

فصل دوم – لایه کاربرد

هدف:

□ درک مفاهیم و جنبه های پیاده سازی

پروتکل های کاربرد شبکه

❖ مدل های سرویسی لایه انتقال

❖ الگوی مشتری / سرویس دهنده

❖ الگوی همتا به همتا

□ یادگیری پروتکل ها با بررسی پروتکل های

معروف

❖ HTTP

❖ FTP

❖ SMTP / POP3 / IMAP

❖ DNS

ایجاد یک کاربرد شبکه

هدف و اصل شبکه - کاربردهای شبکه می باشند که:

- بر روی سیستم های پایانی (مختلف) اجرا شوند.
- از طریق شبکه با هم ارتباط داشته باشند.
- همانند نرم افزارهای سرویس دهنده وب که با نرم افزارهای مرورگر ارتباط برقرار می نمایند.

هیچ نیازی به نوشتن نرم افزار برای دستگاههای هسته شبکه نیست

- دستگاههای هسته شبکه، کاربردهای کاربر را اجرا نمی کنند.

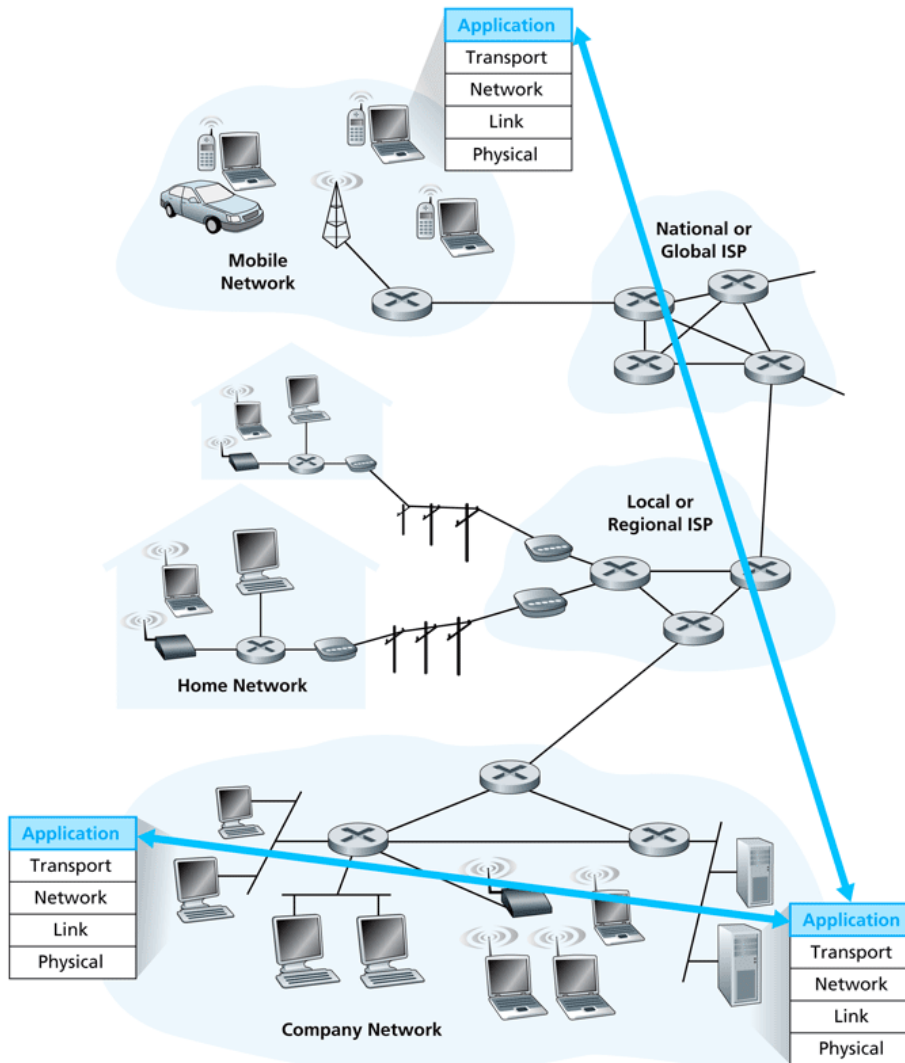


Figure 2.1 ♦ Communication for a network application takes place between end systems at the application layer.

تهیه کننده: فرناد آهنگری

انواع معماری برنامه های کاربردی

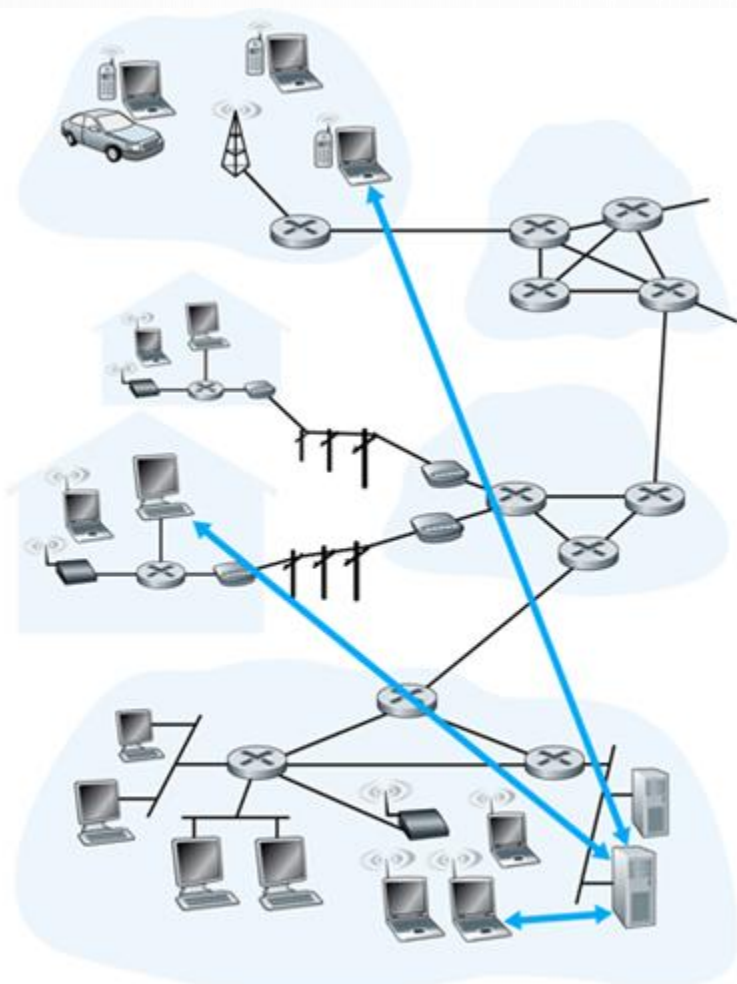
□ انواع معماریها

❖ مشتری / سرویس دهنده (Client / Server)

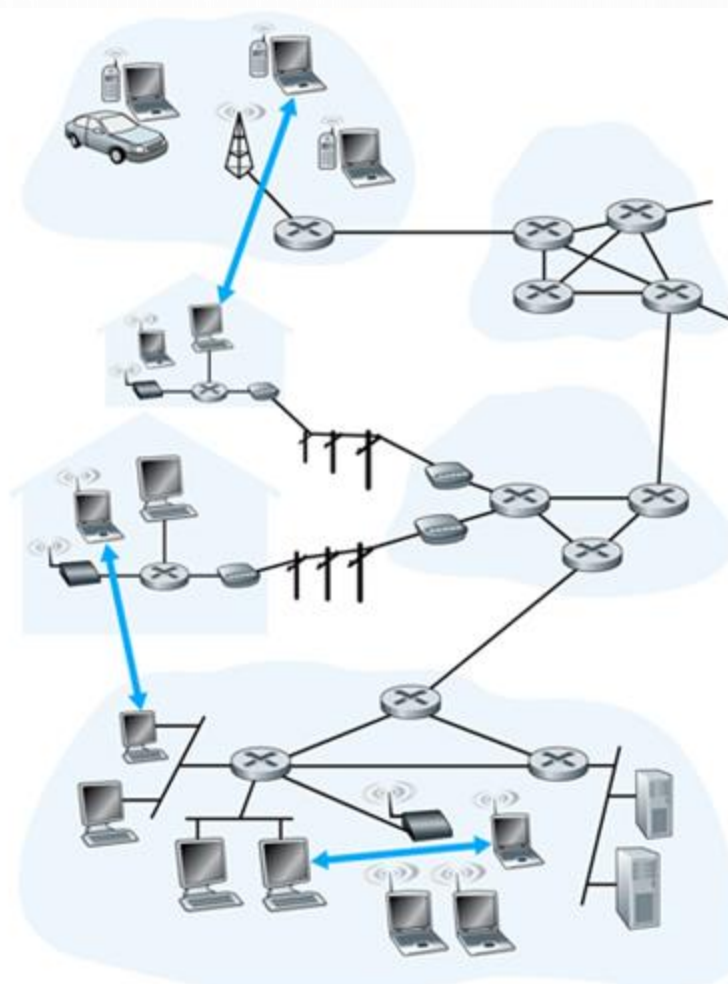
❖ همتا به همتا (Peer-to-Peer)

❖ ترکیبی از C/S و P2P

انواع معماری برنامه های کاربرد

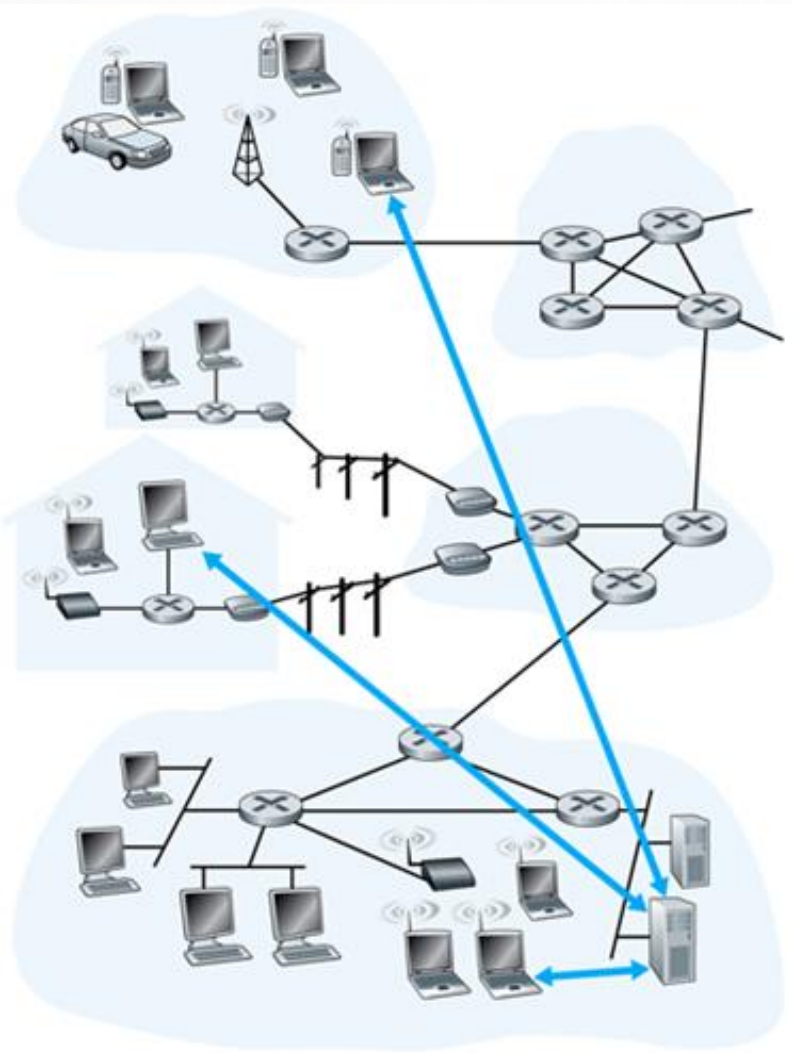


a. Client-server architecture



b. Peer-to-peer architecture

معماری مشتری / سرویس دهنده



a. Client-server architecture

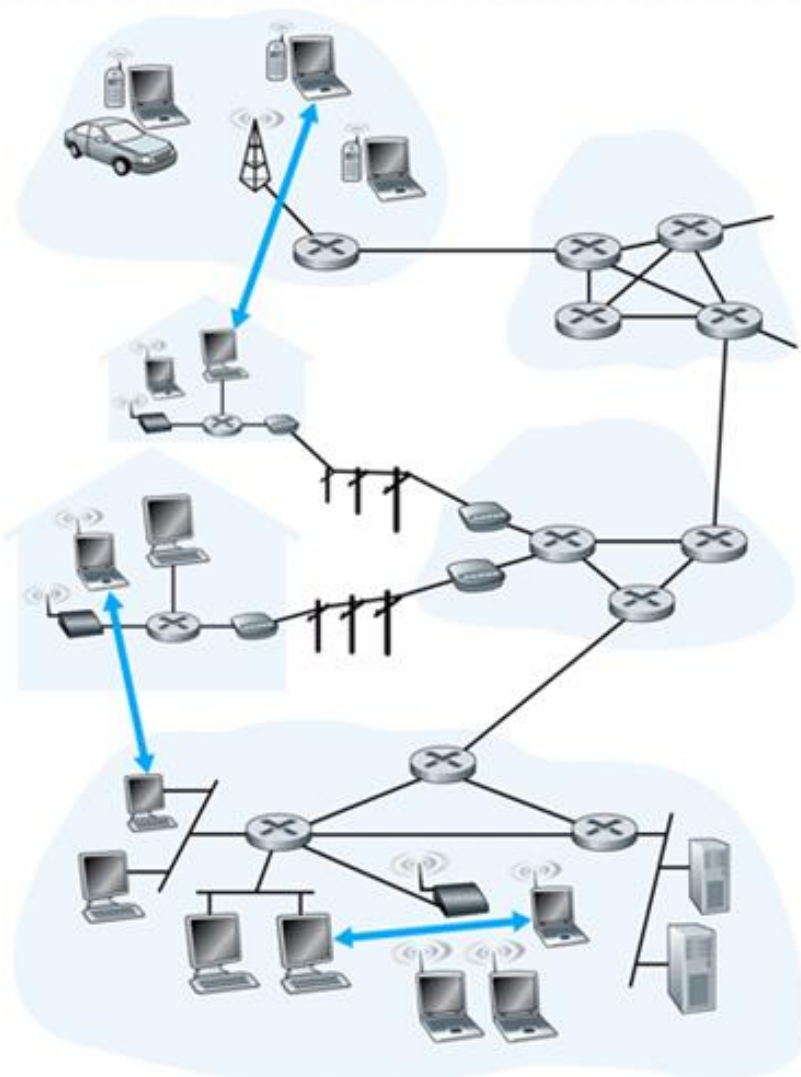
سرویس دهنده

- میزبانی همیشه روشن می باشد
- دارای آدرس IP ثابتی می باشد.
- برای مقیاس پذیر بودن (برای داشتن قابلیت پاسخگویی به سایت های اجتماعی بزرگ) از گروهی سرویس دهنده (SERVER FARMS) استفاده می نماید.

مشتریها

- با سرویس دهنده ارتباط برقرار می نمایند.
- ممکن است بطور متناوب به شبکه وصل شوند.
- ممکن است دارای آدرس های IP دینامیک باشند.
- بطور مستقیم با هم (با مشتریهای دیگر) ارتباط برقرار نمی نمایند (در صورت نیاز، از طریق سرویس دهنده)

معماری همتا به همتا به خالص



b. Peer-to-peer architecture

❑ دارای هیچ سرویس دهنده همواره روشنی نیست.

❑ سیستم های پایانی دلخواه می توانند مستقیماً به همدیگر متصل شوند.

❑ همتا ها می توانند بطور متناوب به شبکه وصل شوند و آدرس IP خود را تغییر دهند.

فوق العاده مقیاس پذیر بوده

اما مدیریت آنها دشوار می باشد.

معماری ترکیبی همتا به همتا و مشتری / سرویس دهنده

اسکایپ (Skype)

- ❖ برنامه همتا به همتای صدا بر روی IP می باشد.
- ❖ دارای سرویس دهنده مرکزی، برای یافتن آدرس طرف مقابل می باشد.
- ❖ ارتباط مشتری با مشتری مستقیم است (نه از طریق سرویس دهنده)

پیام رسانی فوری (Instant Messaging – IM)

- ❖ چت بین دو کاربر بصورت P2P می باشد.
- ❖ استفاده از سرویس دهنده مرکزی ← جهت تعیین حاضر بودن مشتریها و یافتن محل (آدرس) آنها
 - زمانی که کاربری آنلاین می شود، آدرس IP او در سرویس دهنده مرکزی ثبت می شود.
 - کاربران برای یافتن دوستان خود، با سرویس دهنده مرکزی تماس برقرار می نمایند.

ارتباط بین پروسس ها

پروسس مشتری : پروسسی که شروع کننده ارتباط می باشد.

پروسس سرویس دهنده : پروسسی که منتظر تماس می باشد.

❑ توجه : برنامه های با معماری P2P هم دارای پروسس مشتری و هم پروسس سرویس دهنده می باشند.

پروسس : برنامه ای است که بر روی یک میزبان در حال اجرا می باشد.

❑ ارتباط بین پروسس ها در درون یک میزبان، از طریق مکانیزم “ارتباط بین پروسس ها” که توسط سیستم عامل مدیریت می شود، انجام می پذیرد.

❑ ارتباط بین پروسس ها در میزبان های متفاوت، از طریق مکانیزم “مبادله پیام” صورت می پذیرد.

سوکتها (SOCKETS)

❑ سوکت یک رابط نرم افزاری است. (API)

❑ و پروسس ها از آن برای ارسال و دریافت پیام استفاده می نمایند.

❑ سوکت مشابه درب می باشد

❖ برنامه نویس کنترل کاملی بر روی لایه کاربرد (بالای سوکت) دارد

❖ اما کنترل بسیار کمی (در حد تنظیم برخی پارامترها و تعیین نوع پروتکل انتقال) بر روی سمت

پایین سوکت (لایه انتقال) دارد.

سوکتها (SOCKETS)

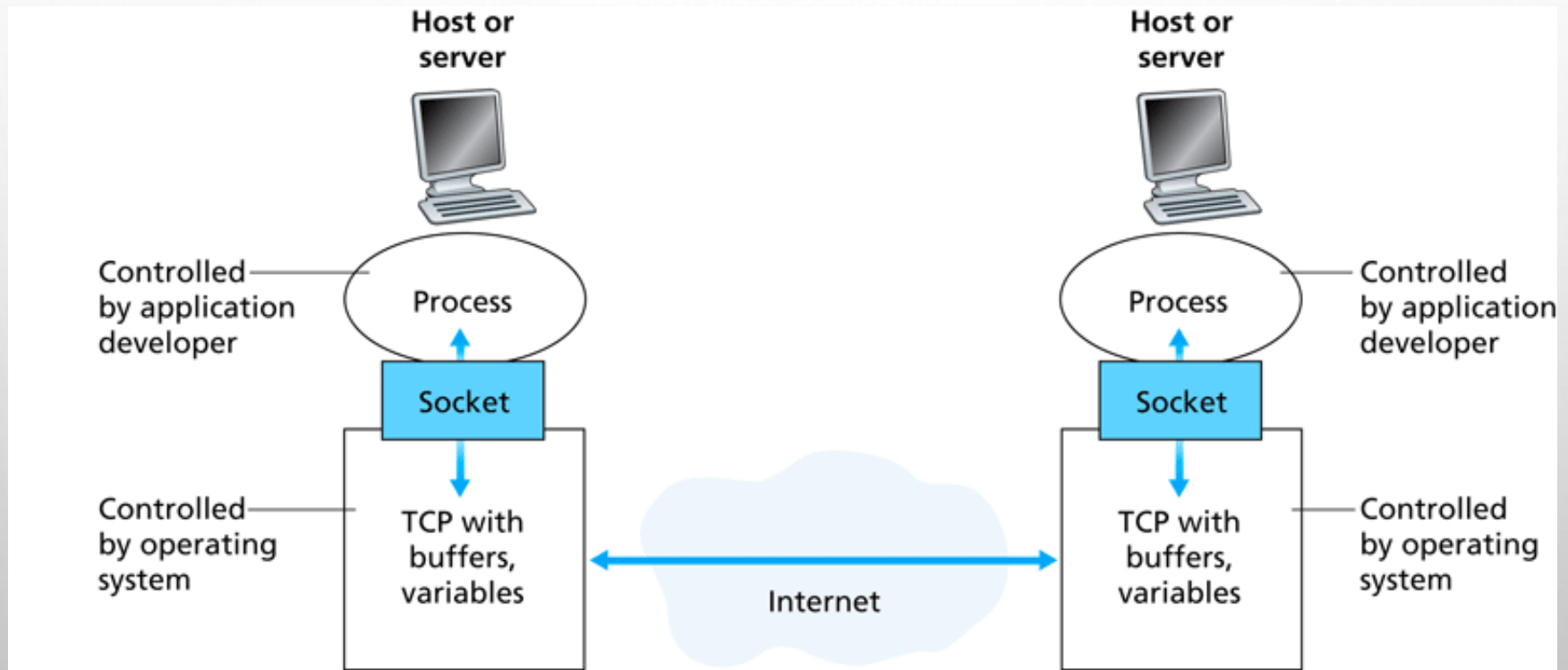


Figure 2.3 ♦ Application processes, sockets, and underlying transport protocol

آدرس دهی پروسی ها

❑ هر پروسی برای دریافت پیام ، لازم است

دارای شناسه منحصر به فرد باشد.

❑ میزبانها از آدرس های منحصر به فرد ۳۲ بیتی

IP استفاده می نمایند.

❑ سوال : آیا داشتن یک آدرس IP برای تعیین

یک پروسی کافی می باشد؟

❖ جواب : خیر، زیرا پروسی های زیادی بر

روی یک میزبان در حال اجرا می باشند.

❑ شناسه بایستی هم شامل آدرس IP و هم

شماره پورت - که به پروسی درون میزبان

اختصاص داده شده- باشد

❑ مثالهایی از شماره پورتهای

❖ سرویس دهنده وب (HTTP) : ۸۰

❖ سرویس دهنده ایمیل : ۲۵

❑ برای ارسال پیام HTTP به سرویس دهنده وبی

مانند `gaia.cs.umass.edu` لازم است:

❖ IP Address : 128.119.245.12

❖ Port Number : 80

پروتکل لایه کاربرد

□ پروتکل های با دامنه عمومی

❖ در RFC ها تعریف می شوند.

❖ امکان همکاری وجود دارد.

❖ همانند HTTP و SMTP

□ پروتکل های اختصاصی

❖ مشخصات آنها در دسترس عموم نمی باشد.

❖ همانند Skype

موارد زیر را تعیین می نماید.

□ نوع پیامهای مبادله شده

❖ درخواستی، پاسخی

□ ساختار و ترکیب پیام

❖ چه فیلدهایی در پیام موجود است

❖ هر فیلد چه معنایی دارد.

❖ قوانینی برای پروسس که چه زمانی و چگونه به

پیام ها پاسخ دهد.

یک کاربرد به چه سرویسهای انتقالی نیاز دارد؟

توان عملیاتی (Throughput)

❑ برخی برنامه ها (مانند چند رسانه ای) برای “کارا بودن”، به یک حداقلی از توان عملیاتی نیاز دارند.

❑ کاربردهای دیگر (کاربردهای الاستیک) می توانند با هر توان عملیاتی موجود کار کنند.

امنیت (Security)

❑ در بعضی از مواقع نیاز است که داده ها رمز شود، محرمانه بودن، جامعیت داده ها و ...

مسئله از دست دادن داده ها (DATA LOSS)

❑ برخی برنامه ها (مانند صدا) می توانند کمی از دست دادن را تحمل نمایند.

❑ کاربردهای دیگر (مانند انتقال فایل، TELNET) نیاز به انتقال داده ۱۰۰٪ مطمئن دارند.

مسئله زمان بندی (TIMING)

❑ برخی کاربردها (مانند تلفن اینترنتی، بازیهای تعاملی) برای کارا بودن نیاز به تاخیر پایین دارند.

سرویسهای انتقالی مورد نیاز کاربردهای معمول

Application	Data Loss	Bandwidth	Time-Sensitive
File transfer	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps—1 Mbps Video: 10 kbps—5 Mbps	Yes: 100s of msec
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

Figure 2.4 ♦ Requirements of selected network applications

سرویسهای پروتکل‌های انتقال در اینترنت

سرویس UDP

- ❑ انتقال داده غیر مطمئن را بین پروسسهای مشتری و سرویس دهنده برقرار می نماید.
- ❑ پشتیبانی نمی کند: برقراری اتصال، انتقال مطمئن، کنترل جریان، کنترل ازدحام، زمانبندی، توان عملیاتی تضمین شده و امنیت

سرویس TCP

- ❑ اتصالگرا (CONNECTED-ORIENTED): بین پروسسهای مشتری و سرویس دهنده اتصالی برقرار می شود.
- ❑ انتقال مطمئن (RELIABLE TRANSFER) بین پروسسهای فرستنده و گیرنده
- ❑ کنترل جریان (FLOW CONTROL): فرستنده نمی تواند گیرنده را با ارسال سریع داده ها سرریز نماید.
- ❑ کنترل ازدحام (CONGESTION CONTROL): زمانیکه شبکه دارای ترافیک بالا می باشد، سرعت ارسال فرستنده ها را کنترل و تنظیم می نماید.
- ❑ پشتیبانی نمی کند: زمانبندی، حداقلی از توان عملیاتی، امنیت

کاربردهای اینترنتی: پروتکل های لایه کاربرد و لایه انتقال

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 2821]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube), RTP	TCP or UDP
Internet telephony	SIP, RTP, or proprietary (e.g., Skype)	Typically UDP

Figure 2.5 ♦ Popular Internet applications, their application-layer protocols, and their underlying transport protocols

WEB AND HTTP

برخی اصطلاحات

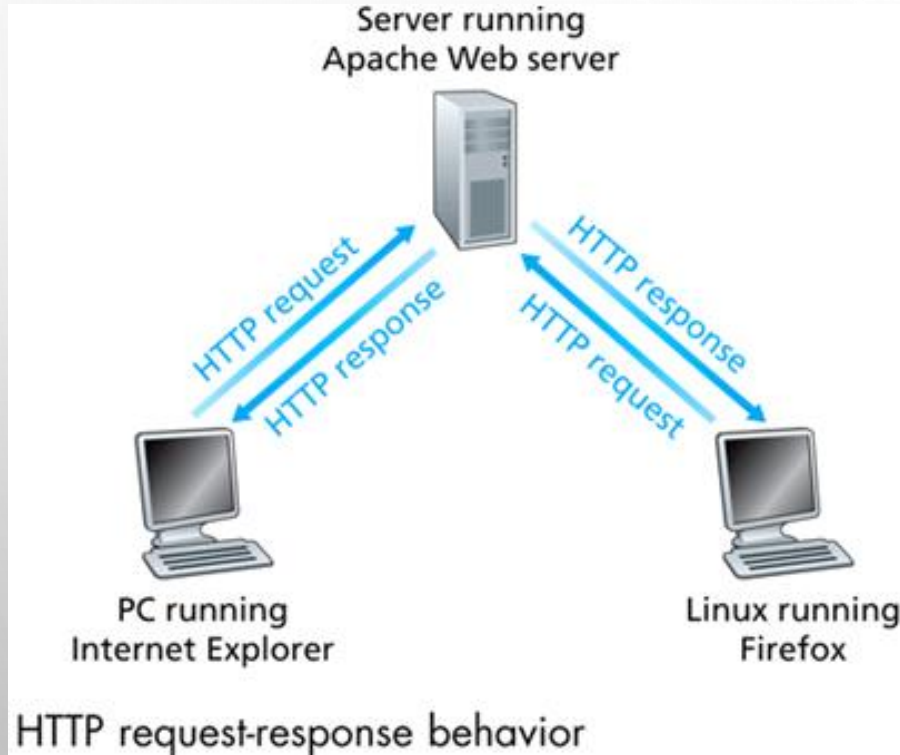
- ☐ صفحه وب (Web Page) شامل تعدادی شی می باشد.
- ☐ اشیاء می توانند فایل HTML، تصویر JPEG، اپلت های جاوا، فایل های صدا و ... باشند.
- ☐ هر صفحه وب شامل یک فایل HTML پایه می باشد، که می تواند ارجاعاتی به اشیاء دیگر داشته باشد.
- ☐ هر شی توسط یک URL آدرس پذیر می باشد.
- ☐ مثالی برای URL:

`www.someschool.edu/someDept/pic.gif`

host name

path name

مرور کلی بر HTTP



HTTP: HYPERTEXT TRANSFER PROTOCOL

□ یک پروتکل لایه کاربرد وب می باشد.

□ بر اساس مدل مشتری / سرویس دهنده می باشد

❖ مشتری : مرورگر که اشیاء وب را درخواست،

دریافت و نمایش می دهد.

❖ سرویس دهنده : سرویس دهنده وب است، که

اشیاء را در پاسخ به درخواست ها، ارسال می

دارد.

مرور کلی بر HTTP (ادامه)

HTTP بصورت بدون نگهداری وضعیت می

باشد. (Stateless)

- سرویس دهنده هیچ اطلاعاتی از درخواست قبلی مشتری را نگهداری نمی نماید.

موضوع جداگانه

پروتکل‌هایی که وضعیت را نگهداری می نمایند، بسیار پیچیده می باشند.

- اگر ارتباط مشتری / سرویس دهنده ای قطع شود، می توانند با استفاده از اطلاعات "وضعیت"، اقدام به بازسازی ارتباط از نقطه قطع شده بنمایند.

از TCP استفاده می نماید:

- مشتری یک ارتباط TCP را با پورت ۸۰ سرویس دهنده برقرار می نماید (با ایجاد سوکت)
- سرویس دهنده این ارتباط از طرف مشتری را می پذیرد.

- پیامهای HTTP (پیامهای پروتکل لایه کاربرد) بین مرورگر (بعنوان مشتری HTTP) و سرویس دهنده وب (بعنوان سرویس دهنده HTTP) مبادله می شود.

- ارتباط TCP بسته می شود.

ارتباطات HTTP

HTTP دائم

❑ در هر ارتباط TCP که بین مشتری و سرویس

دهنده برقرار می شود، می توان چندین شی را ارسال نمود.

❑ بعد از ارسال یک شی از طرف سرویس دهنده،

ارتباط همچنان توسط سرویس دهنده برای پاسخ به تقاضاهای دیگر، باز خواهند ماند.

HTTP غیر دائم

❑ در هر ارتباط TCP، حداکثر می توان یک شی را ارسال نمود.

❑ بعد از ارسال یک شی از طرف سرویس دهنده، ارتباط توسط سرویس دهنده قطع می شود.

Nonpersistent HTTP (cont.)

فرض کنید که کاربر آدرس زیر را وارد می نماید: (این فایل شامل ارجاع به ۱۰ تصویر JPEG می باشد)


`WWW.SOMESCHOOL.EDU/SOMEDEPARTMENT/HOME.INDEX`

-
- 1 a. HTTP client initiates TCP connection to HTTP server (process) at `www.someSchool.edu` on port 80
 - 1 b. 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 `someDepartment/home.index`
 3. HTTP server receives request message, forms *response message* containing requested object, and sends message into its socket

time

NONPERSISTENT HTTP (CONT.)

4. HTTP server closes 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



time

HTTP غیر دائم : زمان پاسخ

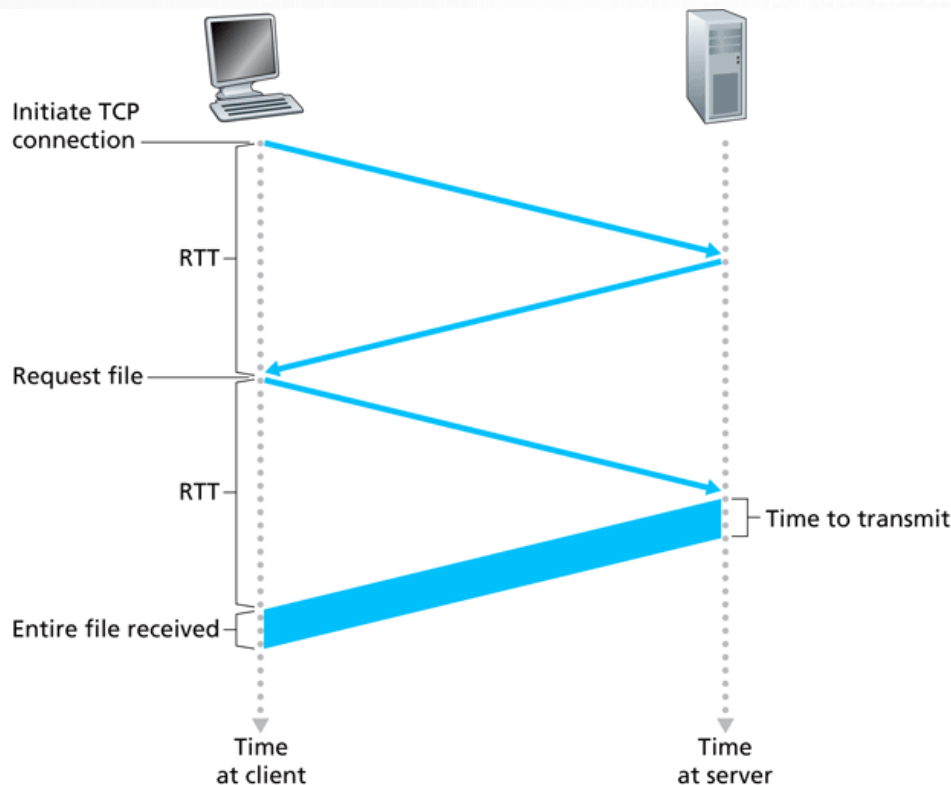


Figure 2.7 ♦ Back-of-the-envelope calculation for the time needed to request and receive an HTML file

تعریف RTT: زمان لازم برای اینکه یک بسته کوچک از مشتری به سرویس دهنده برود و برگردد.

زمان پاسخ شامل :

□ یک RTT برای برقراری ارتباط TCP

□ یک RTT برای "HTTP درخواست" و چندین

بایت اول از "HTTP پاسخ" برای برگشت.

□ زمان ارسال فایل

$$\text{total} = 2\text{RTT} + \text{transmit time}$$

HTTP دائم

پیامدهای HTTP غیر دائم

□ برای هر شی، نیاز به $2 * RTT$ می باشد.

□ به ازای هر ارتباط TCP، سرباری را به سیستم عامل وارد می نماید.

□ برای جبران کاهش سرعت (بدلیل سری بودن عملیات درخواست و پاسخ) ، مرورگرها از ارتباطات چندگانه موازی با سرویس دهنده استفاده می نمایند.

HTTP دائم

□ سرویس دهنده بعد از ارسال پاسخ، ارتباط را همچنان باز نگه میدارد.

□ پیامهای بعدی، از طریق همان ارتباط اولیه ارسال می شوند.

□ مشتری بلافاصله بعد از برخورد با یک شی جدید، درخواستی را ارسال می دارد.

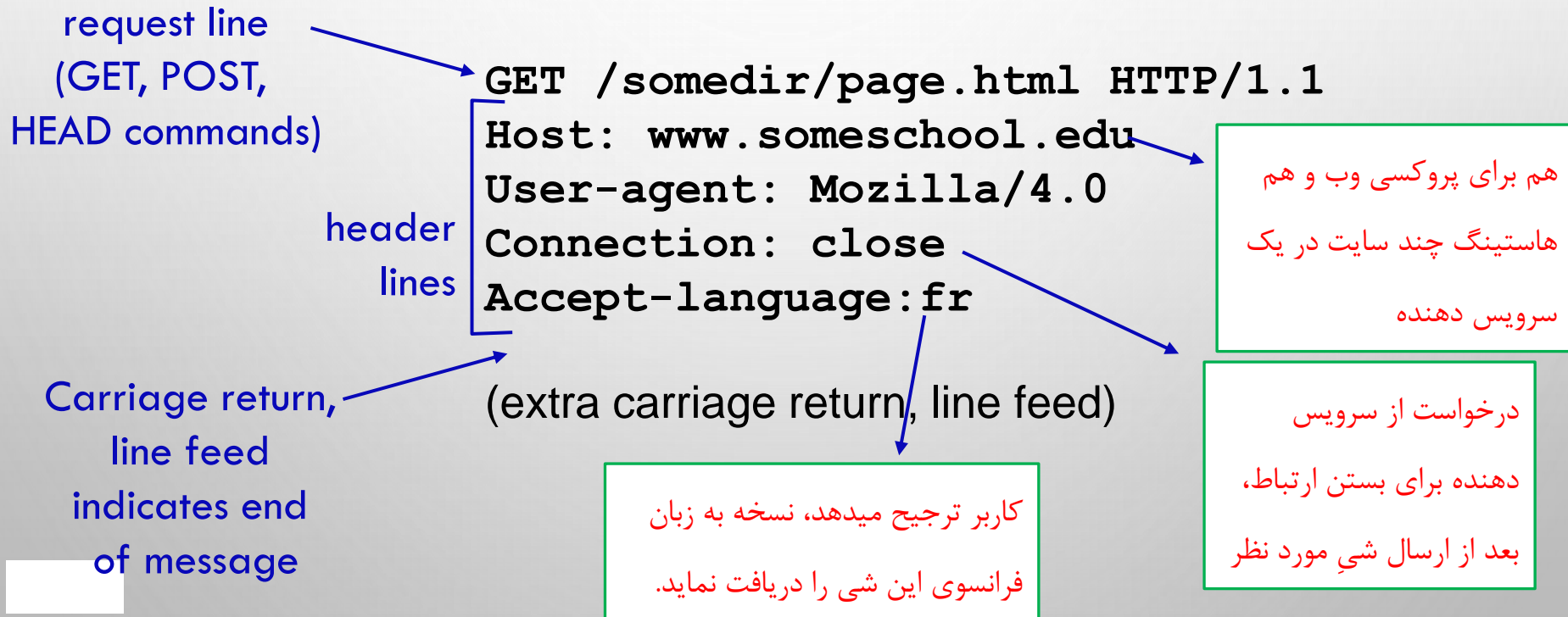
HTTP request message [RFC 2616]

پیام HTTP درخواست

□ دو نوع پیام HTTP موجود می باشد: درخواست و پاسخ (REQUEST, RESPONSE)

HTTP REQUEST MESSAGE □

❖ بصورت اسکی (با فرمت قابل فهم برای انسان) می باشد.



وضعیت (حالت) کاربر – سرویس دهنده : کوکی ها

بسیاری از سایت های مهم از کوکی ها استفاده می نمایند.

شامل چهار مولفه می باشد:

- ☐ یک خط مربوط به کوکی در پیام HTTP پاسخ
- ☐ یک خط مربوط به کوکی در پیام HTTP درخواست
- ☐ فایل کوکی بر روی میزبان مشتری، که توسط مرورگر مدیریت می شود.
- ☐ پایگاه داده ای در سمت سرویس دهنده

HTTP پروتکلی بدون نگهداری حالت می باشد.

- ☐ باعث مقیاس پذیر شدن سرویس دهنده می شود.
- ☐ اما به هر حال سرویس دهنده نیاز به شناسایی کاربر و محدود کردن دسترسی او دارد.
- ☐ برای اینکار از کوکی ها می توان استفاده نمود.

کوکی ها : نگهداری حالت (ادامه)

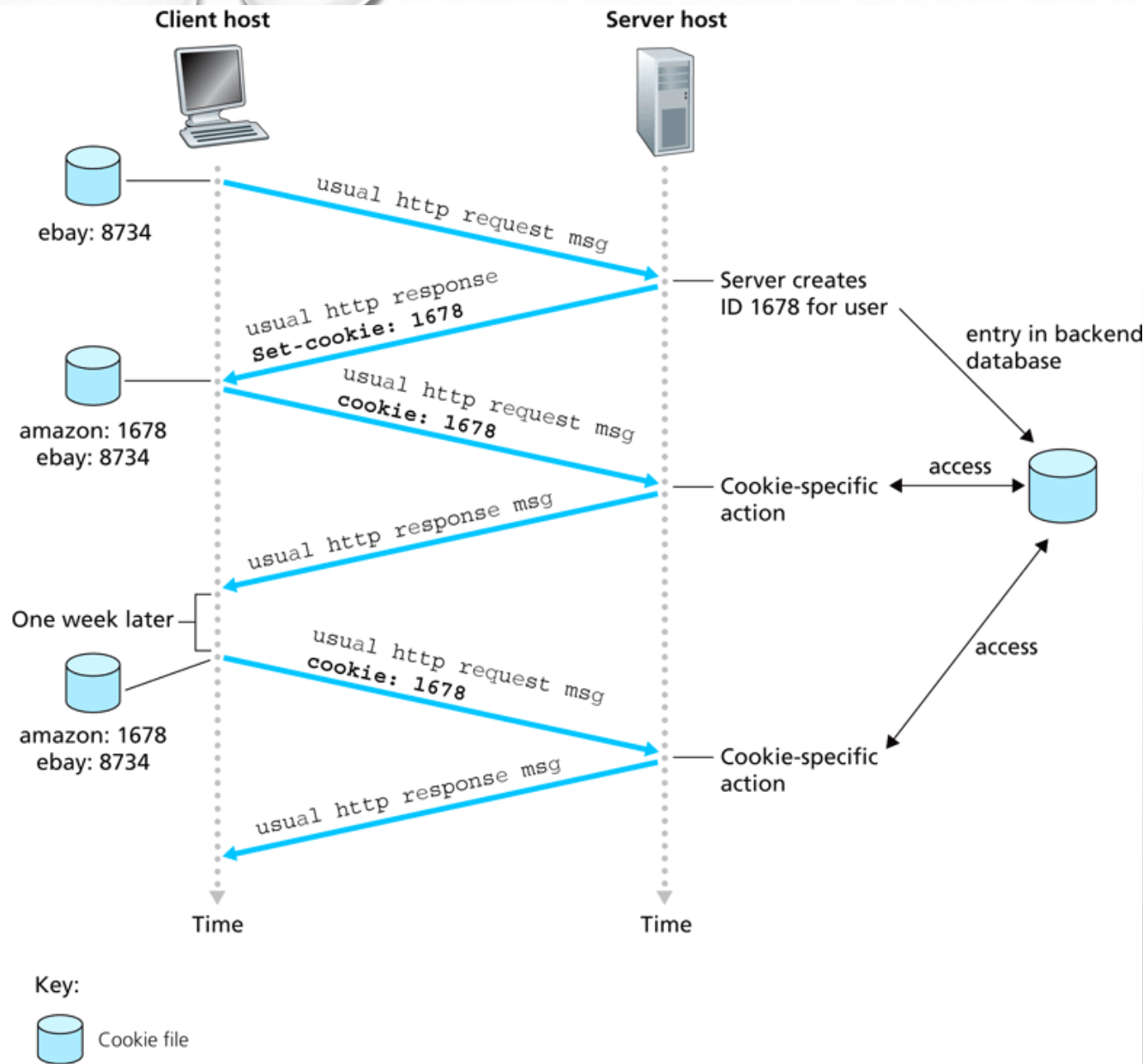


Figure 2.10 ♦ Keeping user state with cookies

کوکی ها (ادامه)

کاربرد کوکی ها

کوکی ها و حریم خصوصی

- ❑ کوکی ها به سایت اجازه مقداری یادگیری درباره ما می دهند.
- ❑ شما ممکن است در سایتهایی ایمیل و نام خود را وارد نمایید.

❑ اجازه دادن (Authorization)

❑ کارتهای خرید (Shopping Cards)

❑ نگهداری وضعیت جلسه کاربر (User Session

State) جهت ایمیل های وبی – مانند یاهو، جی

میل و ...

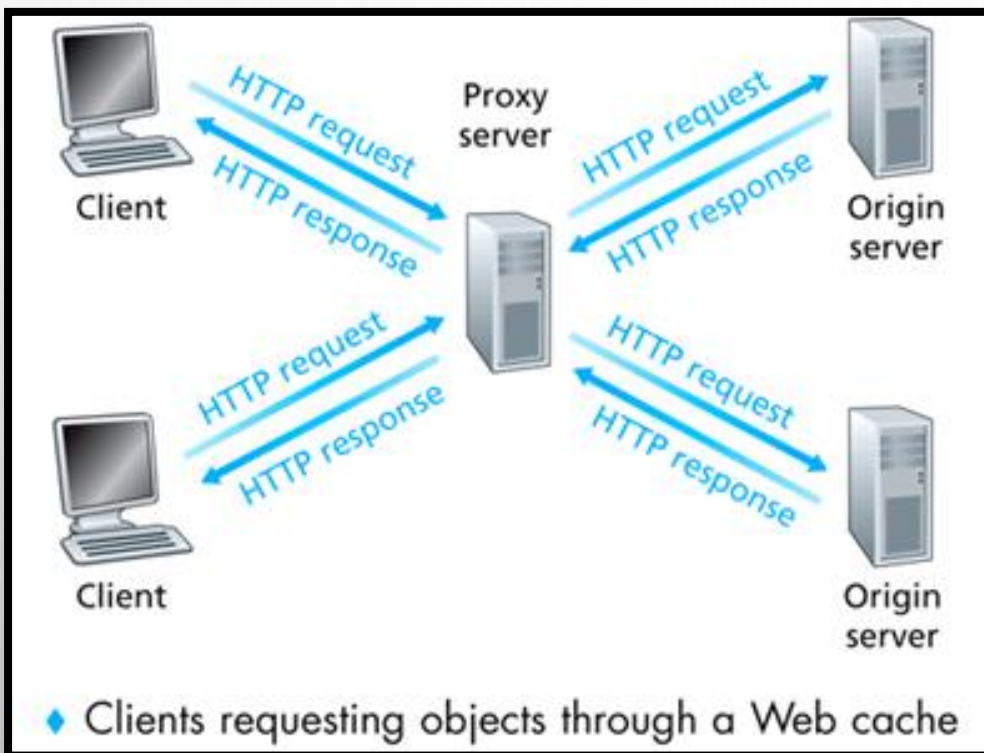
نحوه نگهداری "وضعیت"

- ❑ توسط پروتکل های پایانی نقطه-به-نقطه، وضعیت در فرستنده/گیرنده بر روی تراکنش ها نگهداری می شود.
- ❑ کوکی ها : پیامهای http حمل کننده وضعیت می باشند.

WEB CACHES

نهانگاههای وب

هدف : ارسال پاسخ به مشتریان، بدون درگیر کردن سرویس دهنده اصلی



❑ کاربر مرورگر خود را برای دسترسی به وب، به پروکسی تنظیم می نماید.

❑ مرورگر تمام درخواستهای خود را به سمت پروکسی ارسال می دارد.

❖ اشیایی که در نهانگاه باشند، پروکسی آنها را برمی گرداند.

❖ در غیر اینصورت، پروکسی اشیاء را از سرویس دهنده اصلی درخواست کرده و سپس آنها را به مشتری برمی گرداند.

نشانگاههای وب (سرورهای پروکسی) WEB CACHES (PROXY SERVER)

دلایل استفاده از نشانگاه وب

- ❑ کاهش زمان پاسخ برای درخواستهای مشتریان
- ❑ کاهش ترافیک بر روی لینک دسترسی

❑ پروکسی هم بعنوان مشتری و هم سرویس

دهنده عمل می نماید.

❑ معمولاً پروکسی بوسیله ISP (دانشگاه، شرکت و

یا ISP های محلی نصب می شوند)

مثالی از WEB CACHING

فرضیات

❑ متوسط اندازه اشیاء = ۱۰۰۰۰۰ بیت

❑ متوسط نرخ درخواستها از طرف مرورگرهای

دانشگاه به سرویس دهنده اصلی = ۱۵۰ در ثانیه

❑ تاخیر زمانی از مسیر یاب دانشگاه به هر سرویس

دهنده اصلی و برگشت به مسیر یاب = ۲ ثانیه

نتایج

❑ Utilization on LAN = 15%

❑ Utilization on access link = 100%

❑ total delay = Internet delay +
access delay + LAN delay

❑ = 2 sec + minutes + milliseconds

Origin servers

Public Internet

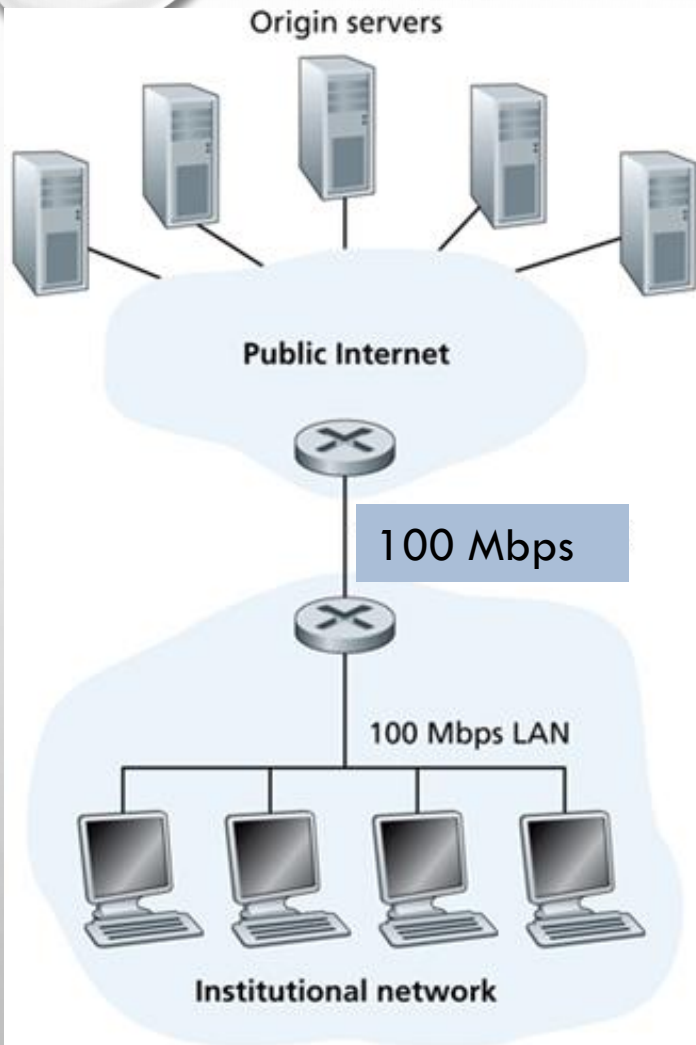
15 Mbps access link

100 Mbps LAN

Institutional network

Bottleneck between an institutional
network and the Internet

مثالی از WEB CACHING (ادامه)



◆ Bottleneck between an institutional network and the Internet

یک جواب ممکن

❑ ارتقاء پهنای باند لینک دسترسی، مثلاً به

100Mbps

نتایج

❑ utilization on LAN = 15%

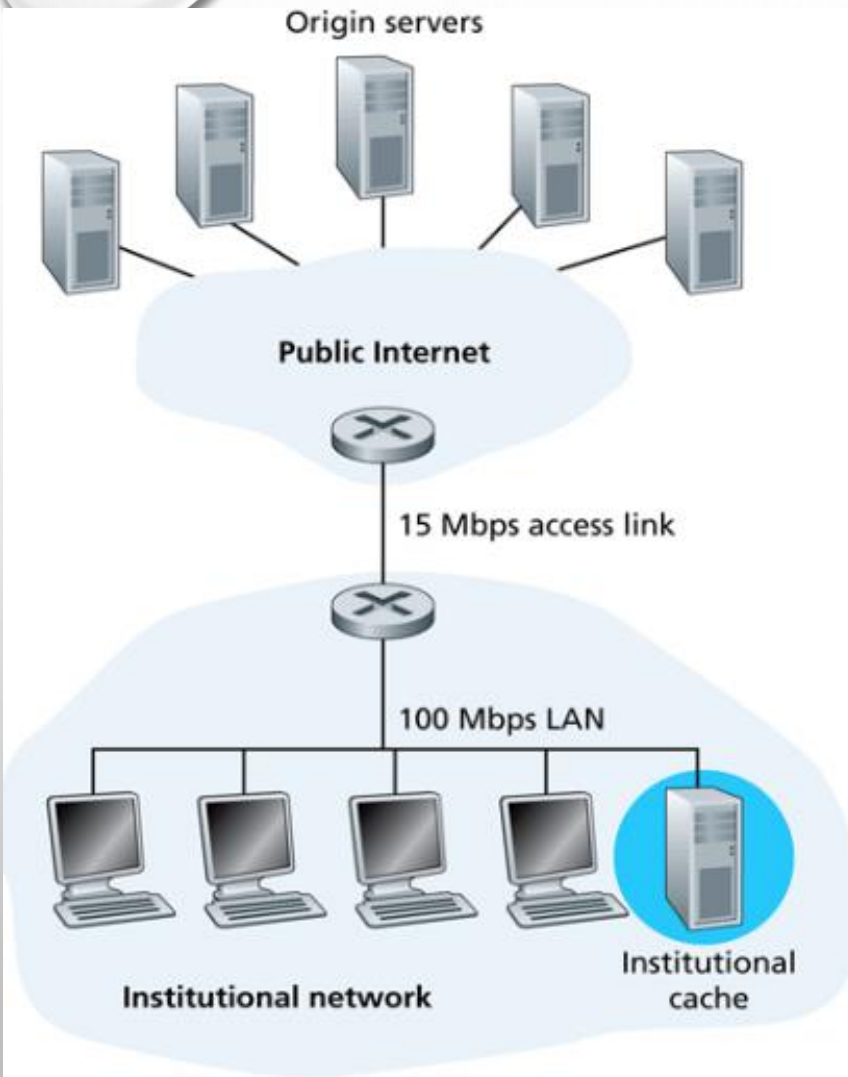
❑ utilization on access link = 15%

❑ Total delay = Internet delay +
access delay + LAN delay

❑ = 2 sec + msec + msec

❑ often a costly upgrade

مثالی از WEB CACHING (ادامه)



Adding a cache to the institutional network

یک جواب ممکن دیگر: نصب نهانگاه وب

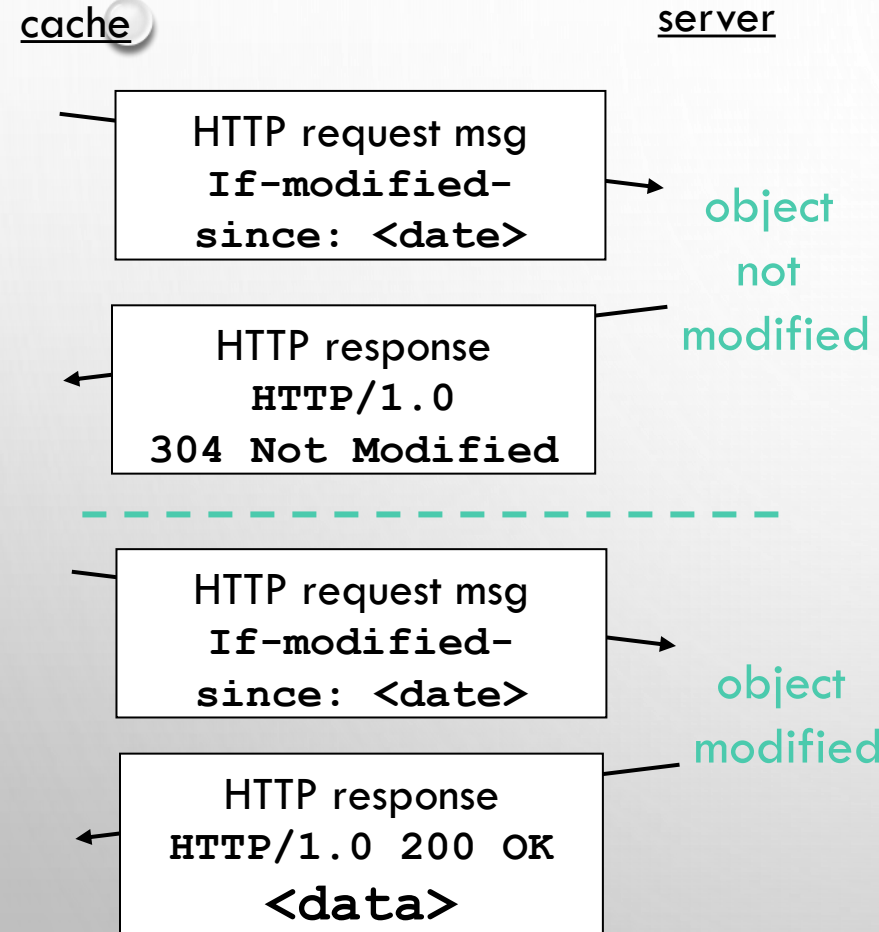
❑ فرض کنید نرخ برخورد ۰.۴ باشد.

❑ نتایج

- ❑ 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 = $.6 * (2.01)$ secs + $.4 * \text{milliseconds} < 1.4$ secs

CONDITIONAL GET

دریافت شرطی



هدف: اگر شی مورد نظر در نهانگاه می باشد، نیازی به

ارسال آن از طرف سرویس دهنده نمی باشد

❑ توسط نهانگاه: تاریخ کپی شی درون نهان در دستور درخواست ذکر می شود:

❑ If-modified-since: <date>

❑ سرویس دهنده: اگر کپی شی درون نهانگاه به روز می باشد، هیچ داده ای را ارسال نمی دارد.

❑ HTTP/1.0 304 Not Modified

FTP: پروتکل انتقال فایل (FILE TRANSFER PROTOCOL)

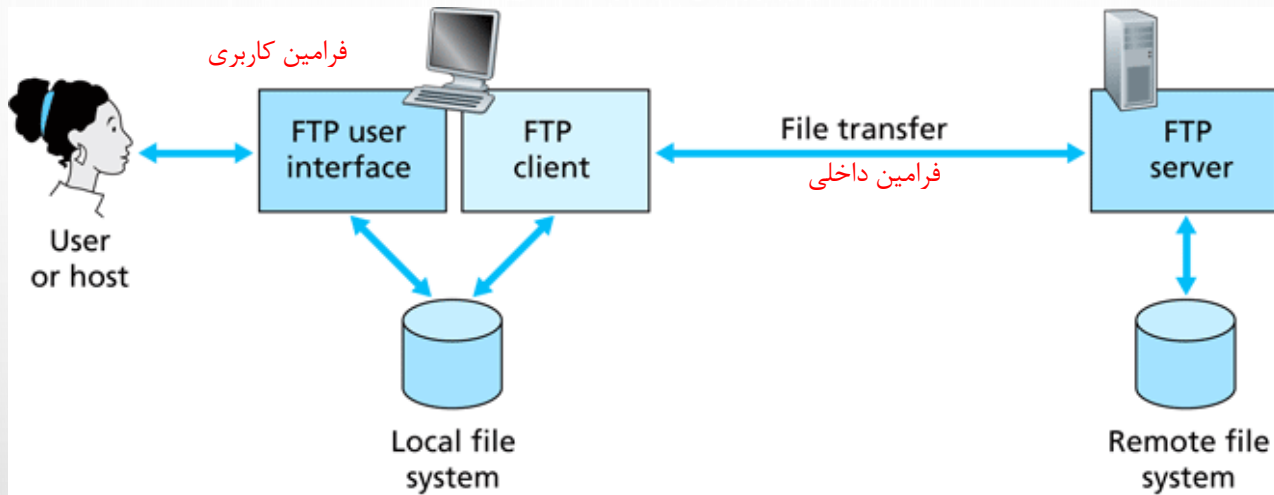
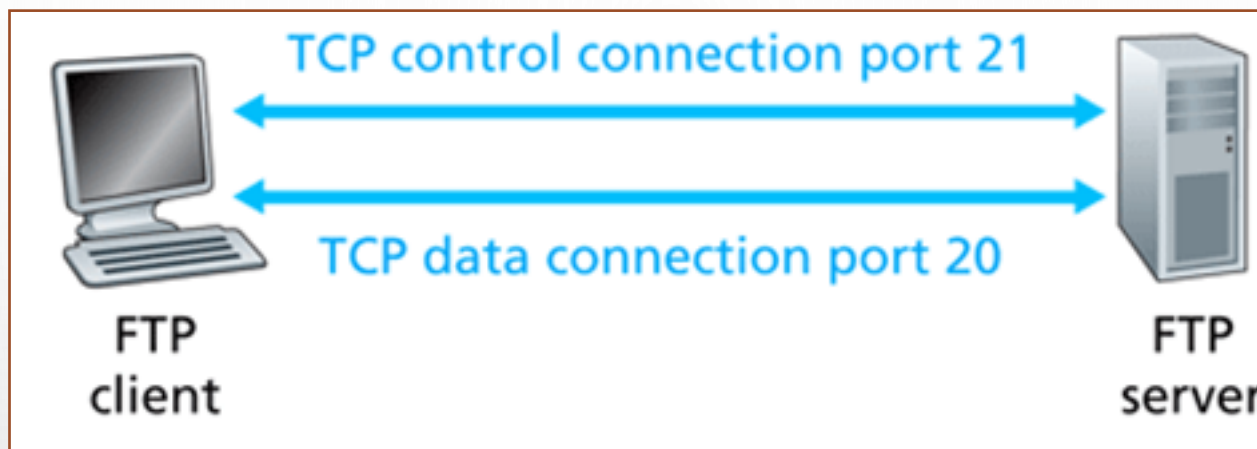


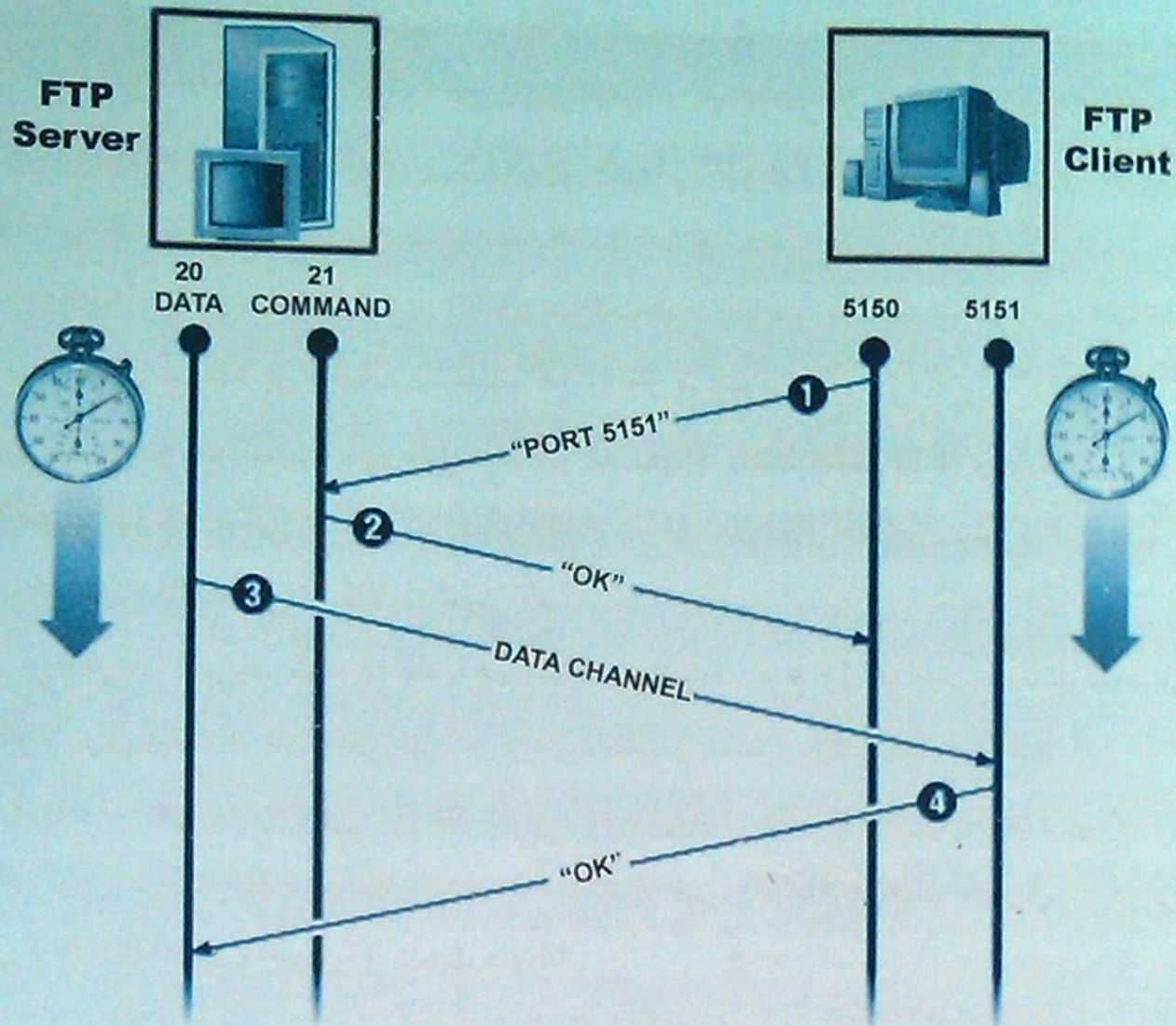
Figure 2.14 ♦ FTP moves files between local and remote file systems

- ❑ انتقال فایل به/ از سیستم راه دور
- ❑ استفاده از مدل مشتری / سرویس دهنده
- ❖ مشتری : سمتی است که شروع کننده انتقال است (یا به سیستم راه دور یا از آن)
- ❖ سرویس دهنده : سیستم راه دور
- ❑ تعریف شده در RFC 959
- ❑ شماره پورت مورد استفاده : ۲۱
- ❑ هر فرمان کاربری به یک فرمان داخلی نگاشت می شود

FTP: ارتباطات مجزای کنترل و داده

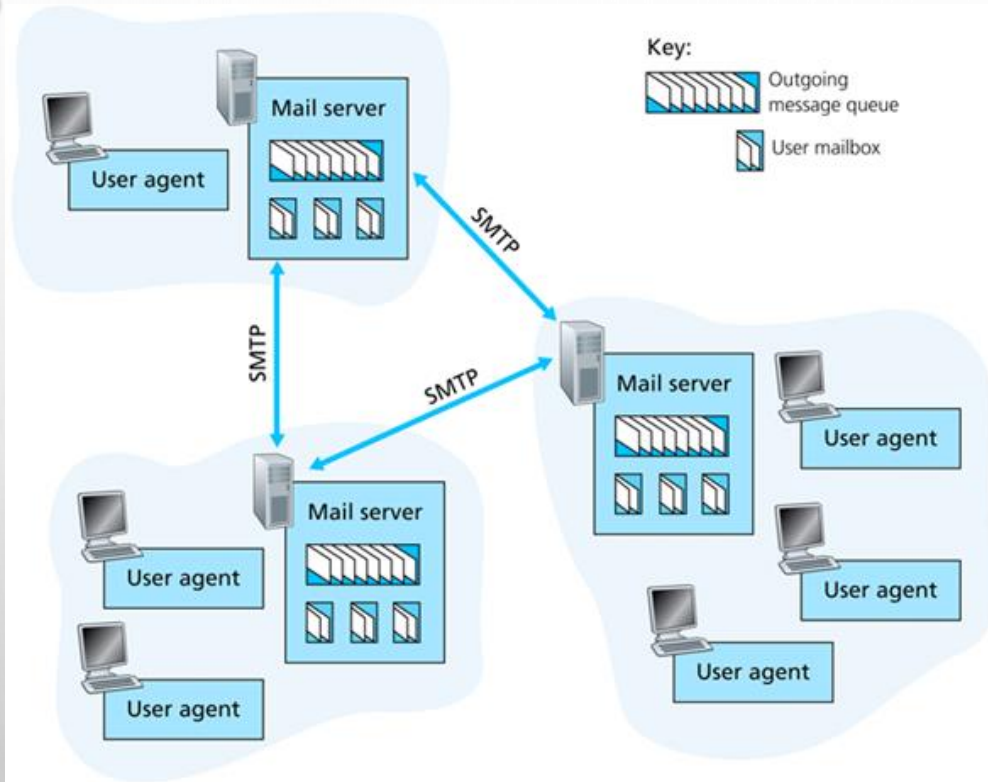


- ☐ مشتری با پورت شماره ۲۱ سرویس دهنده یک ارتباط TCP برقرار می نماید.
- ☐ مشتری از طریق این پورت کنترلی، اعتبار سنجی می شود (از طریق نام کاربری و رمز عبور)
- ☐ مشتری اقدام به مشاهده لیست فایل‌های سرویس دهنده می نماید (با ارسال فرمان مربوطه روی خط کنترلی)
- ☐ زمانی که سرویس دهنده فرمان انتقال فایل را دریافت می نماید، ارتباط TCP دومی را با مشتری باز می کند!
- ☐ بعد از انتقال یک فایل، سرویس دهنده ارتباط TCP مربوط به داده را می بندد (ارتباط کنترلی برقرار است)
- ☐ برای انتقال فایلی دیگر، سرویس دهنده ارتباط TCP داده ای دیگری را باز می نماید.
- ☐ سرویس دهنده FTP، "حالت" مربوط به دایرکتوری جاری و اعتبار سنجی قبلی را نگهداری می نماید.



شکل ۸-۵ مثالی از یک نشست FTP به روش معمولی

نامه الکترونیکی



❑ از زمانهای اولیه اینترنت وجود داشته است.

❑ در آن دوره ها از معمولترین و عمومی ترین سرویه های اینترنتی بود.

❑ یکی از روشهای انتقال آسنکرون می باشد.

❑ ایمیل های امروزی بسیار قدرتمند تر شده اند.

❖ با استفاده از لیست های پستی، می توان هزاران ایمیل را ارسال نمود.

❖ پیامهای ایمیل می توانند شامل ضمیمه، لینک

ها، متون فرمت دهی شده (بصورت HTML)،

38 عکس و ... باشد.

ELECTRONIC MAIL

سه مولفه مهم :

❑ برنامه کاربر (USER AGENT)

❑ سرورهای ایمیل

❑ پروتکل انتقال ایمیل ساده (SMTP)

برنامه کاربر

❑ بطور ساده ایمیل خوان

❑ با قابلیت ایجاد، ویرایش و ارسال پیامها

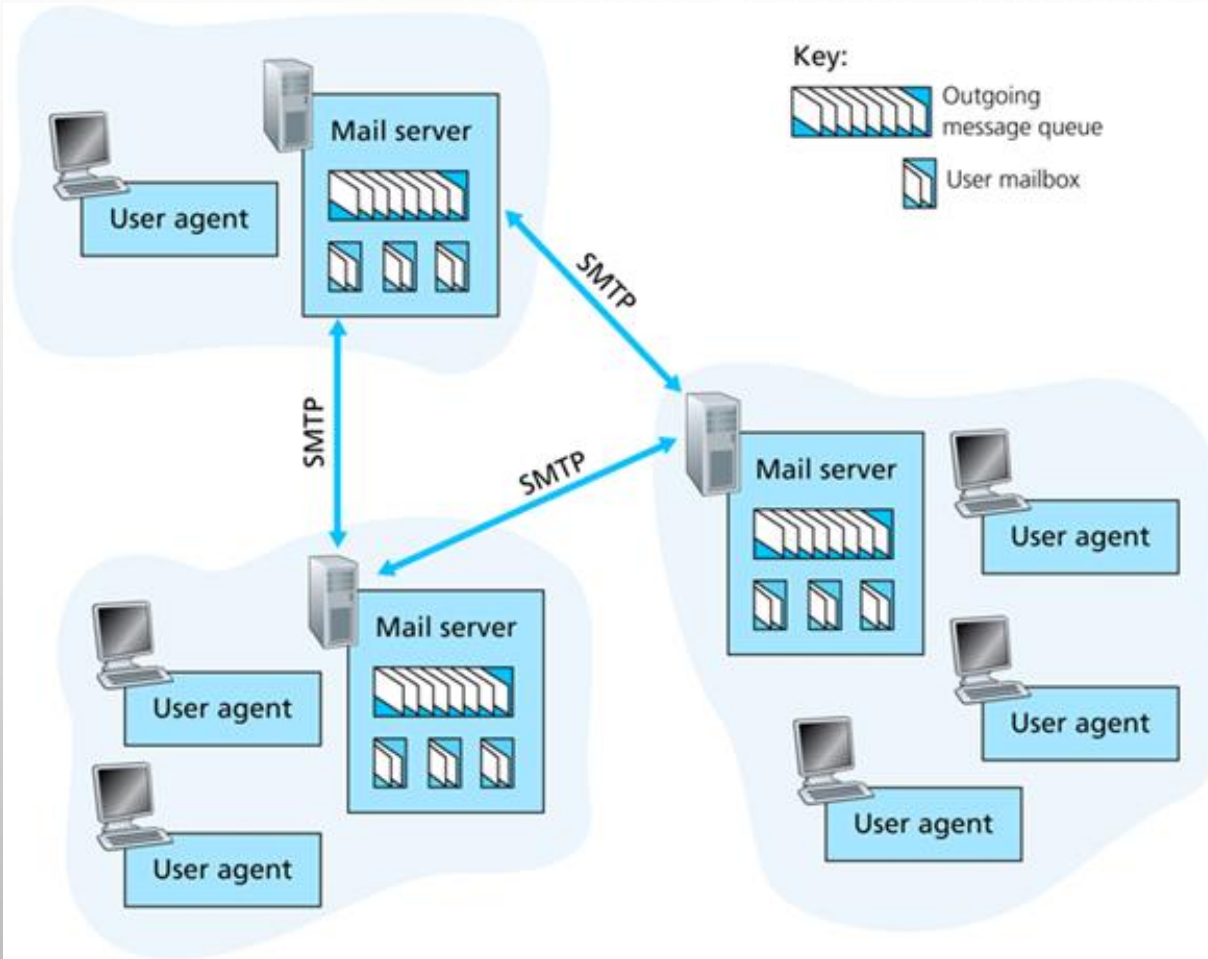
❑ بطور مثال برنامه های EUDORA,

OUTLOOK, ELM, MOZILLA

THUNDERBIRD

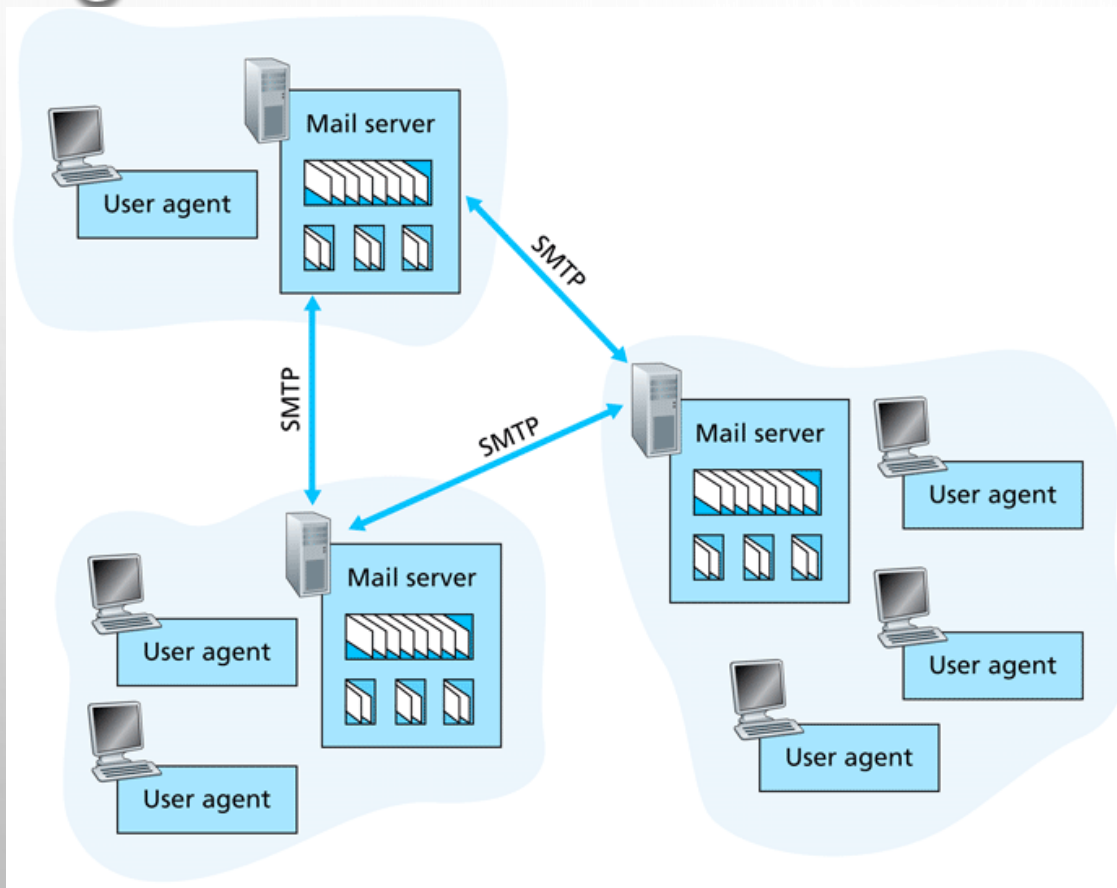
❑ پیامهای خروجی و ورودی در سرور ایمیل

ذخیره می شوند



ELECTRONIC MAIL: MAIL SERVERS

سرورهای ایمیل:



❑ صندوق پستی شامل پیامهای ورودی کاربران

❑ صف پیامهای خروجی برای ارسال

❑ در پروتکل SMTP که بین سرورهای ایمیل برای ارسال پیامهای ایمیل عمل می کند، شامل دو وضعیت است:

❖ مشتری : سروری است که ارسال کننده ایمیل است

❖ سروری : سروری که دریافت کننده ایمیل است

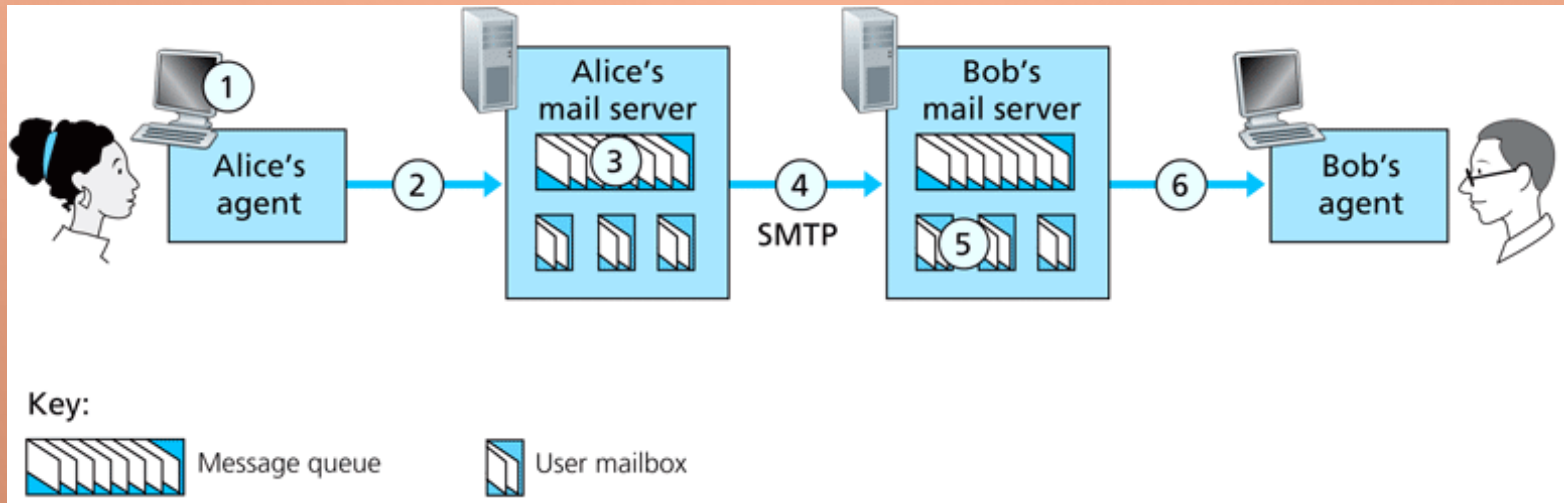
ELECTRONIC MAIL: SMTP [RFC 2821]

- ❑ از پروتکل TCP برای انتقال مطمئن پیامهای ایمیل از سمت مشتری با پورت ۲۵ سرور استفاده می نماید.
- ❑ انتقال پیام ها بصورت مستقیم بین سرور ارسال کننده و سرور گیرنده می باشد.
- ❑ عملیات در سه مرحله انجام می شود:
 - ❖ هماهنگی ارتباط (دست تکانی)
 - ❖ انتقال پیامهای ایمیل
 - ❖ بستن ارتباط
- ❑ تعاملات بصورت دستور (از سمت مشتری) و پاسخ (از طرف سرور) می باشد.
 - ❖ دستورات : به قالب اسکی می باشند.
 - ❖ پاسخ ها : شامل کدهای وضعیت و عبارات می باشند.

SCENARIO: ALICE SENDS MESSAGE TO BOB

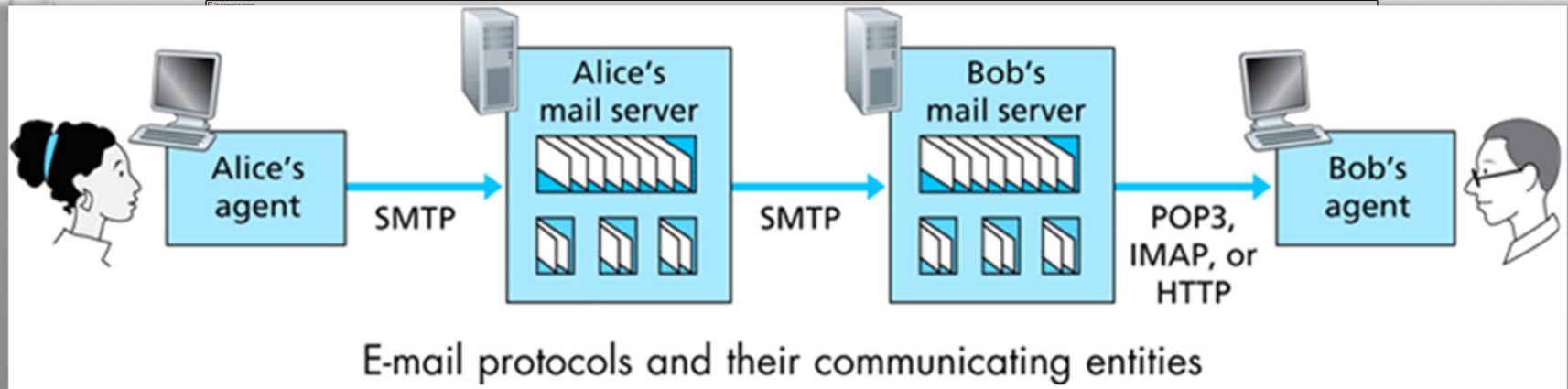
سناریو: ارسال پیام از طرف آلیس به باب

- (1) آلیس با استفاده از برنامه کاربری خود، یک پیام ساخته و به آدرس bob@someschool.edu تنظیم می کند.
- (2) آلیس این پیام را به سرور ایمیل خود ارسال می نماید؛ پیام در صف خروجی قرار می گیرد.
- (3) بخش سمت مشتری SMTP، یک ارتباط TCP با سرور ایمیل باب برقرار می کند.
- (4) بخش سمت مشتری، پیام آلیس را از طریق ارتباط TCP ارسال می دارد.
- (5) سرور ایمیل باب، این پیام دریافتی را در صندوق پستی باب قرار می دهد.
- (6) باب بعداً با کمک برنامه کاربری خود این ایمیل را دریافت کرده و می خواند.



MAIL ACCESS PROTOCOLS

پروتکل‌های دسترسی به ایمیل



❑ SMTP: DELIVER

❑ MAIL ACCESS

❖ POP: PC

- Alice
- Alice

❖ IMAP: INTERNET MAIL ACCESS PROTOCOL [RFC 1730]

تهیه کننده: فرناد آهنگری

- MORE FEATURES (MORE COMPLEX)
- MANIPULATION OF STORED MSGS ON SERVER

❑ SMTP: پروتکلی برای تحویل و ذخیره سازی

نامه به سرویس دهنده (مقصد) می باشد. (و نه تحویل به کاربر)

❑ پروتکل‌های دسترسی: جهت دریافت ایمیل از

سرویس دهنده (مقصد) و تحویل به کاربر می باشند.

POP3 (MORE) AND IMAP

IMAP

❑ POP3 از روش دانلود و حذف استفاده می

نماید.

❑ اگر باب (کاربر مقصد) کامپیوتر کاری خود

را عوض نماید، دیگر نمی تواند، مجددا

ایمیل های خود را بخواند

❑ تمام پیامها را در یک محل (سرویس

دهنده) نگهداری می نماید.

❑ اجازه ایجاد فولدرهای مختلف و سازماندهی

ایمیل ها درون آنها را می دهد.

❑ امکان نگهداری "حالت کاربر" برای جلسات

مختلف را فراهم می آورد.

❑ امکان دسترسی فقط به بخشی از ایمیل را

هم به کاربر میدهد.

DNS : DOMAIN NAME SYSTEM

سیستم نام حوزه (دامنه)

❑ افراد دارای شناسه های مختلفی می باشند:

❖ کد ملی، نام، شماره پاسپورت و ...

❑ میزبانهای اینترنت و مسیریابها نیز:

❖ آدرسهای IP (۳۲ بیتی) - برای آدرسی دهی دیتاگرامها

❖ "نام" همانند www.yahoo.com - مورد استفاده برای انسانها

❑ سوال: چگونه بین آدرسهای IP و نام ها، نگاشت (ترجمه، تبدیل) برقرار می شود.

DNS : DOMAIN NAME SYSTEM

سیستم نام حوزه (دامنه)

❑ پایگاه داده توزیع شده - که بصورت **سلسله مراتبی** در تعدادی سرویس دهنده نام پیاده سازی شده است.

❑ پروتکلی در لایه کاربرد می باشد.

که اجازه می دهد میزبانها و مسیر یابها از پایگاه داده توزیع شده سوال نمایند.

(برای **تحلیل نام** : تبدیل آدرس به نام)

❑ سرویس دهنده های نام معمولاً ماشین های یونیکس هستند که نرم افزار

BIND (Berkeley Internet Domain Name) را اجرا می کنند.

❑ DNS با معماری مشتری / سرویس دهنده بوده و از **پروتکل UDP** و **پورت شماره 53** استفاده می

نماید.

❑ **DNS** همانند **پروتکل های دیگر لایه کاربرد نیست**، و کاربر مستقیماً با آن سروکار ندارد، بلکه پروتکل های

دیگر (نظیر HTTP, FTP,...) از آن استفاده می نمایند.

DNS

سرویس های DNS

❑ تبدیل نام میزبان به آدرس IP

❑ اسامی مستعار میزبانها (host aliasing)

❖ نام های رسمی و نامهای مستعار

❑ نامهای مستعار سرویس دهنده ایمیل

❑ توزیع بار

❖ در سرویس دهنده های وب های تکراری :

استفاده از مجموعه ای از آدرس های IP به ازای

ترجمه یک نام،

چرا DNS سیستمی متمرکز نیست؟

❑ نقطه خرابی تنها (Single Point of Failure)

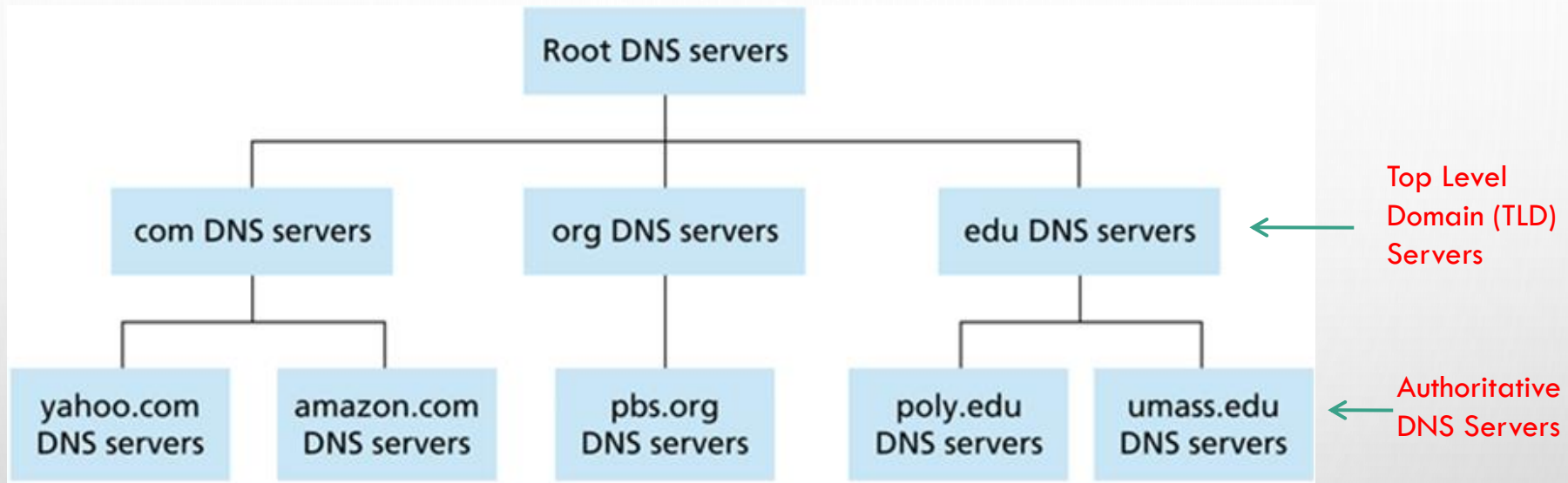
❑ ایجاد بار ترافیکی بالا روی آن

❑ دشواری نگهداری (سیستمی به این بزرگی!)

❑ و در نهایت مقیاس پذیر نخواهد شد.

DISTRIBUTED, HIERARCHICAL DATABASE

پایگاه داده سلسه مراتبی توزیع شده



مشتری آدرس IP مربوط به www.amazon.com را می خواهد : تقریب اول

❑ مشتری از سرویس دهنده ریشه، برای یافتن سرویس دهنده نام **com** درخواست می نماید.

❑ مشتری از سرویس دهنده نام **com** برای یافتن سرویس دهنده نام **amazon.com** درخواست می نماید

❑ مشتری از سرویس دهنده نام **amazon.com** برای بدست آوردن آدرس IP مربوط به **www.amazon.com**

تهیه کننده: فرناد آهنگری

48 درخواست می نماید.

DNS: ROOT NAME SERVERS

سرویس دهنده های نام ریشه

❑ توسط سرویس دهنده های نام محلی که قادر به تحلیل نام نمی باشند، مورد تماس قرار می گیرد.

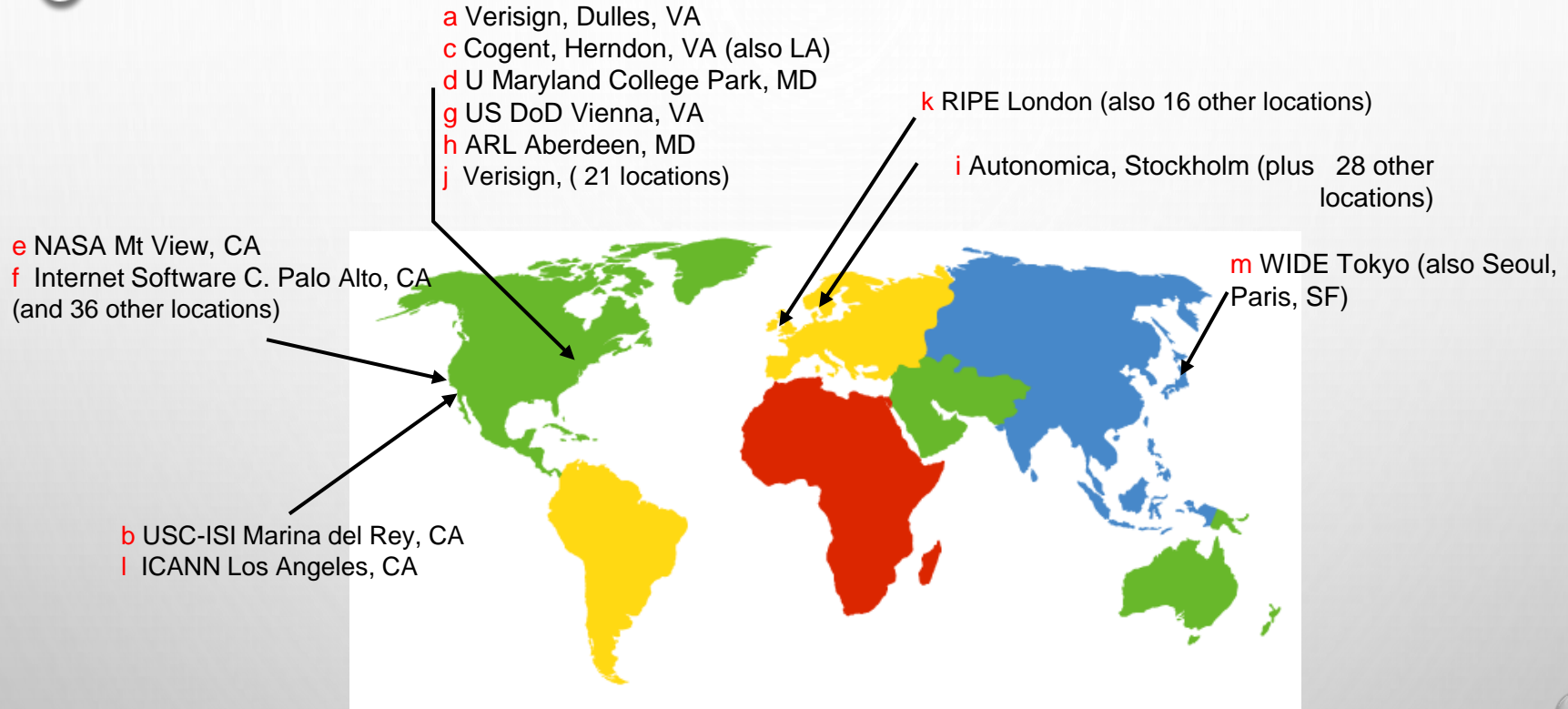
❑ سرویس دهنده های نام ریشه:

❖ اگر نگاشت نام ناشناخته باشد، با سرویس دهنده های نام مسئول تماس برقرار می نمایند.

❖ نگاشت را به سرویس دهنده های نام محلی بر می گردانند.

DNS: ROOT NAME SERVERS

سرویس دهنده های نام ریشه



13 root name servers
worldwide

TLD AND AUTHORITATIVE SERVERS

سرویس دهنده های نام ریشه و مسئول

❑ سرویس دهنده های TLD

❖ مسئول حوزه های سطح بالای `com, org, net, edu, ...` و تمام حوزه های سطح بالای کشورها نظیر `uk, fr, ca, jp, ir, ...` می باشند.

❖ سرویس دهنده های مربوط به حوزه سطح بالای `com` توسط `Network Solutions` نگهداری می شود.

❖ برای حوزه سطح بالای `edu` توسط `Educause`

❑ سرویس دهنده های DNS مسئول

❖ سرویس دهنده های `DNS` سازمانها، نگاشت نام به آدرس های `IP` را برای سرویس دهنده های سازمانها (نظیر وب و ایمیل) فراهم می آورند.

❖ می توانند توسط خود سازمانها و یا شرکتهای فراهم آورنده سرویس (`Service Provider`) نگهداری شوند.

LOCAL NAME SERVER

سرویس دهنده های نام محلی

❑ متعلق به سلسله مراتب حساب نمی شوند.

❑ هر ISP (خانگی، شرکتی و دانشگاهی) یکی برای خودش دارد.

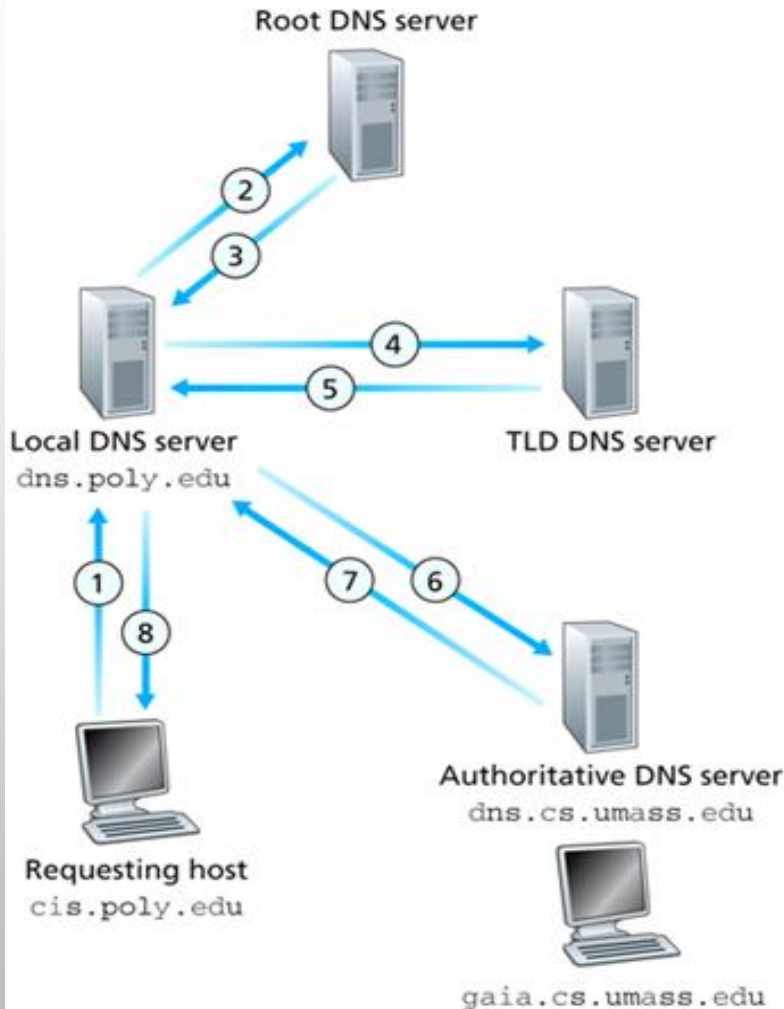
❖ تحت عنوان “default name server” شناخته می شود.

❑ زمانی که میزبانی درخواست DNS می نماید، درخواستها به این سرویس دهنده DNS محلی ارسال می شوند.

❖ همانند پروکسی عمل نموده و درخواستها را به سوی سلسله مراتب هدایت می نماید.

DNS NAME RESOLUTION EXAMPLE

مثالی از تحلیل نام DNS



♦ Interaction of the various DNS servers

□ میزبان ما در cis.poly.edu بوده و آدرس IP
gaia.cs.umass.edu را میخواهد.

روش درخواست تکراری (Iterative Query)

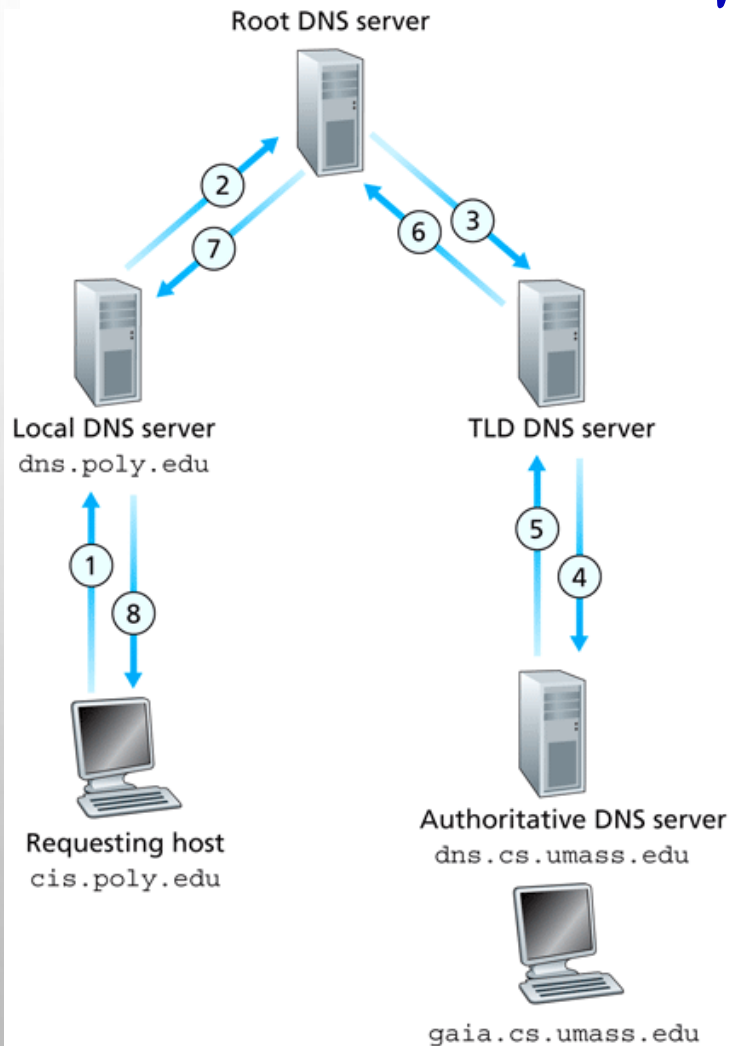
□ سرویس دهنده تماس گرفته شده، نام سرویس
دهنده ای را که باید با آن تماس بگیریم را برمی
گرداند.

□ ”من این نام را نمی شناسم، اما از این سرویس
دهنده سوال کن“

تحقیق : مراحل ثبت یک نام در اینترنت را بررسی

DNS NAME RESOLUTION EXAMPLE

مثالی از تحلیل نام DNS



روش درخواست بازگشتی (Recursive Query)

□ بار تحلیل نام را بر عهده سرویس دهنده نام

تماس گرفته شده، می گذاریم!

□ آیا بار سنگین خواهد شد؟

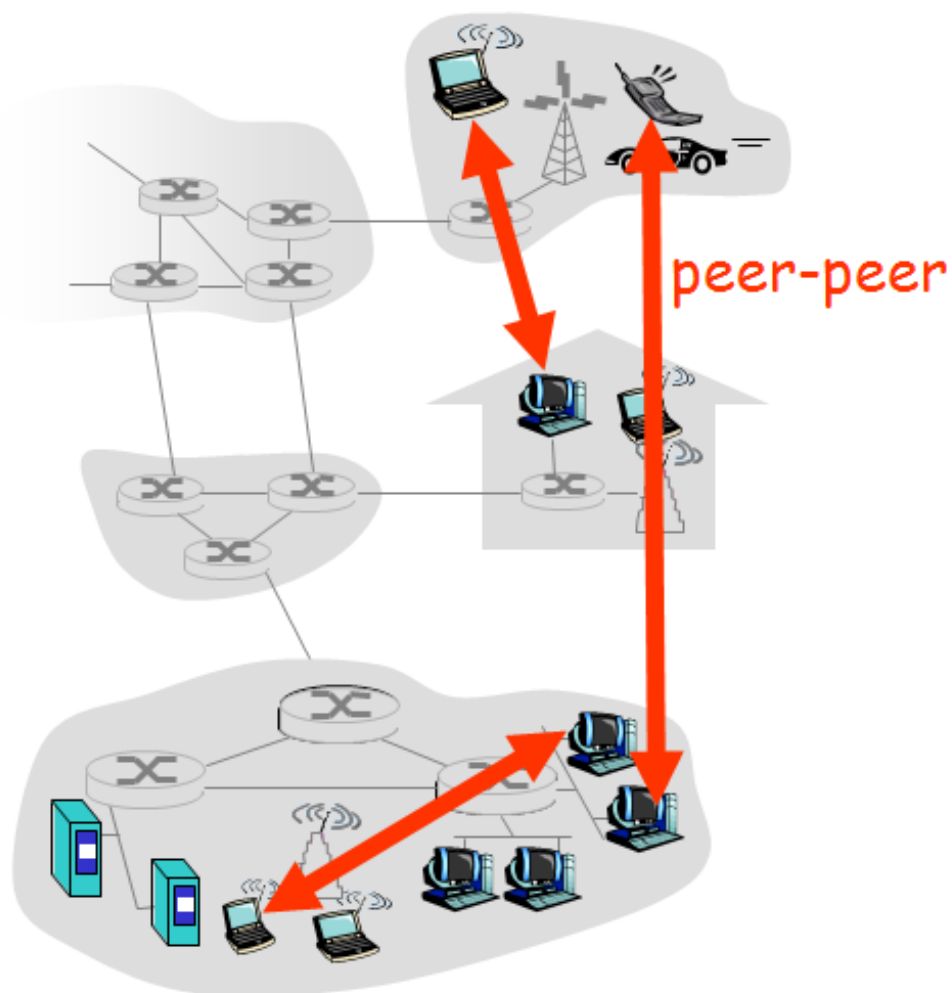
DNS: CACHING RECORDS

کش کردن رکوردها

- زمانیکه (هر) سرویس دهنده نامی، نگاشت (نام به IP) را یاد گرفت، آنرا کش می نماید.
- ❖ ورودیهای کش دارای زمان انقضاء می باشند. (بعد از مدتی از بین میروند)
- ❖ معمولا (آدرس) سرویس دهنده های TLD در سرویس دهنده های نام محلی کش می شود.
- بنابراین سرویس دهنده های ریشه کمتر بازدید میشوند.

PURE P2P ARCHITECTURE

معماری همتا به همتای خالص



❑ عدم نیاز به سرویس دهنده همواره روشن

❑ سیستم های پایانی دلخواه می توانند با همدیگر ارتباط داشته باشند.

❑ همتاها بطور متناوب وصل شده و تغییر آدرس IP میدهند.

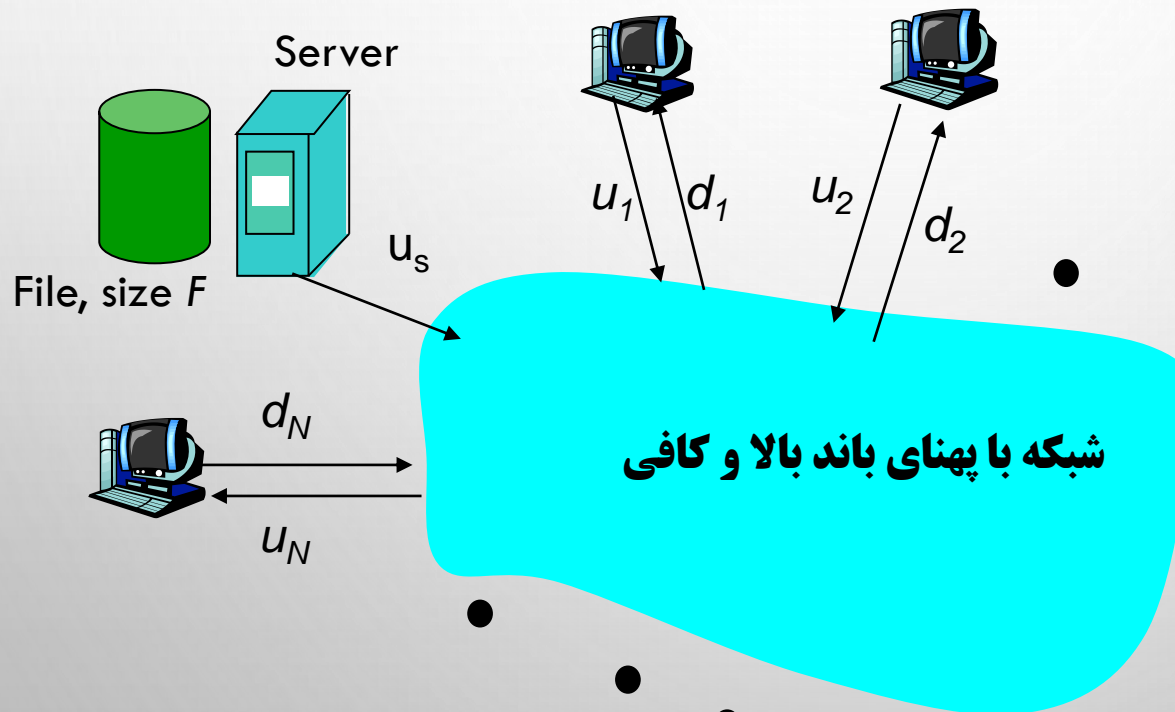
❑ THREE TOPICS:

- ❖ FILE DISTRIBUTION
- ❖ SEARCHING FOR INFORMATION
- ❖ CASE STUDY: SKYPE

FILE DISTRIBUTION: SERVER-CLIENT VS. P2P

توزیع فایل : سرویس دهنده – مشتری در مقابل همتا به همتا

سوال: زمان مورد نیاز برای توزیع یک فایل از یک سرویس دهنده به N همتا چه مقدار می باشد؟



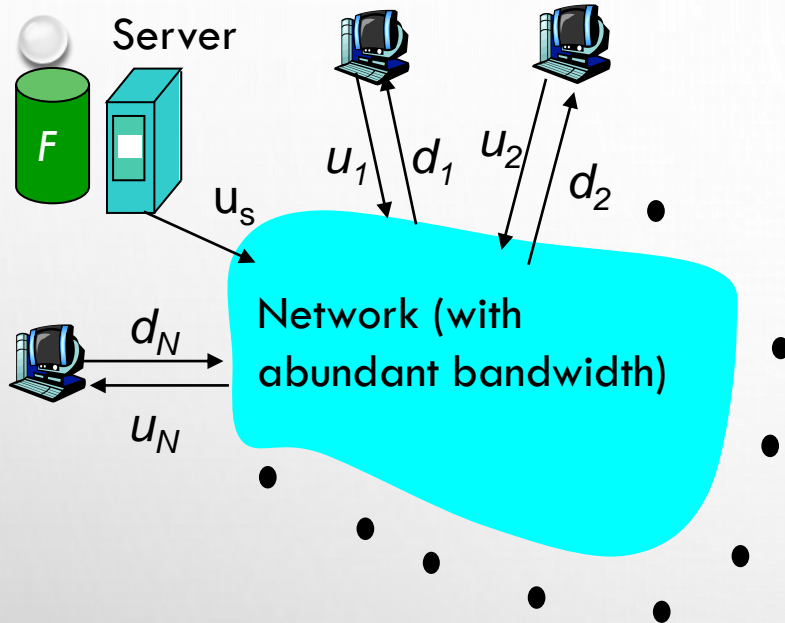
u_s : server upload bandwidth

u_i : peer i upload bandwidth

d_i : peer i download bandwidth

FILE DISTRIBUTION TIME: SERVER-CLIENT

زمان توزیع فایل : سرویس دهنده – مشتری



□ سرویس دهنده بایستی به تنهایی، N کپی

از فایل را (برای تمام مشتریان) ارسال نماید.

❖ زمان لازم : NF/u_s

□ مشتری i ام، نیاز به F/d_i برای دانلود دارد.

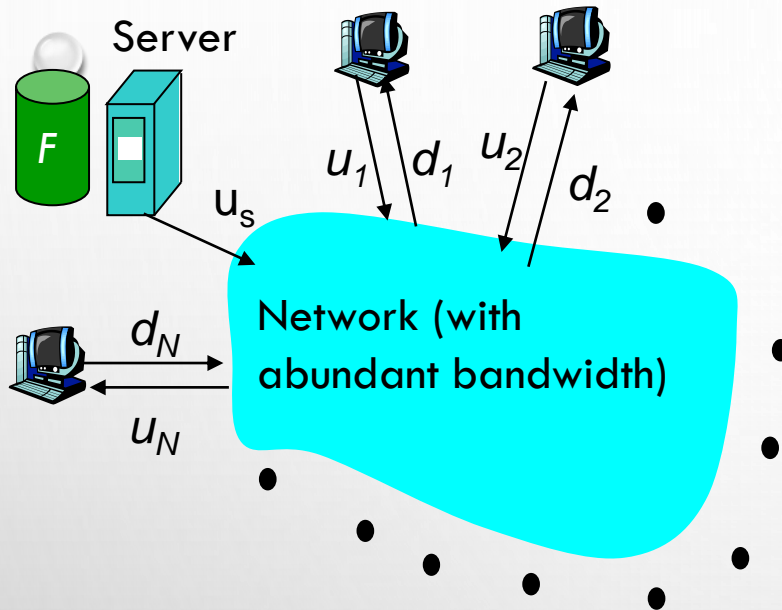
$$d_{cs} = \max \left\{ NF/u_s, F/\min(d_i) \right\}$$

زمان لازم برای توزیع N کپی از فایل بین تمام مشتریان

increases linearly in N
(for large N)

FILE DISTRIBUTION TIME: P2P

زمان توزیع : همتا به همتا



❑ سرویس دهنده بایستی حداقل یک کپی از فایل را ارسال نماید. (زیرا تنها منبع فایل، سرویس دهنده است)

❖ زمان لازم : F/u_s

❑ مشتری i ام، نیاز به زمان F/d_i برای دانلود دارد.

❑ حال هر مشتری می تواند بعد از دریافت فایل، در آپلود فایل به مشتریان دیگر کمک کند.

❑ بایستی NF بیت دانلود شود. (بطور جمعی)

❖ سریعترین نرخ آپلود ممکن : $u_s + \sum u_i$

$$d_{P2P} = \max \left\{ F/u_s, F/\min(d_i), NF/(u_s + \sum u_i) \right\}$$

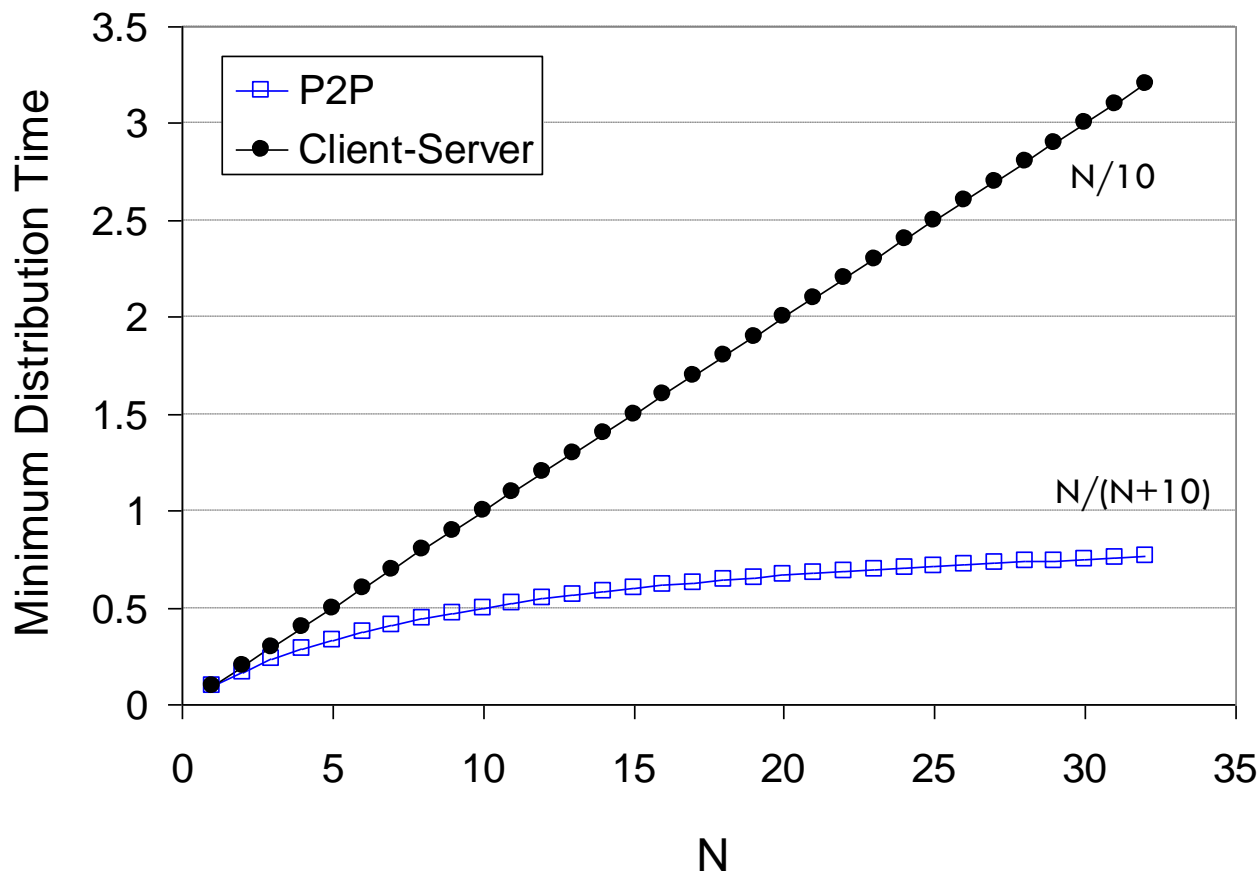
i

تهیه کننده: فرناد آهنگری

Server-client vs. P2P: example

مثالی از مقایسه سرویس دهنده-مشتری و همتا به همتا

Client upload rate = u , $F/u = 1$ hour, $u_s = 10u$, $d_{\min} \geq u_s$



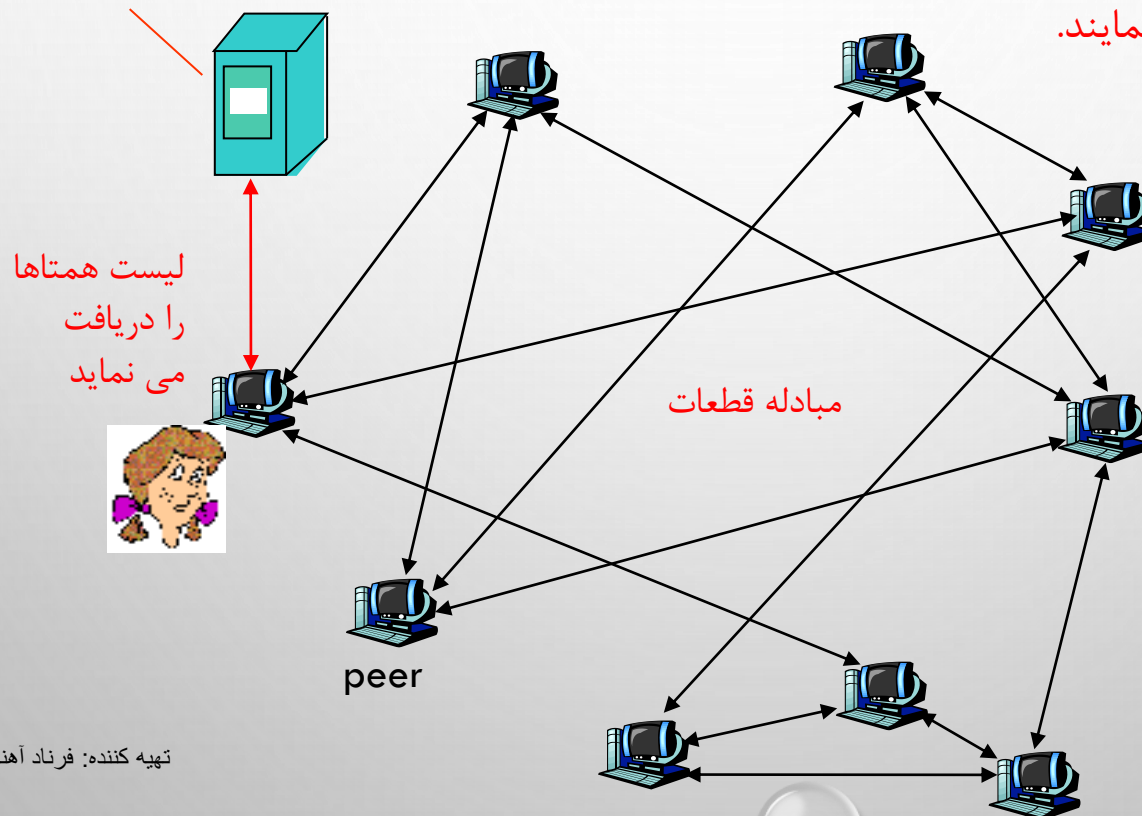
FILE DISTRIBUTION: BITTORRENT

توزیع فایل : بیت تورنت

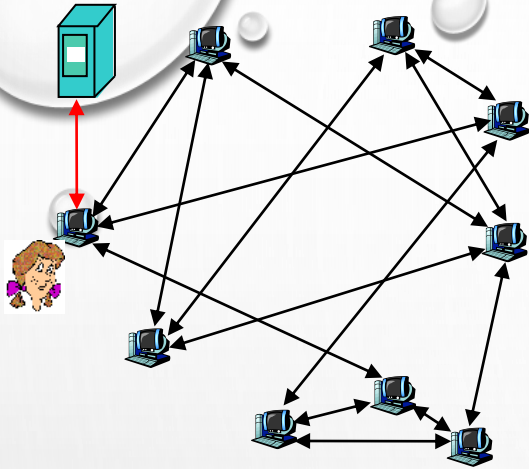
□ یک سیستم توزیع فایل همتا به همتا می باشد.

Tracker: سیستمی است که همتهای شرکت کننده در تورنت را دنبال می نماید.

Torrent: مجموعه ای از همتهای که قطعات فایل مبادله می نمایند.



BITTORRENT (1)



❑ فایل به قطعات ۲۵۶ کیلوبایت تقسیم می شود. (chunk)

❑ همتایی که به تورنت می پیوندد:

❖ دارای هیچ قطعه ای نیست، اما به تدریج آنها را جمع خواهد نمود.

❖ برای بدست آوردن لیست همتاها خود را در تراکر ثبت می نماید، و به زیر مجموعه ای از همتاها (همسایه ها) متصل می شود.

❑ هر همتا، هنگامی که فایلی را دانلود می نماید، قطعاتی را نیز به همتاهای دیگر آپلود می نماید.

❑ همتاها ممکن است وارد شوند و یا خارج شوند.

❑ زمانی که همتایی فایل را بطور کامل دریافت نمود، ممکن است (بطور خودخواهانه ای) تورنت را ترک

نموده و یا همچنان باقی بماند (برای کمک به آپلود فایل به دیگران)

BITTORRENT (2)

دریافت قطعات

□ در هر لحظه ای از زمان، همتهای مختلف، قطعات مختلفی را در اختیار دارند.

□ همتایی (نظیر الیس) بطور تناوبی، لیست قطعات همسایه های خود درخواست می نماید.

□ آلیس قطعاتی را که ندارد، درخواست می نماید.

❖ ابتدا کمیاب ترین قطعه (بین تمام تورنت ها) را درخواست می نماید.

ارسال قطعات

□ آلیس، به چهار همسایه ای که بالاترین نرخ ارسال قطعات به او را دارند، قطعات ارسال می دارد.

❖ هر ۱۰ ثانیه یکبار، ۴ تا از بالاترین را دوباره محاسبه می نماید.

□ هر ۳۰ ثانیه : بطور تصادفی یک همتای دیگری را انتخاب می نماید.

❖ ممکن است همتای جدید، جزء یکی از ۴ بالاترین بشود.

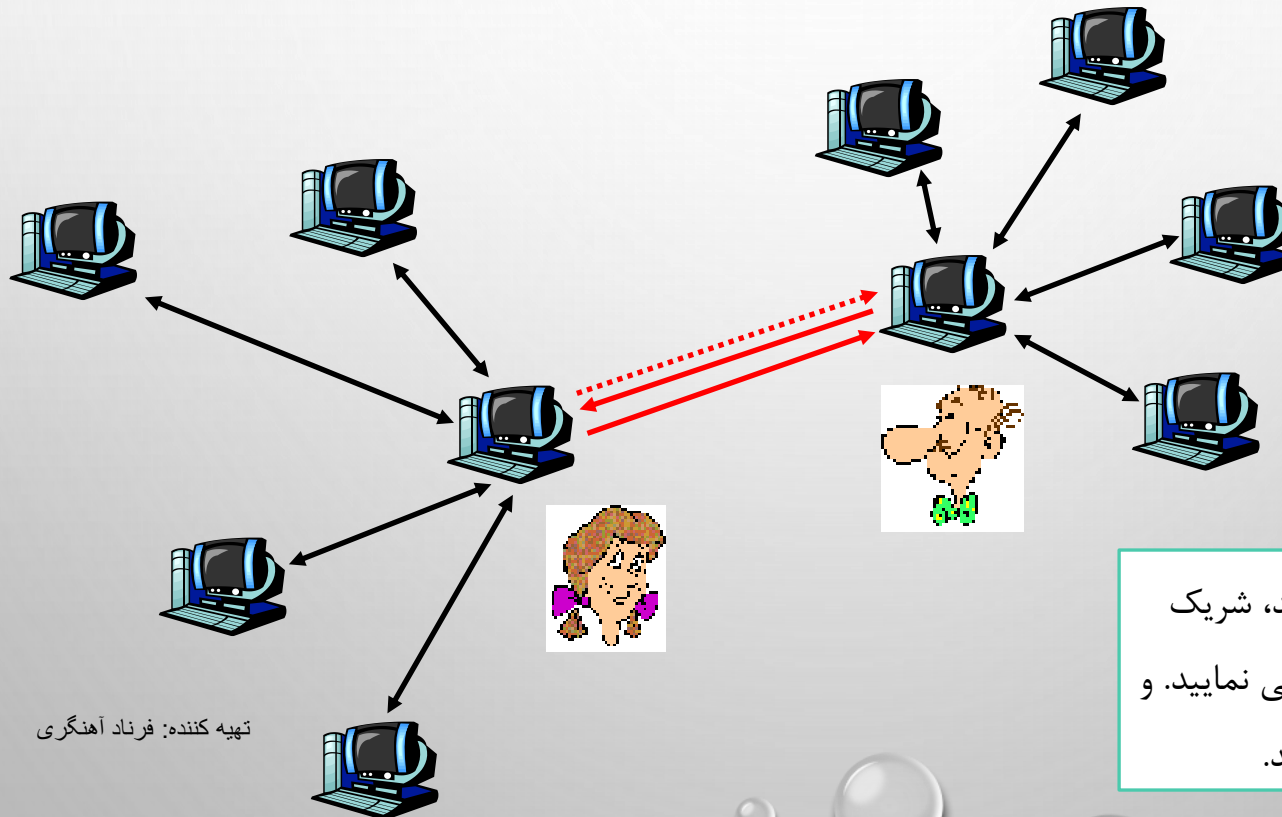
□ اگر دو همتا در معامله با هم راضی باشند، در لیست

۴ نفره همدیگر قرار می گیرند. تهیه کننده: فرناد آهنگری

BITTORRENT: TIT- FOR – TAT

بیت تورنت: این به جای آن

- (1) آلیس با سرعت مناسبی به باب قطعات ارسال می دارد.
- (2) آلیس جزء یکی از ۴ بالاترین ارسال کننده به باب می شود، باب شروع به معامله (ارسال) با آلیس می نماید.
- (3) باب هم یکی از ۴ بالاترین آلیس می شود.



هر چه نرخ آپلود شما بالاتر باشد، شریک
بهتری برای دریافت فایل پیدا می نمایید. و
سریعتر فایل را دریافت می دارید.

P2P: SEARCHING FOR INFORMATION

همتا به همتا: جستجوی اطلاعات

عمل ایندکس کردن در سیستم همتا به همتا : نگاشت اطلاعات (فایل و ...) به محل (آدرس IP و پورت) همتا در سیستم های همتا به همتا، ایندکس کردن یکی از حساس ترین و مهمترین وظایف می باشد.

پیام رسانی فوری (Instant messaging)

□ ایندکس، نام کاربران را به محل آنها نگاشت می نماید.

□ زمانیکه کاربری برنامه IM خود را اجرا می نماید، بایستی محل خود را به ایندکس اطلاع دهد.

□ همتها، برای یافتن آدرس IP کاربر مورد نظر، ایندکس را جستجو می نمایند.

اشتراک گذاری فایل (بطور مثال e-mule)

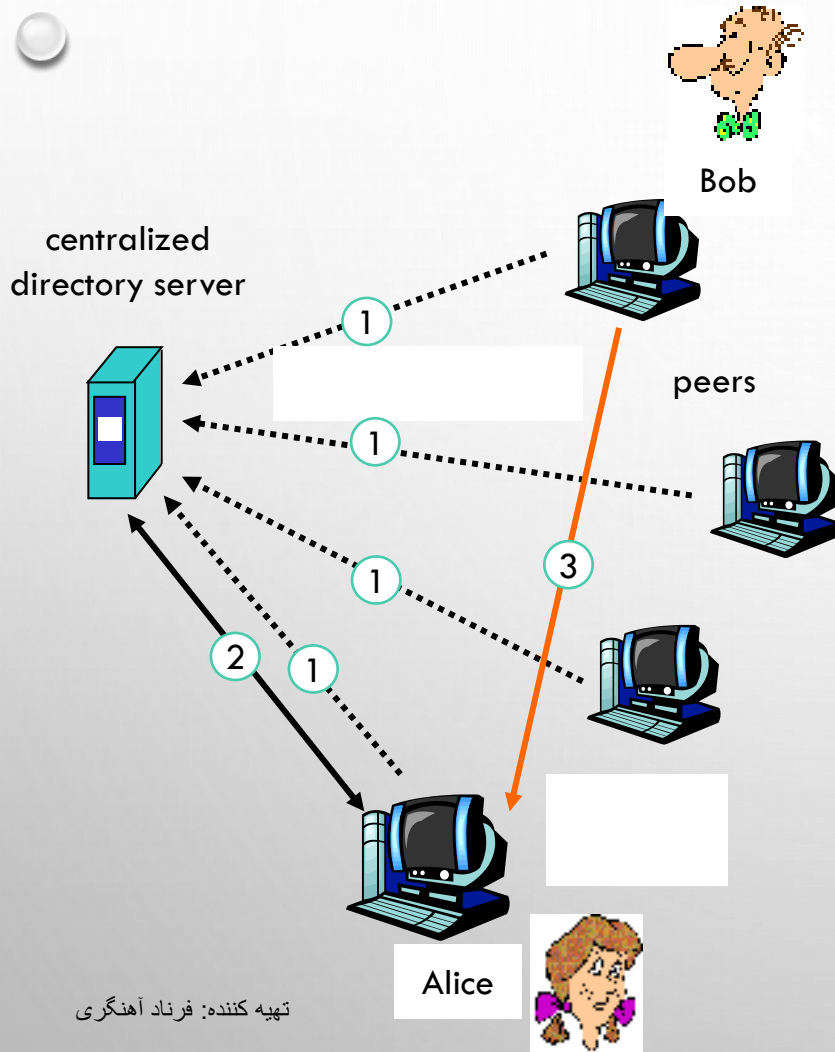
□ ایندکس، بطور دینامیک محل فایلهایی که همتها به اشتراک گذاشته اند را دنبال می نماید.

□ همتها بایستی، آنچه که آنها دارند (برای به اشتراک گذاری) را به ایندکس اعلام نمایند.

□ همتها، برای یافتن فایل مورد نظر، ایندکس را جستجو می نمایند.

P2P: CENTRALIZED INDEX

همتا به همتا : ایندکس متمرکز



در طرح اولیه "Napster" مورد استفاده قرار گرفته بود.
نیاز به سرویس دهنده بسیار بزرگی می باشد.

(1) زمانی که همتایی وارد می شد، به سرویس دهنده مرکزی اطلاع میدهد.

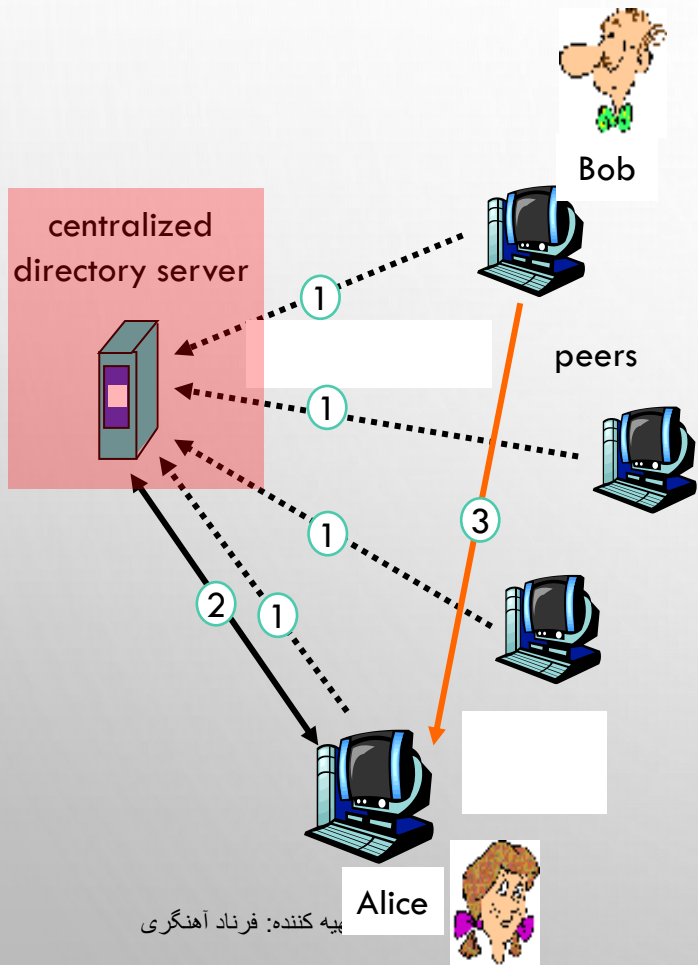
❖ آدرس IP خود و فایل‌هایی را که دارد.

(2) آلیس بدنبال "Hey Jude" می گردد، بنابراین درخواست آنرا به ایندکس می فرستد.

(3) با دریافت محل داده مورد نظر (از ایندکس)، آنرا بطور مستقیم از همتای مربوطه دریافت می دارد.

P2P: PROBLEMS WITH CENTRALIZED DIRECTORY

همتا به همتا : مشکلات دایرکتوری متمرکز



پیه کننده: فرناد آهنگری

- ❑ نقطه تنهای خرابی (Single point of failure)
- ❑ گلوگاهی برای کارایی سیستم (Performance bottleneck)
- ❑ نقض کپی رایت (Copyright infringement)

انتقال فایل بصورت غیر متمرکز می باشد
اما یافتن محل فایلها کاملا متمرکز می باشد.

QUERY FLOODING

سیلاب پرس و جوها

شبکه پوششی : گراف

- اگر بین همتای X و همتای Y یک ارتباط TCP برقرار باشد، یک یال در نظر گرفته می شود.
- مجموع تمام همتاها و یالها، یک شبکه پوشش را تشکیل می دهد.
- یالها، لینکهای مجازی هستند (و نه فیزیکی)
- چنین شبکه پوشش ممکن است شامل صدها هزار همتا باشد.
- هر همتایی عموماً به کمتر از ۱۰ همسایه در شبکه پوشش متصل می باشد.

□ کاملاً توزیع شده (غیر متمرکز) می باشد.

❖ هیچ سرویس دهنده مرکزی وجود ندارد.

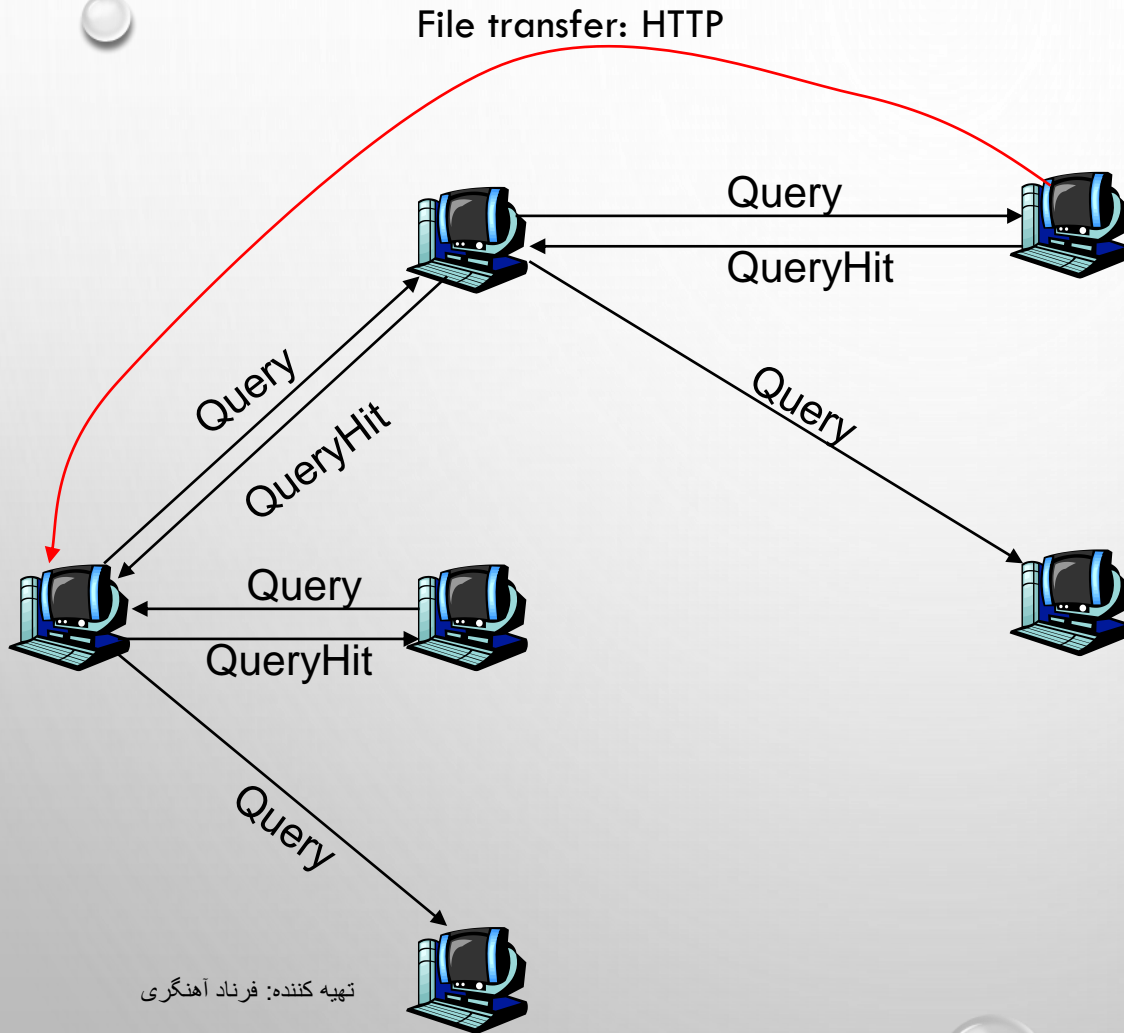
❖ اندیس بطور کامل در بین اجتماع همتاها توزیع می شود.

□ توسط گنوتلا (Gnutella) مورد استفاده قرار گرفته است.

□ هر همتا فقط مسئول ایندکس کردن فایل های خود که قرار است به اشتراک بگذارد، می باشد.

QUERY FLOODING

سیلاب پرس و جوها



تهیه کننده: فرناد آهنگری

□ پیامهای پرس و جو از طریق

ارتباطات TCP موجود، ارسال می شوند.

□ همتاها پیامهای پرس و جو را هدایت

به جلو می نمایند

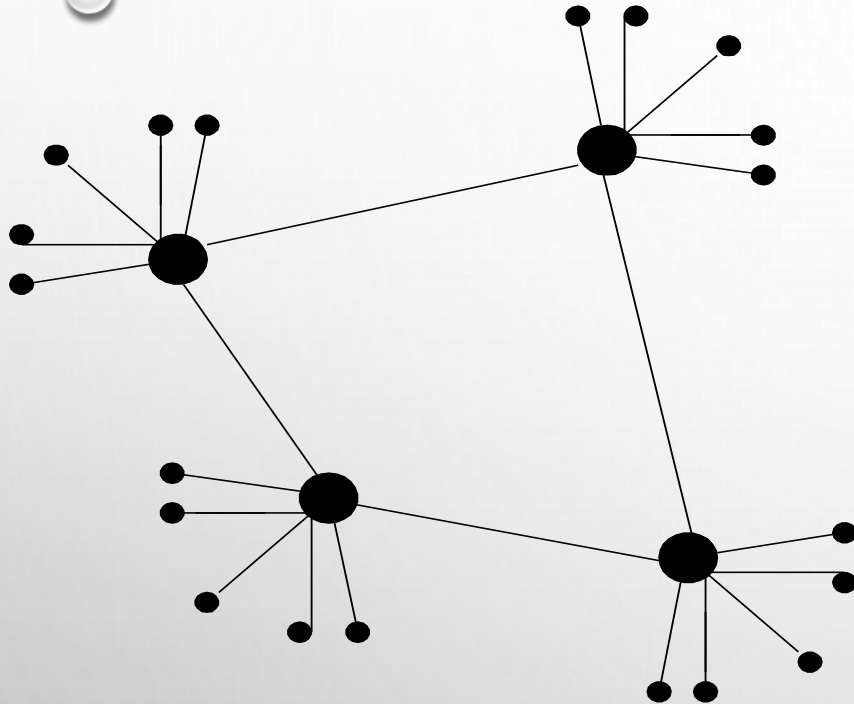
□ برخورد پرس و جوها (QueryHit)

از طریق مسیر معکوس ارسال می شوند.

مقیاس پذیری : سیلاب با دامنه
محدود استفاده می شود.

HIERARCHICAL OVERLAY

پوشش سلسله مراتبی



● ordinary peer

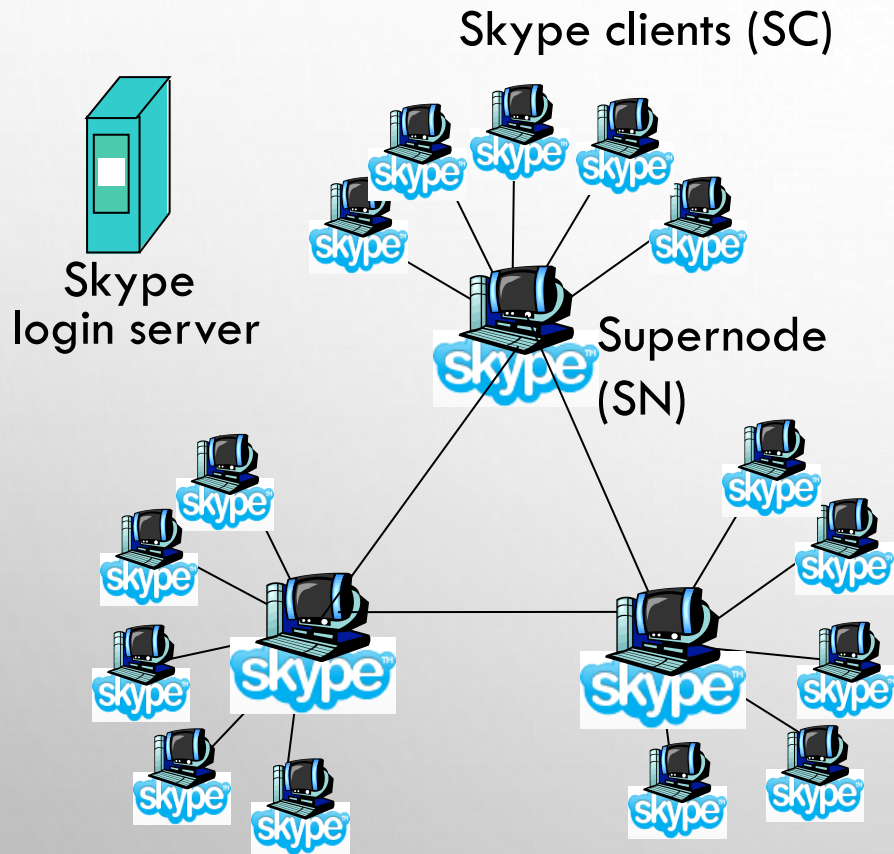
● group-leader peer

neighboring relationships
in overlay network

- طرحی است بین ایندکس متمرکز و سیلاب
- هر همتا، یا یک سوپر نود است و یا به سوپر نودی منتسب شده است.
- ❖ ارتباط TCP بین همتا به سوپر نود
- ❖ ارتباط TCP بین برخی همتا های سوپر نود ها
- سوپر نود محتوای (فایلها و ...) مربوط به فرزندان را دنبال می نماید.

P2P CASE STUDY: SKYPE

مطالعه موردی : اسکایپ



□ اساساً طرحی همتا به همتا می باشد: زیرا همتاها

مستقیماً با هم ارتباط برقرار می نمایند.

□ دارای پروتکل لایه کاربرد اختصاصی می باشد

(شرح پروتکل موجود نیست، اما آنرا از طریق

مهندسی معکوس استخراج نموده اند)

□ از شبکه پوششی سلسله مراتبی با سوپر نود

استفاده می نماید.

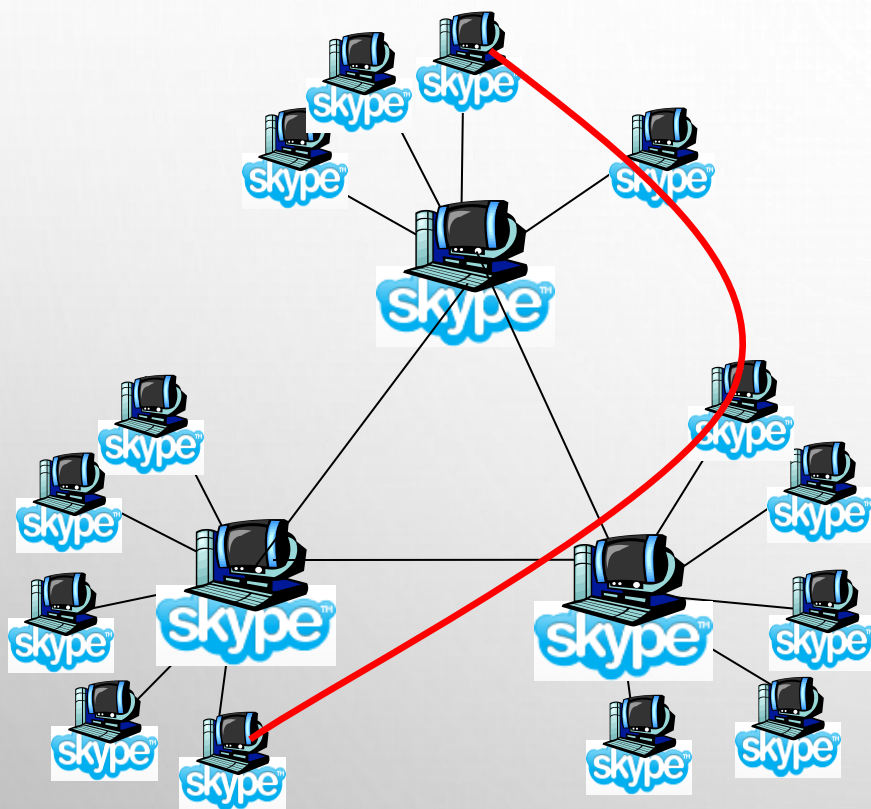
□ ایندکس، بین نام کاربران و آدرس IP آنها،

نگاشت برقرار می نماید.

□ ایندکس بین سوپر نودها توزیع شده است.

PEERS AS RELAYS

همتاها بعنوان رله (کننده)



❑ زمانی که دو همتا (آلیس و باب) پشت NAT قرار

داشته باشند، مشکل برقراری ارتباط با هم را پیدا می نمایند.

❖ زیرا NAT مانع برقراری شروع ارتباط از همتای

بیرون (از NAT) به همتای داخل می شود.

❑ راه حل :

❖ سوپر نود مربوط به آلیس و باب بعنوان رله مورد

استفاده قرار می گیرد.

❖ هر همتا با سوپر نود خود ارتباط برقرار نموده

❖ حال، همتاها می توانند از طریق رله از NAT از

رله عبور نمایند.