_logo1.png
HTTP/1.1

Host: www.baidu.com

Connection: close

User-agent: Mozilla/5.0

Accept-language: zh-CN
```

```
HTTP/1.1 200 OK

Accept-Ranges: bytes

Age: 3858

Cache-Control: max-age=315360000

Content-Length: 7877

Content-Type: image/png

Date: Wed, 30 Mar 2016 02:41:35 GMT


<html>

<head>

<title>bd_logo1.png (540×258)</title>

</head>

<body style="margin: 0px; background: #0e0e0e;">

<img style=“-webkit-user-select: none;cursor: zoom-in;”
src=“https://www.baidu.com/img/bd_logo1.png” width=“504” height=“240”/>

</body>

</html>
```

How to download a file from a remote server?



Micerlabs-0 – sftp://ftpuser@120.79.57.151 – FileZilla

```
Status:	Connecting to 120.79.57.151...
Status:	Connected to 120.79.57.151
Status:	Retrieving directory listing...
Status:	Listing directory /home/ftpuser
Status:	Directory listing of "/home/ftpuser" successful
Status:	Retrieving directory listing of "/home/ftpuser/wyuan"...
Status:	Listing directory /home/ftpuser/wyuan
Status:	Directory listing of "/home/ftpuser/wyuan" successful
Status:	Connecting to 120.79.57.151...
Status:	Connected to 120.79.57.151
Status:	Starting download of /home/ftpuser/wyuan/wyuan.crt
Status:	remote:/home/ftpuser/wyuan/wyuan.crt => local:/Users/mulan/Downloads/huaxi-config/wyuan.crt
Status:	File transfer successful, transferred 5246 bytes in 1 second
```

Local site: /Users/mulan/Downloads/huaxi-config/

Filename	Filesize	Filetype
..		
ccd		Directory
confluent		Directory
pki		Directory
wyuan		Directory
.DS_Store	15364	File
EasyRSA-3.0.4.tgz	37721	Gzip Tar Arc...
apache-tomcat-8.5.33.tar.gz	9621331	Gzip File
confluent-oss-5.0.0-2.11.tar.gz	3108364...	Gzip File
confluent.sh	85	Shell script
gpuserver.crt	5286	certificate

29 files and 4 directories. Total size: 811546739 bytes

Remote site: /home/ftpuser/wyuan

```
▼ ? /
  ▼ ? home
    ▼ ftpuser
        wyuan
```

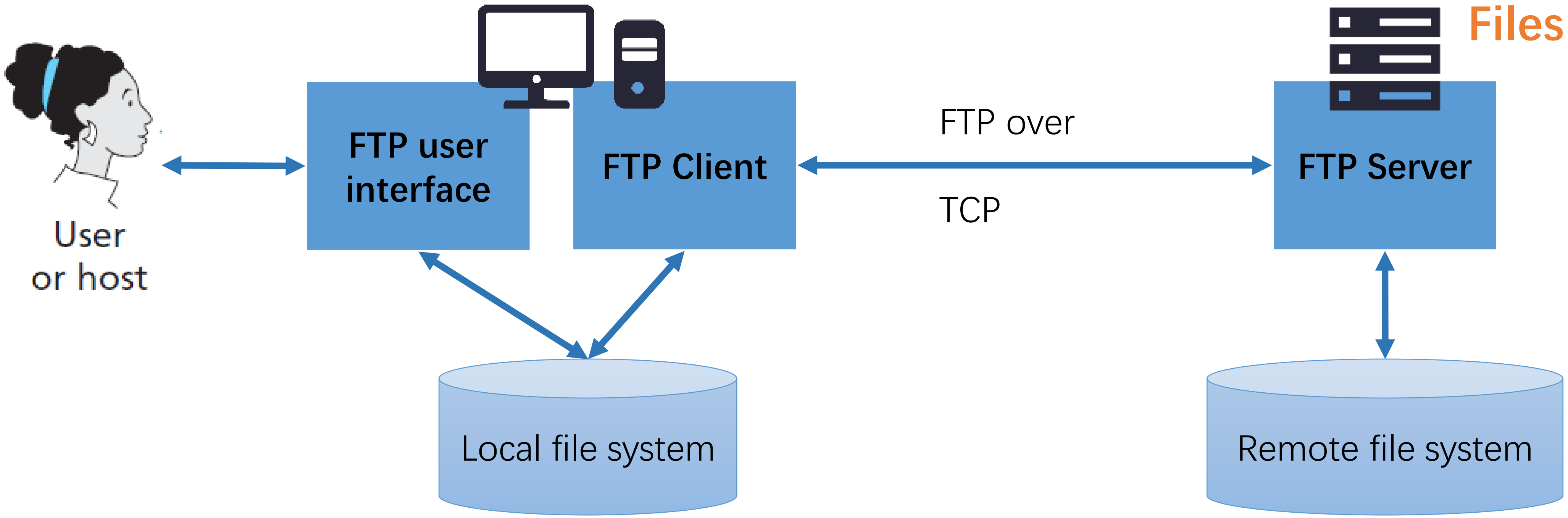
Filename	Filesize	Filetype	Last mo
..			
ca.crt	1667	certificate	08/24/
wyuan.crt	5246	certificate	08/24/
wyuan.key	1704	key-file	08/24/

Selected 1 file. Total size: 5246 bytes

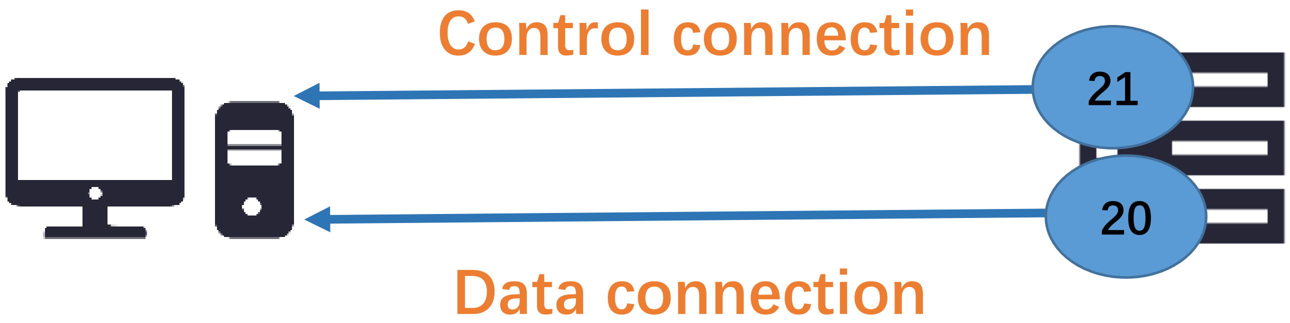
Server/Local file	Direction	Remote file	Size	Priority	Status

File Transfer: FTP

- Client-Server model
- Run on top of TCP, used for download and upload files to FTP Server



FTP: Stateful control and data connection



the FTP server must maintain **state** about the user

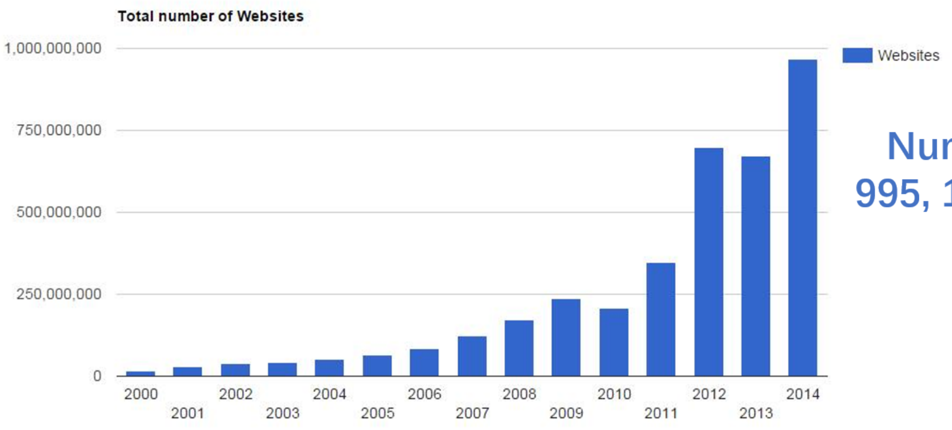
- The commands, from client to server, and replies, from server to client, are sent across the control connection in 7-bit ASCII format.
- Each command consists of four uppercase ASCII characters, some with optional arguments.
- **USER username**: Used to send the user identification to the server.
- **PASS password**: Used to send the user password to the server.
- **LIST**: Used to ask the server to send back a list of all the files in the current remote directory.
- **RETR filename**: Used to retrieve (that is, get) a file from the current directory of the remote host. This
- **STOR filename**: Used to store (that is, put) a file into the current directory
- of the remote host.

The history of Web

- Invented by Sir Tim Berners-Lee, 1989

How big is the Internet?

@http://www.internetlivestats.com/total-number-of-websites/

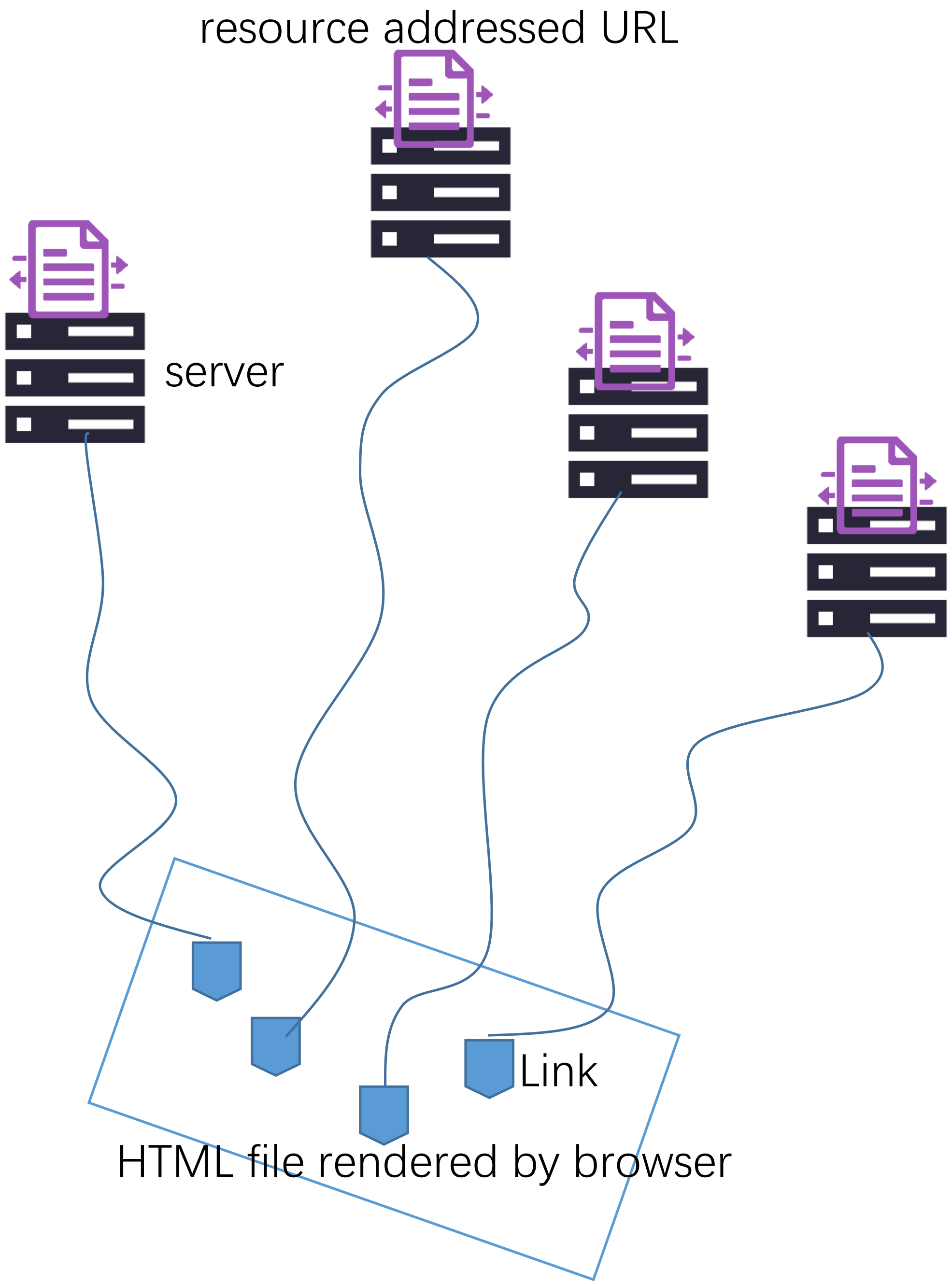


Number of websites 995, 152, 321 at Feb 27, 2016
约 10亿

Sir Tim Berners-Lee

My Visions of Web

- URL (Uniform Resource Locator): Assign an unique address for every resources on the Internet
- HTML: structure relevant resources in to a single document (known as HTML file or Web page) using Hyper Text Mark-up Language (HTML), inform of links (display names associated with URLs)
- HTTP: use Hyper Text Transfer Protocol (HTTP) to download resources on demand.
- A resource also called an object, can be an HTML file, a JPEG image, a Java applet, or a video clip, etc.
- Two components:
  - Web servers, storing resources with addressable URLs
  - Web browsers used by end users to download and use resources from web servers



The URL (or URI)

Addressing every resources in the Web

http://www.baidu.com/img/bd_logo1.png

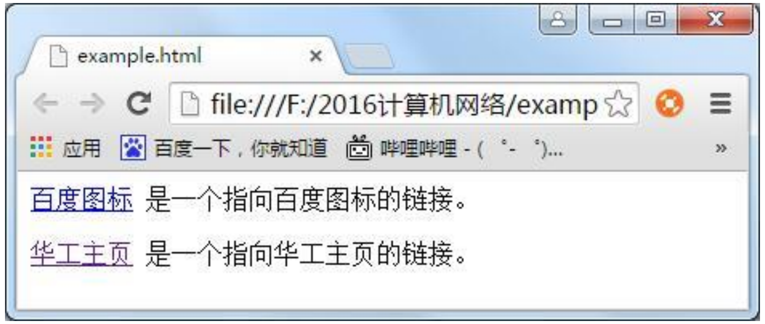
http://www.baidu.com	/img/bd_logo1.png
Host Name globally unique	Path Name Unique within Host

What about relationship between a host name and IP?

- The Domain Name System

The Web Documents (or Pages)

```
<html>
<body>
<p>
<a href="http://www.baidu.com/img/bd_logo1.png">百度图标</a>
是一个指向百度图标的链接。</p>
<p>
<a href="http://www.scut.edu.cn/">华工主页</a>
是一个指向华工主页的链接。</p>
</body>
</html>
```



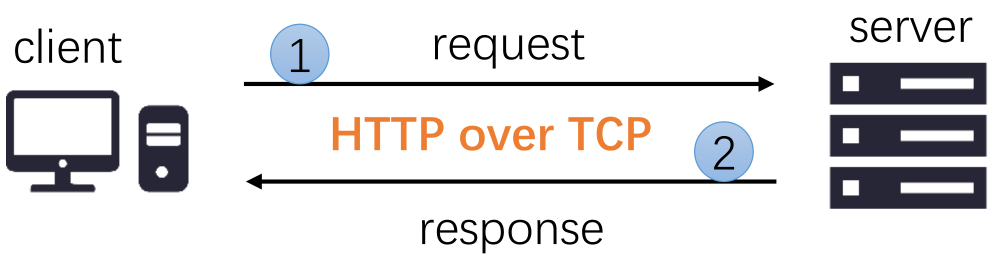
百度图标 是一个指向百度图标的链接。

华工主页 是一个指向华工主页的链接。

华南理工大学
South China University of Technology
电子邮件 | 办公门户 | 网络服务 | 大学城校区 | 校长信箱 | ENGLISH | 旧版网站
学校概况
机构设置
人才培养
科学研究
招生就业
图书馆
校园生活
人才招聘
合作交流
聚華佳圖
学校要闻
进入新闻网»
教学科研通知
更多
热点关注
更多
让追求卓越成为习惯 标杆工程十大卓越团队揭晓
关于开展2015—2016学年度第二学期教师教学能力专项
华南理工大学"岭南追梦"大学生社会主义核心价值

The HTTP Protocol

message types, message syntax and message semantics



client

1 request

HTTP over TCP

2 response

server

HTTP 1.0: RFC 1945
HTTP 1.1: RFC 2068

```
GET /img/bd_logo1.png HTTP/1.1
Host: www.baidu.com
Connection: close
User-agent: Mozilla/5.0
Accept-language: zh-CN
```

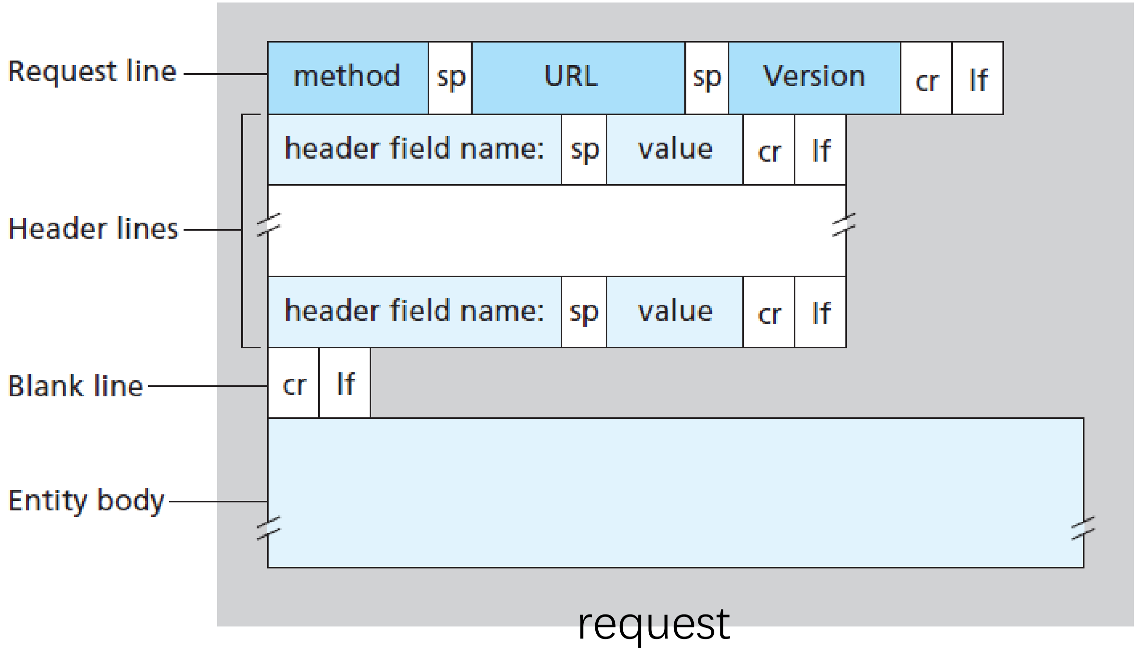
request

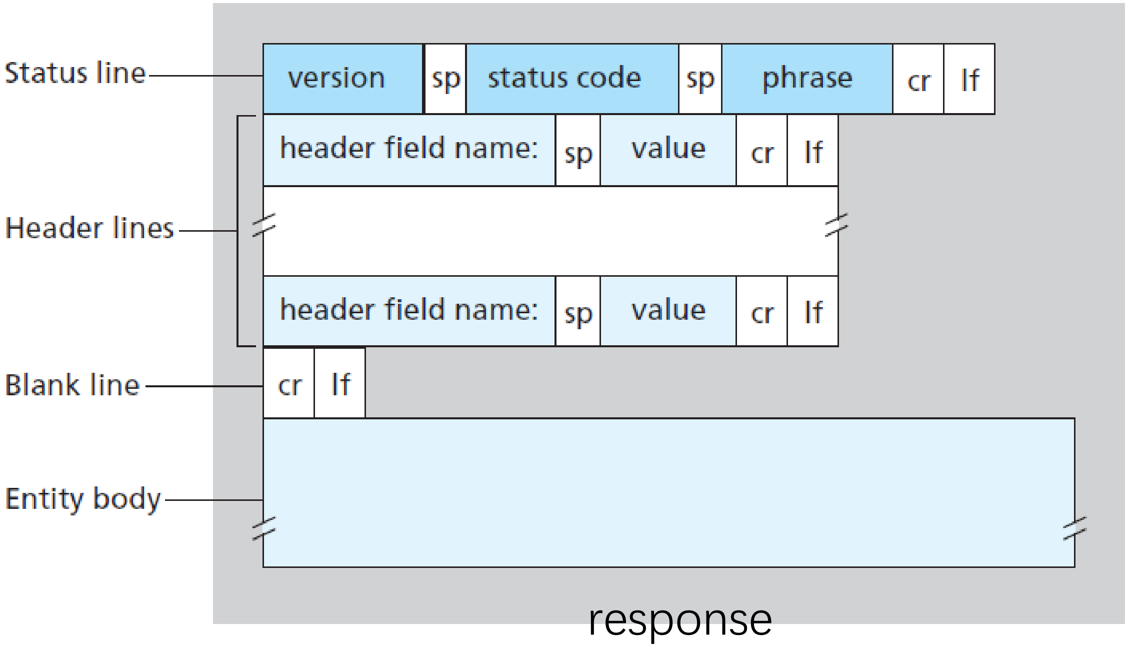
```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Age: 3858
Cache-Control: max-age=315360000
Content-Length: 7877
Content-Type: image/png
Date: Wed, 30 Mar 2016 02:41:35 GMT
ETag: "1ec5-502264e2ae4c0"
Expires: Sat, 28 Mar 2026 02:41:35 GMT
Last-Modified: Wed, 03 Sep 2014 10:00:27 GMT
Server: Apache
X-Cache-Lookup: HIT from jpvip175.nydus-vpn.com:3222
```

response

The HTTP Protocol

message types, message syntax and message semantics





Method: GET, POST, HEAD, PUT, and DELETE

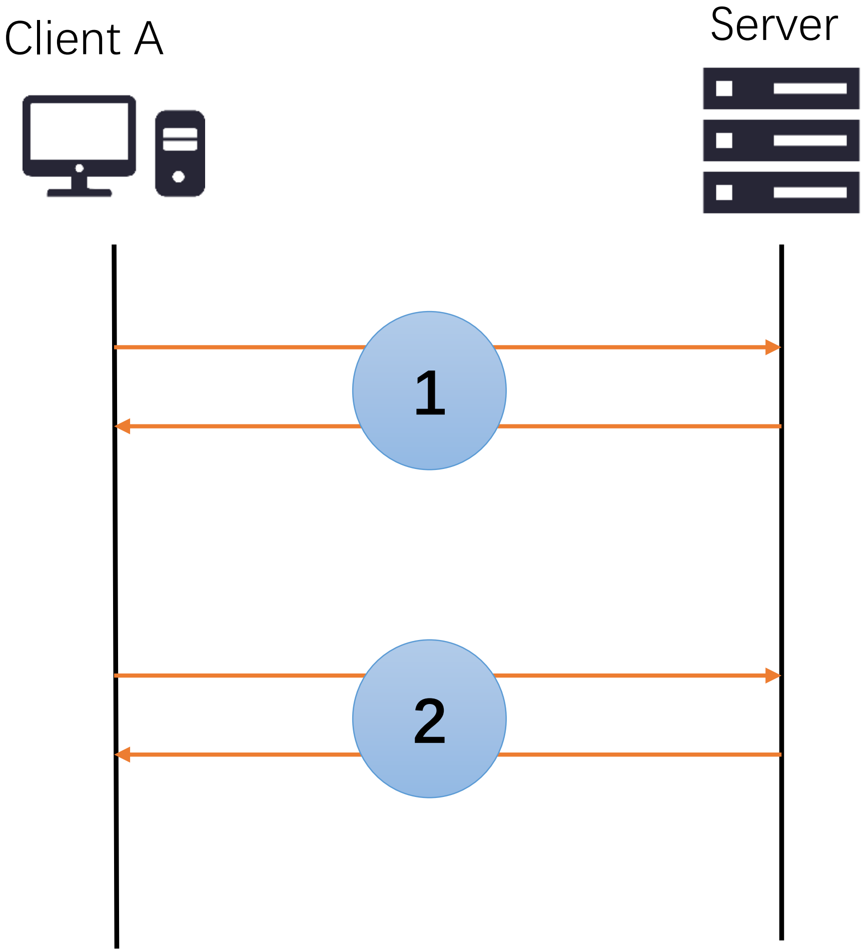
Status code
200 OK
301 Moved Permanently
400 Bad Request
404 Not Found
505 HTTP Version Not Supported

The HTTP Protocol

stateless connections

- Stateless?
  - server maintains no information about past client requests
  - Can the server trace a particular client?

- Protocols that maintain **state** are complex!
  - past history (state) must be maintained
  - if server/client crashes, their views of state may be inconsistent, must be reconciled



Communication 1 and 2 come from the same client, can server know that? If yes, how?

The HTTP Protocol

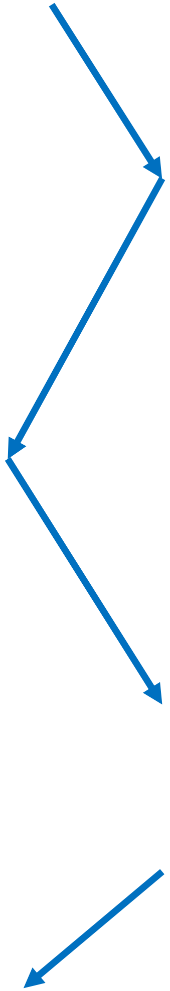
non-persistent or persistent connections

- Non-persistent: HTTP 1.0
  - each request/response pair be sent over a separate TCP connection
- Persistent: HTTP 1.1
  - multiple request/response pairs be sent over the same TCP connection

Non-persistent HTTP

try www.scut.edu.cn/index.html

1a. HTTP client initiates TCP connection to HTTP server (process) at www.someSchool.edu on port 80



1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80. “accepts” connection, notifying client

2. HTTP client sends HTTP *request message* (containing URL) into TCP connection socket. Message indicates that client wants object /index.html

3. HTTP server receives request message, forms *response message* containing requested object, and sends message into its socket

4. HTTP server requests to close TCP connection.

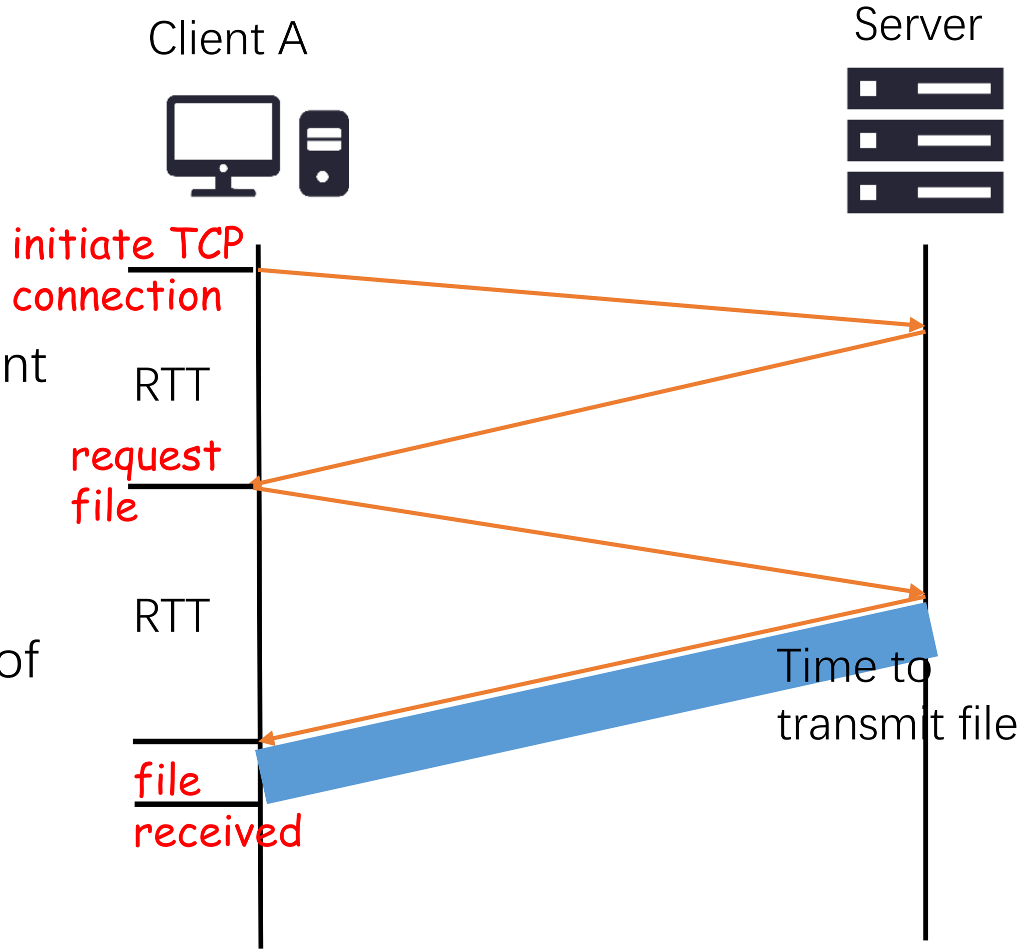
5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-5 repeated for each of 10 jpeg objects

Non-persistent HTTP

response time

- Definition of RTT
  - time to send a small packet to travel from client to server and back.
- Response time
  - one RTT to initiate TCP connection
  - one RTT for HTTP request and first few bytes of HTTP response to return
  - file transmission time
- total = 2RTT+transmit time



Persistent HTTP

- Non-persistent HTTP issues:
  - requires 2 RTTs per object
  - OS overhead for each TCP connection
  - browsers often open parallel TCP connections to fetch referenced objects
- Persistent HTTP
  - server leaves connection open after sending response
  - subsequent HTTP messages between same client/server sent over open connection

Persistent **without pipelining**:

- client issues new request only when previous response has been received
- one RTT for each referenced object

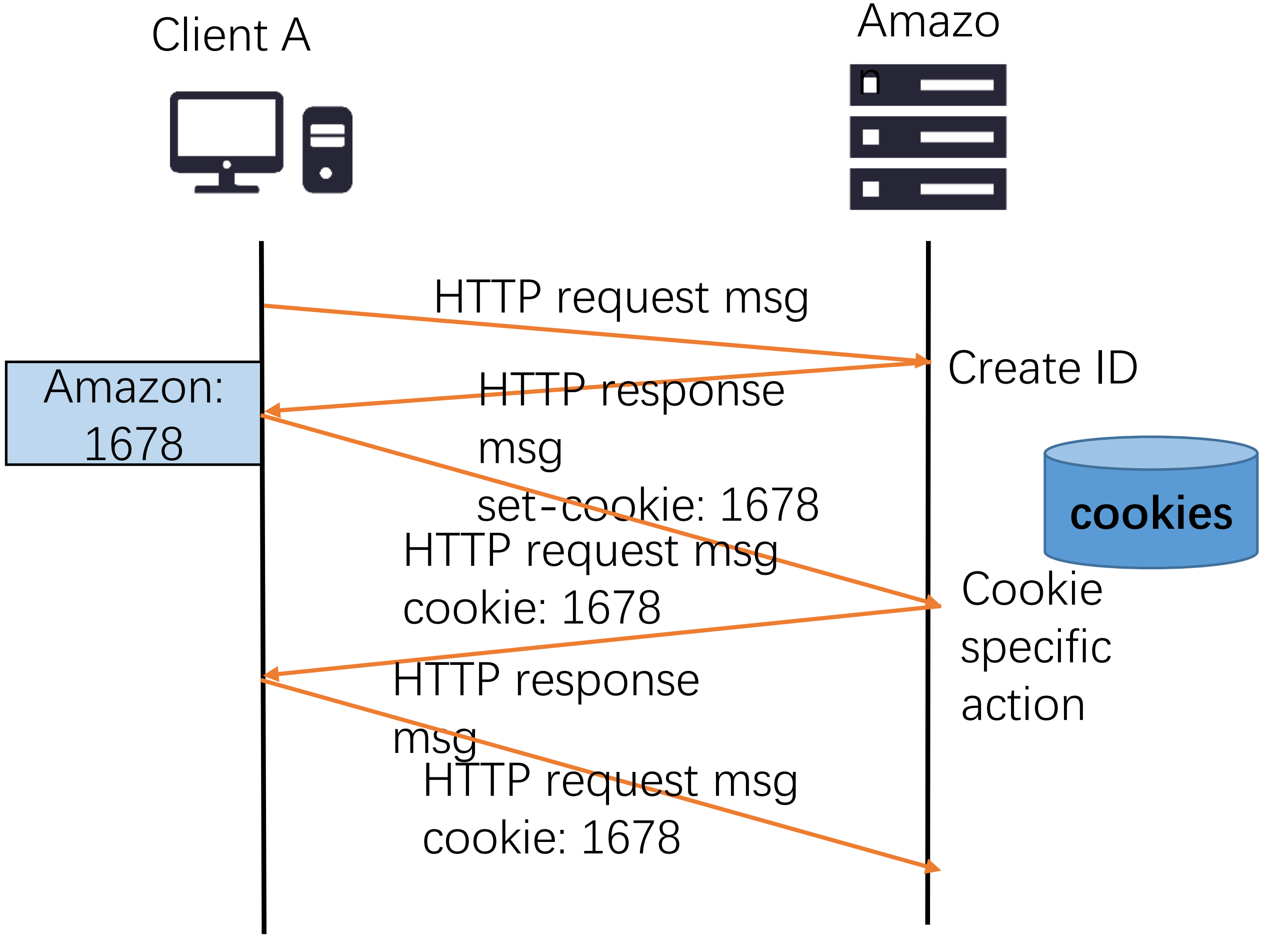
Persistent **with pipelining**:

- default in HTTP/1.1
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects

User-Server Interaction: Cookies

how can server trace users

- Cookie is a value create and maintained by a server, sent to and maintained by a client.
- In every sequent requests, the client sends cookie with the requests for the server to identify the client.
- Can set life time for a cookie value.



Cookies (continued)

What cookies can bring:

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

Cookies and privacy:

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

How to keep “state”:

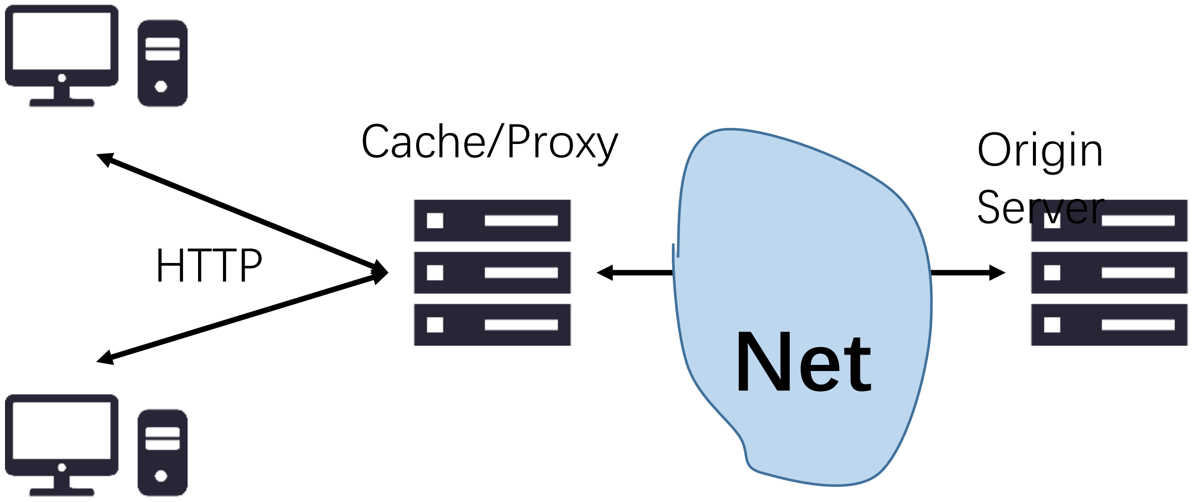
- protocol endpoints: maintain state at sender/receiver over multiple transactions
- cookies: http messages carry state

Web Caching (Proxy server)

storing content closer to end users

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



More about web caching

- cache acts as both client and server
- typically cache is installed by ISP (university, company, residential ISP)

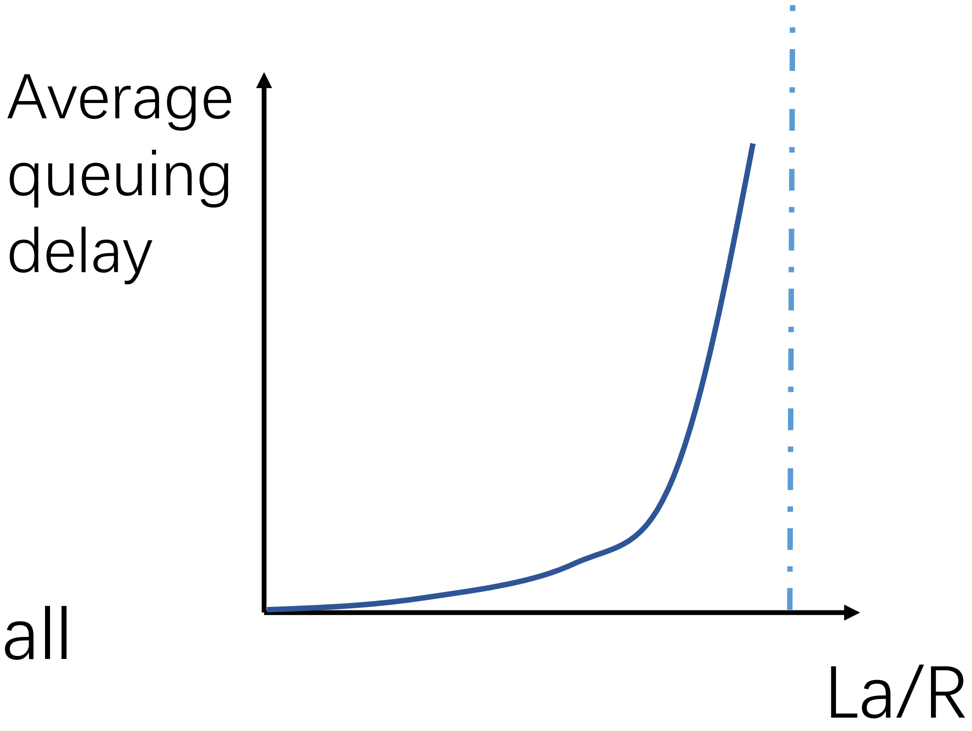
Why Web caching?

- 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 (but so does P2P file sharing)

Web cache of computer network vs Storage cache of computer system

Queuing Delay and Traffic Intensity

- R=link bandwidth (bps)
- L=packet length (bits)
- a=average packet arrival rate



Traffic Intensity = La/R

- La/R ~ 0: average queuing delay small
- La/R -> 1: delays become large
- La/R > 1: more “work” arriving than can be serviced, average delay infinite!

Caching example

the access link is the bottleneck

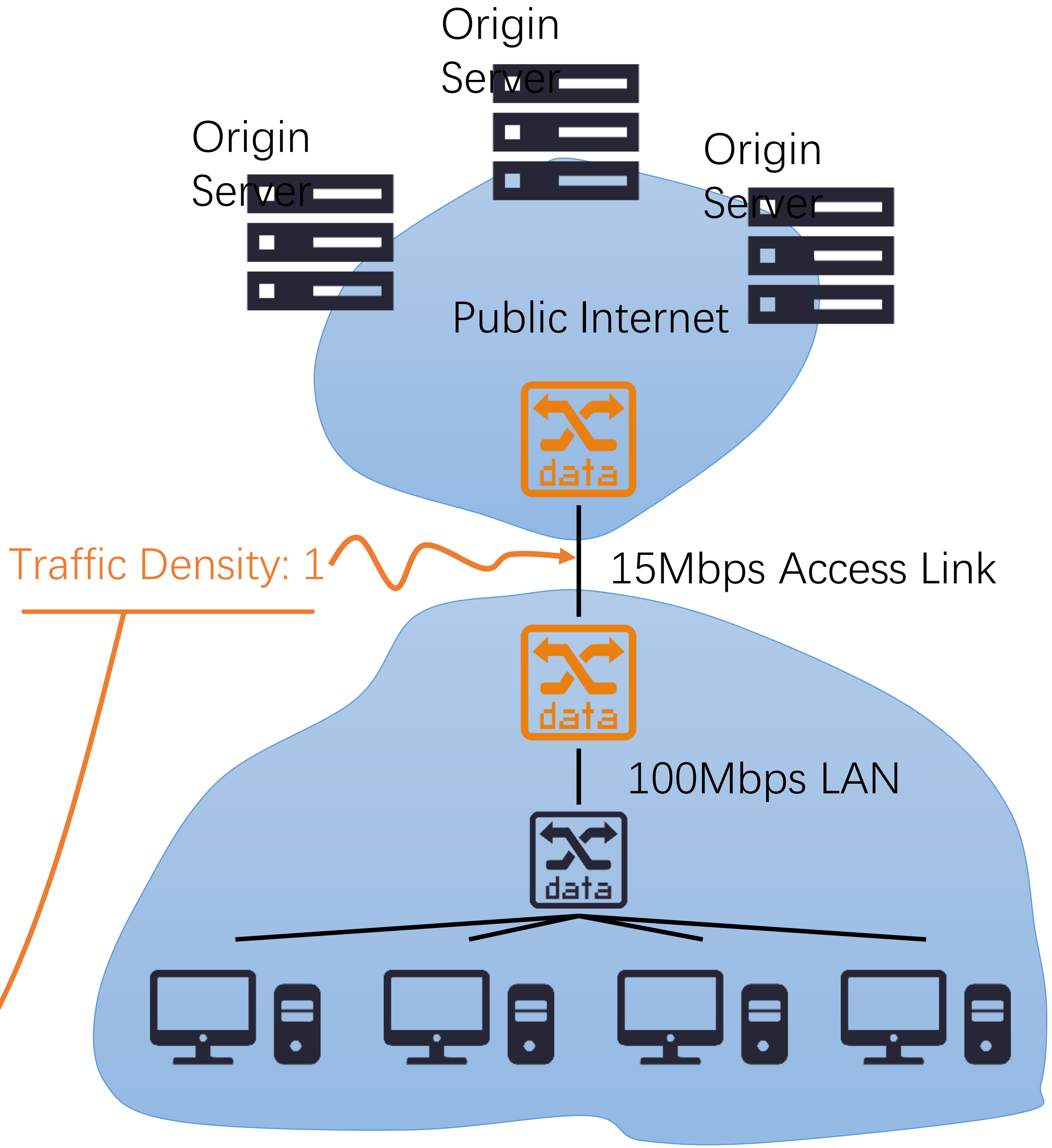
Assumptions

- average object size = 1Mbits
- avg. request rate from institution's browsers to origin servers = 15/sec
- delay from institutional router to any origin server and back to router = 2 sec

Consequences

- utilization on LAN = 15%
- utilization on access link = 100%
- total delay = Internet delay + access delay + LAN delay

= 2 sec + **minutes** + milliseconds



Caching example (continue)

kill the bottleneck by increasing it capability

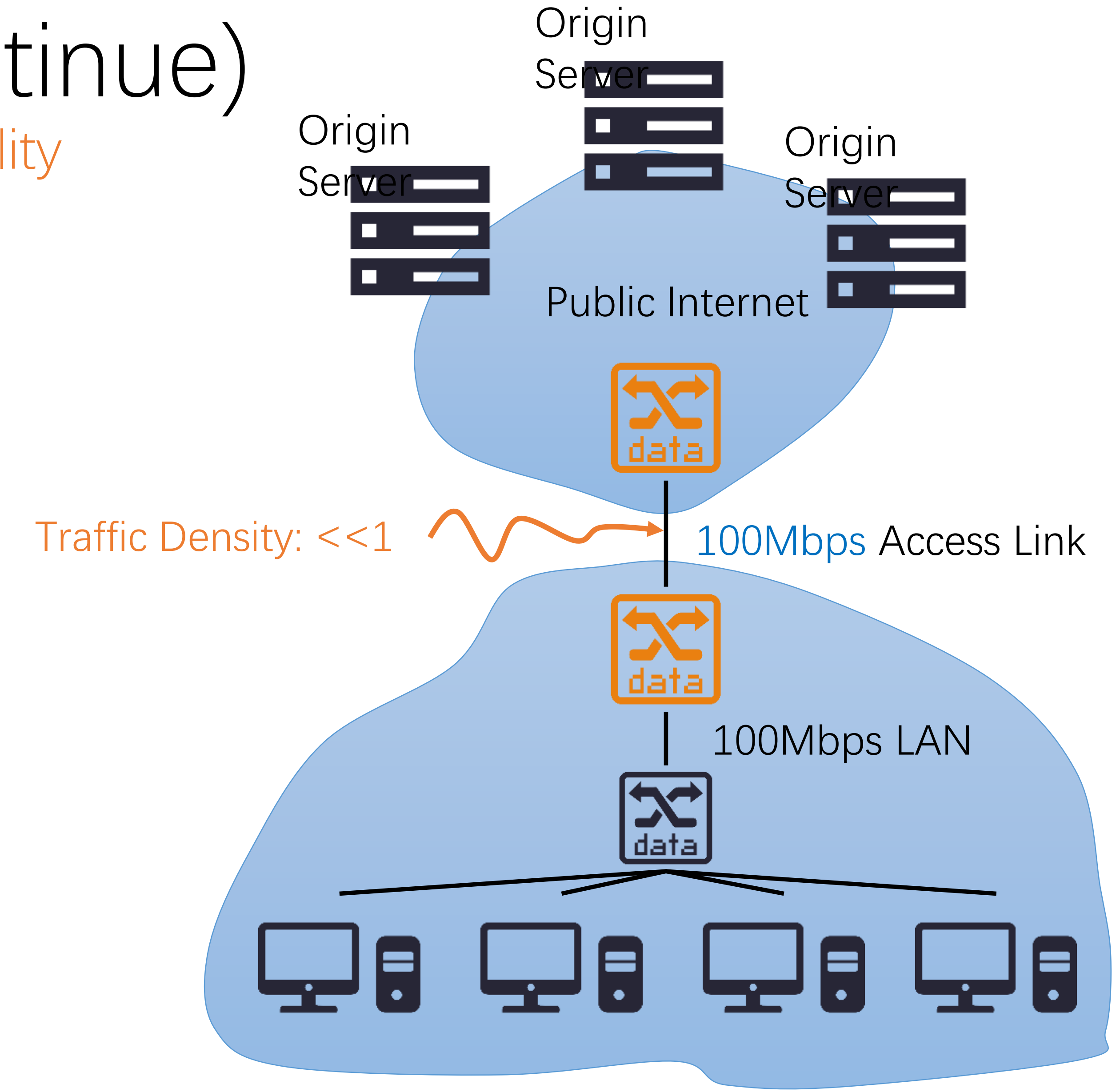
possible solution

- increase bandwidth of access link to, say, 100 Mbps

consequence

- utilization on LAN = 15%
- utilization on access link = 15%
- Total delay = Internet delay + access delay + LAN delay

  = 2 sec + **msecs** + msecs
- often a costly upgrade



Caching example (continue)

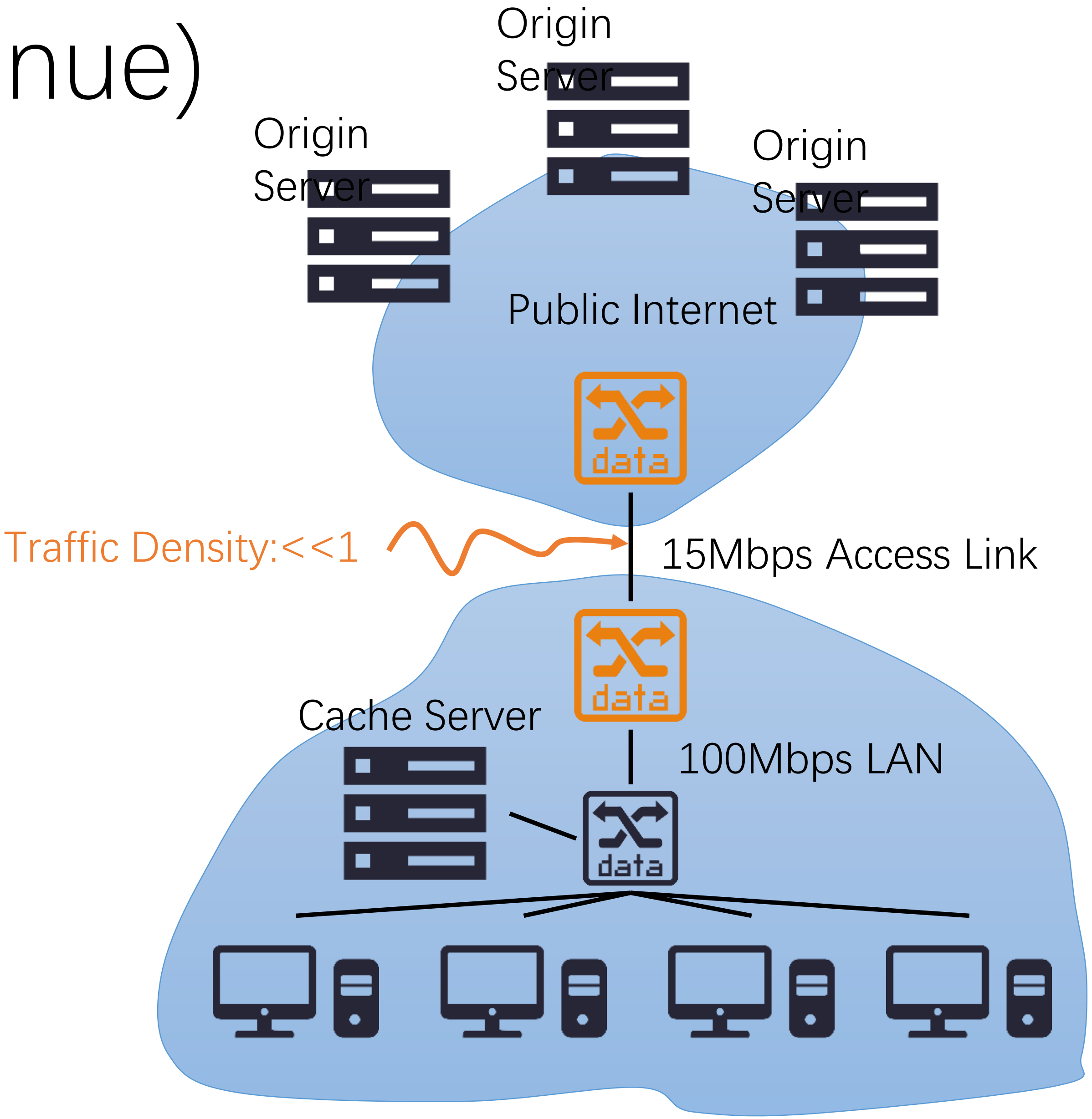
kill the bottleneck by avoiding traffic on it

possible solution: install cache

- suppose hit rate is 0.4

consequence

- 40% requests will be satisfied almost immediately
- 60% requests satisfied by origin server
- utilization of access link reduced to 60%, resulting in negligible delays (say 10 msec)
- total avg delay = Internet delay + access delay + LAN delay = 0.6*(2) + 0.6*0.01+0.4*0.01secs < 1.3 secs



The Conditional GET

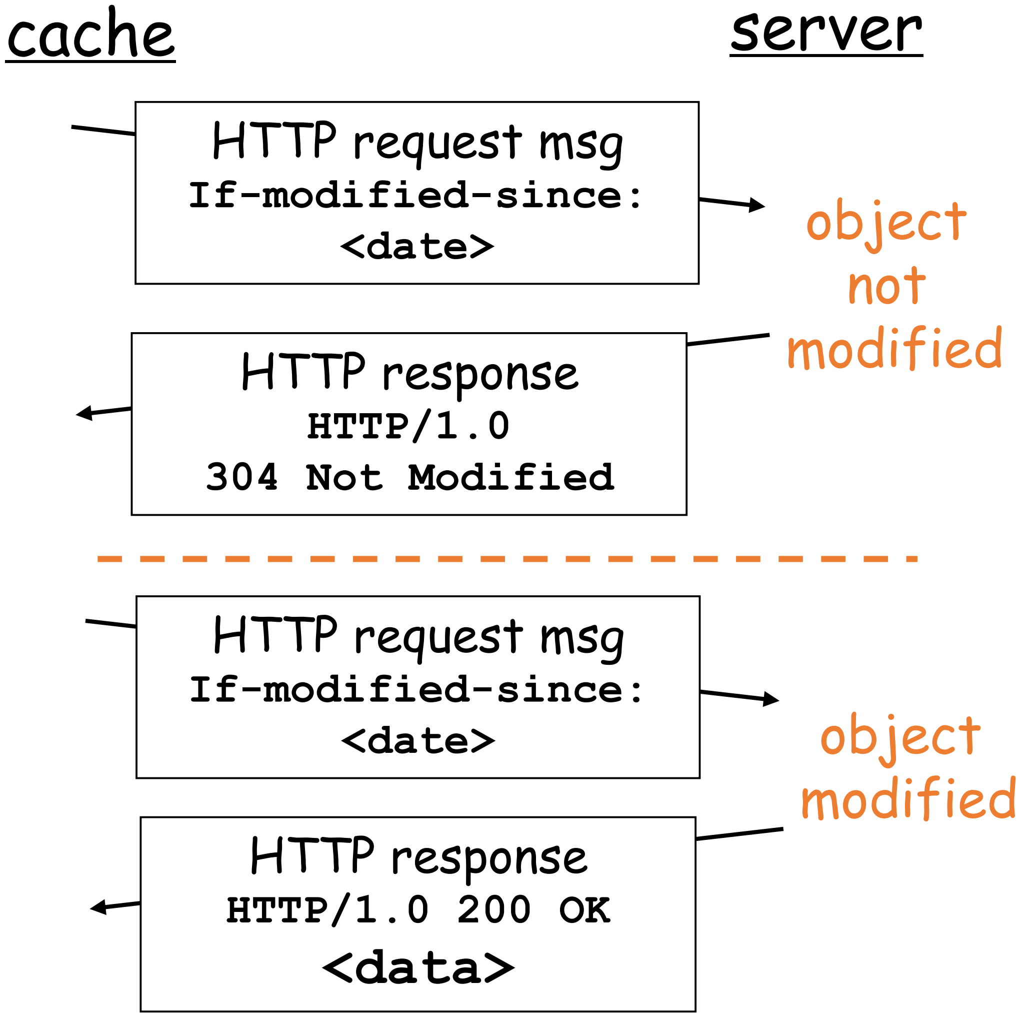
cache at the client

- Goal: don't send object if cache has up-to-date cached version
- cache: specify date of cached copy in HTTP request

```
If-modified-since: <date>
```

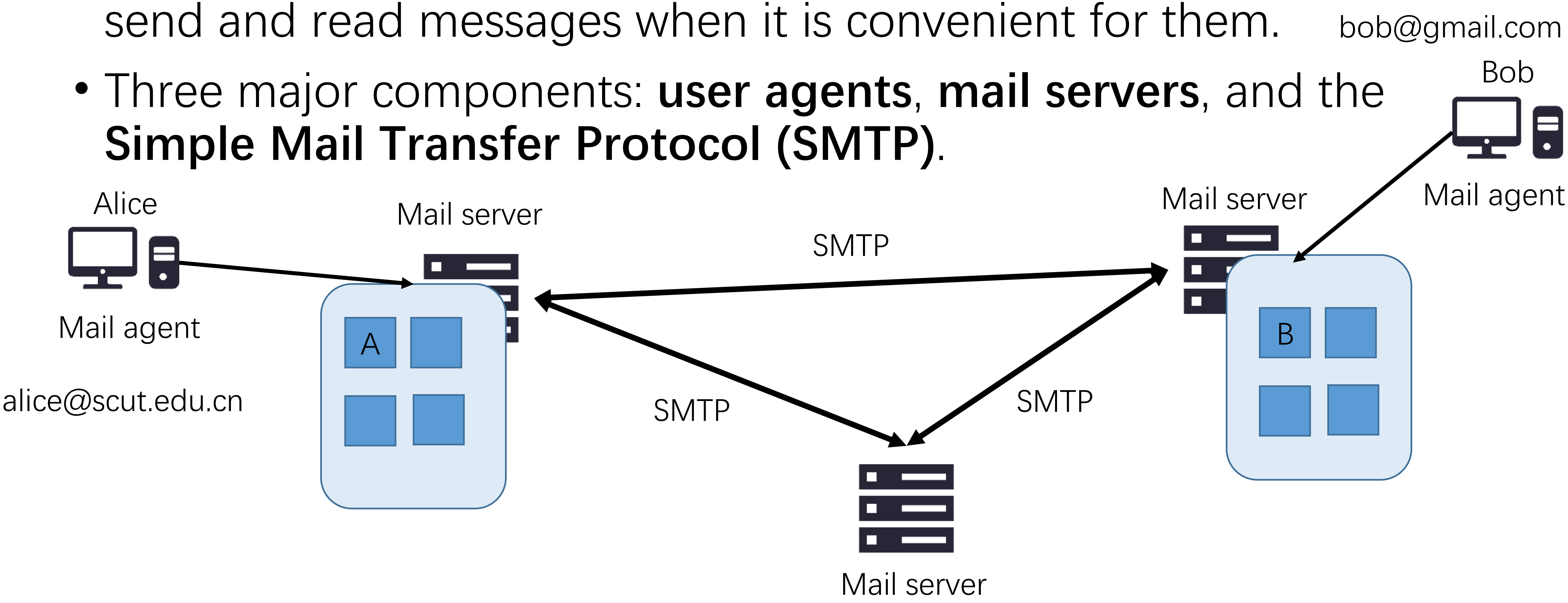
- server: response contains no object if cached copy is up-to-date:

```
HTTP/1.0 304 Not Modified
```



Electronic Mail in the Internet

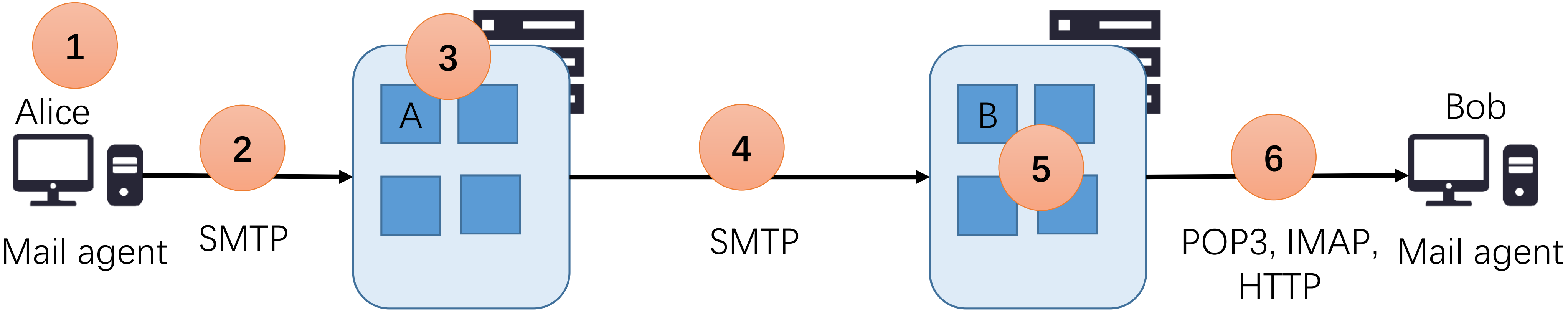
- E-mail is an **asynchronous** communication medium—people send and read messages when it is convenient for them.
- Three major components: **user agents**, **mail servers**, and the **Simple Mail Transfer Protocol (SMTP)**.



SMTP @RFC 5321

- Run on TCP, a push protocol model
- HTTP is pull model
- Mail Message Formats @2.4.3
- POP3 and IMAP @ 2.4.4

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr ... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```



References

- 《图解TCP/IP》与《图解HTTP》(日本人写的)
  - Very good books