

لایه کاربرد

اصول برنامه های کاربردی شبکه

رئوس مطالب

❖ معماری لایه کاربرد و برنامه های کاربردی

❖ معماری وب و پروتکل HTTP

❖ پروتکل انتقال فایل (FTP)

❖ پست الکترونیکی

▪ SMTP, POP3, IMAP

❖ سرویس دهنده نام دامنه (DNS)

❖ برنامه های کاربردی P2P

معماری برنامه های کاربردی

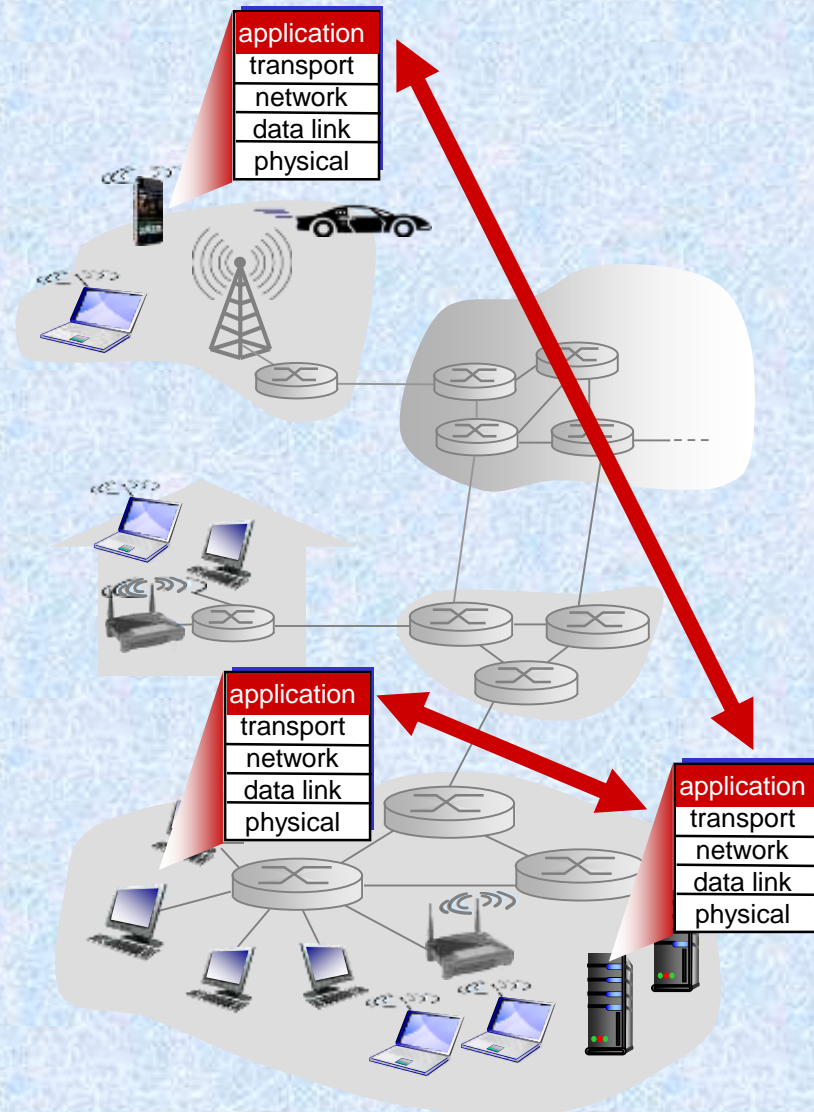
❖ خصوصیات برنامه های کاربردی

- قابل اجرا در تمامی سیستم های انتهایی
- قابلیت ارتباط در شبکه

• مثال: ارتباط برنامه سرویس دهنده وب با مرورگرهای وب

▪ عدم اجرای برنامه سمت مشتری در تجهیزات هسته شبکه

▪ قابلیت برنامه های سمت مشتری برای توسعه پذیری سریع



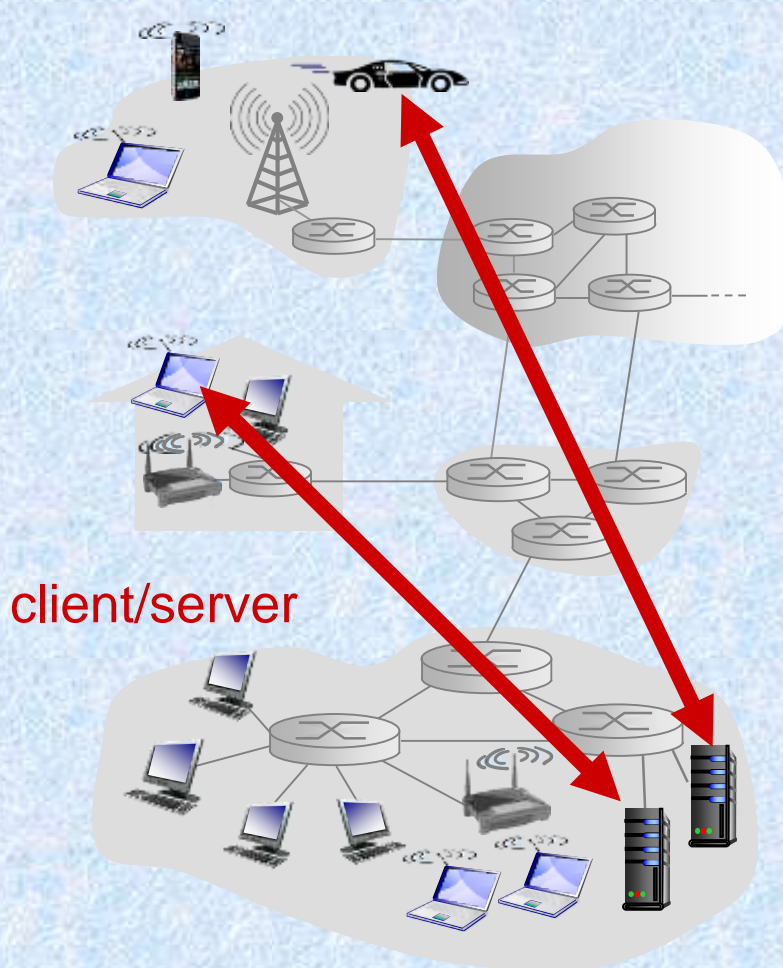
معماری سرویس دهنده-مشتری

❖ سرویس دهنده

- میزبان های همیشه فعال
- دارای آدرس های اینترنتی ثابت
- امکان گستردگی در قالب مراکز داده

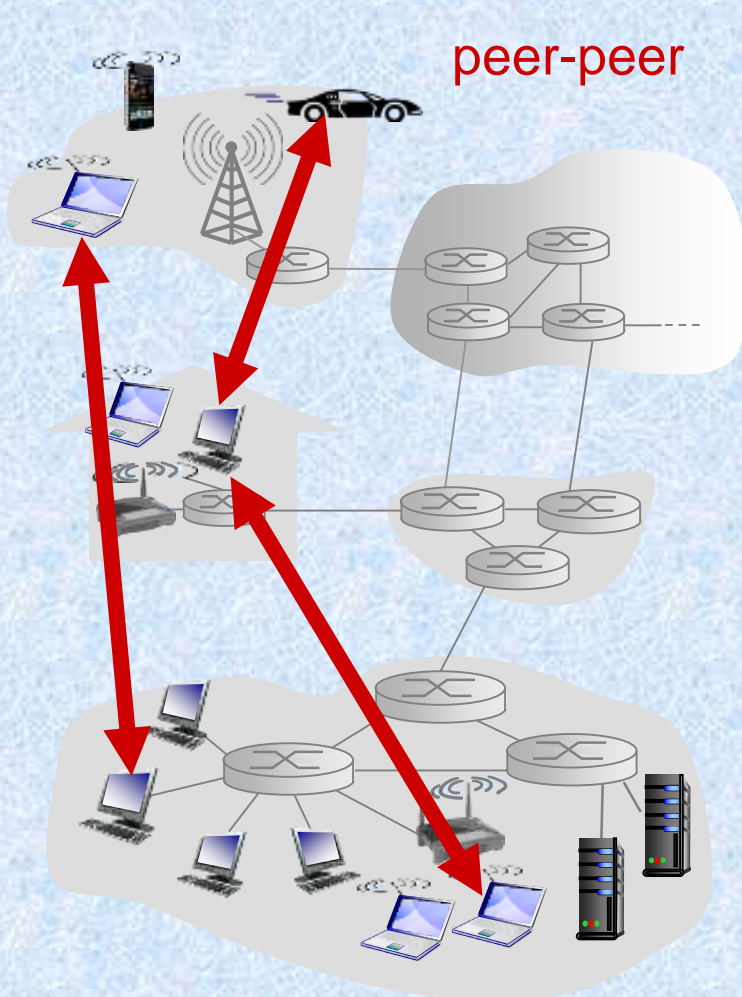
❖ مشتری

- قابلیت ارتباط با برنامه سرویس دهنده
- امکان اتصال متناوب
- دارای آدرس های اینترنتی متغیر یا ثابت
- عدم امکان ارتباط مستقیم مشتری ها با همدیگر



معماری نظیر-به-نظیر (P2P)

peer-peer



❖ عدم نیاز به سرویس دهنده های همیشه فعال

❖ میزبان های متصل به همدیگر

❖ درخواست سرویس توسط یک نظیر از نظیر دیگر

▪ ارائه سرویس ها از همان نظیر دیگر

▪ خود مقیاس پذیر

• افزوده شدن ظرفیت سرویس با افزایش هر نظیر، افزایش درخواست سرویس

❖ اتصال متناوب نظیر ها و تغییر مرتب آدرس های اینترنتی

❖ مدیریت پیچیده، امنیت، مغایرت با سیاست ISP

ارتباط پردازش ها

❖ پردازش

▪ برنامه ی در حال اجرا در یک میزبان

❖ ارتباط بین دو پردازش در یک میزبان
از طریق روال های سیستم عامل
انجام می شود

❖ ارتباط بین دو پردازش از دو میزبان
متفاوت از طریق تبادل پیام انجام می
شود.

❖ پردازش ها در سرویس دهنده-
مشتری

❖ پردازش مشتری

▪ آغاز کننده ارتباط

❖ پردازش سرویس دهنده

▪ منتظر برای درخواست

❖ نکته

▪ معماری P2P نیز دارای پردازش های
سمت مشتری و سرویس دهنده است

ارتباط پردازش و شبکه

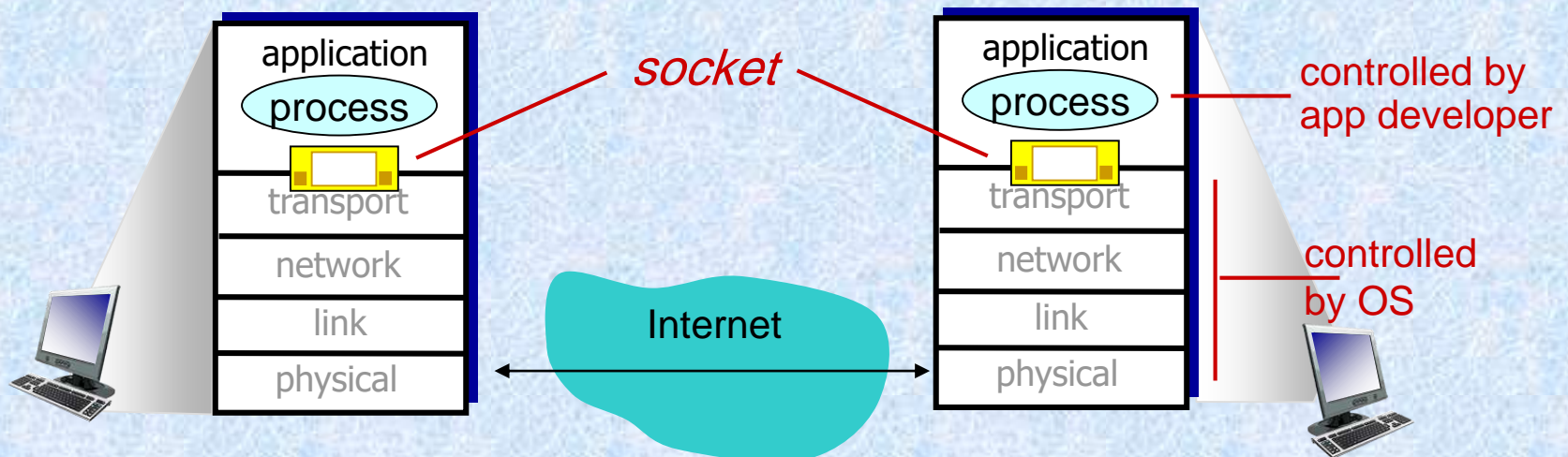
❖ کلیه پیام های بین پردازش ها از شبکه عبور می کند

❖ سوکت (دروازه)

- فرآیند (برنامه)ی برای ارسال پیام از طریق واسط نرم افزاری روی شبکه
- واسط بین لایه کاربرد و انتقال در میزبان (واسط برنامه نویسی کاربردی یا API)

❖ موارد قابل کنترل سوکت در لایه کاربرد

- انتخاب پروتکل، اعمال تنظیمات محدود



آدرس دهی پردازش ها

❖ نیاز به آدرس مقصد برای ارسال پیام
به میزبان مقصد

❖ موارد مورد نیاز

▪ آدرس اینترنتی میزبان مقصد

▪ عبارت از آدرس های IP

▪ شناسه پردازش گیرنده در میزبان مقصد

❖ شناسه پردازش

▪ عبارت از آدرس (نقطه دسترسی) برنامه

سرویس دهنده (TSAP)

❖ آدرس های نقاط سرویس استاندارد

▪ 1024 شماره اول: اختصاص یافته به

سرویس های استاندارد توسط سازمان

IANA

❖ آدرس های توسعه

▪ بقیه آدرس ها تا شماره 65535

▪ قابل استفاده برای برنامه ها (سوکت

ها)ی اختصاصی برنامه نویسی

خصوصیات پروتکل های لایه کاربرد

- ❖ تعریف انواع پیام های قابل انتقال
 - مثال: درخواست، پاسخ ...
- ❖ قالب دستوری پیام
 - فیلدهای موجود و نحوه استفاده
- ❖ مفهوم پیام
 - مفهوم اطلاعاتی در پیام حمل شده
- ❖ قوانین
 - زمان و چگونگی ارسال و یا دریافت پیام
- ❖ پروتکل های لایه کاربرد
 - استاندارد شده توسط RFC
- ❖ با قابلیت ارتباط بین پردازش
- ❖ مثال: HTTP, SMTP...
- ❖ پروتکل های جانبی (خارج از استاندارد)

سرویس های انتقال لایه کاربرد

❖ حفظ جامعیت (یکپارچگی) داده

- نیاز برخی داده ها به قابلیت اطمینان ۱۰۰٪ (هیچ بیت داده ای مجاز به حذف نیست)
- مثال: انتقال فایل، اسناد وب
- داده هایی با تحمل پذیری حذف صوت

❖ برآورد تاخیر زمانی

- نیاز برخی برنامه ها به تاخیر پایین تر برای کارکرد بهتر
- مثال: تلفن اینترنتی، بازی های تعاملی

❖ تامین گذردهی مناسب

- نیاز برخی برنامه ها به تامین آستانه ای از گذردهی قابل پذیرش
- مثال: برنامه های چند رسانه ای
- برنامه هایی با قابلیت استفاده از هر اندازه گذردهی موجود

❖ تامین امنیت

- تامین خدمات امنیتی
- محرمانگی، دسترسی، جامعیت، انکارناپذیری، صحت

نیازهای انتقال برنامه های کاربردی

| نام برنامه | تلف داده | گذردهی | حساسیت به تاخیر |
|-----------------------|-------------|---|-----------------|
| انتقال فایل | بدون تلف | قابل انعطاف | بدون حساسیت |
| پست الکترونیک | بدون تلف | قابل انعطاف | بدون حساسیت |
| | بدون تلف | قابل انعطاف | بدون حساسیت |
| صوت و تصویر بهنگام | قابل انعطاف | صوت: 5kbps-1Mbps تصویر: 10kbps-5Mbps | کمتر از 100ms |
| صوت و تصویر ذخیره شده | قابل انعطاف | صوت: 5kbps-1Mbps تصویر: 10kbps-5Mbps | کمتر از 100ms |
| بازهای تعاملی | قابل انعطاف | بیشتر از 10s kbps | چند ثانیه |
| پیام متنی | بدون تلف | قابل انعطاف | میلی ثانیه |

برنامه های کاربردی اینترنت

| نام برنامه | پروتکل لایه کاربرد | پروتکل لایه انتقال |
|------------------------|--------------------|--------------------|
| پست الکترونیک | SMTP | TCP |
| دسترسی ترمینال راه دور | Telnet | TCP |
| وب | HTTP | TCP |
| انتقال فایل | FTP | TCP |
| جریان چند رسانه ای | HTTP RTP | TCP/UDP |
| تلفن اینترنتی | SIP, RTP | TCP/UDP |

وب و پروتکل HTTP

❖ محتوای صفحات وب شامل انواع مختلف اشیاء

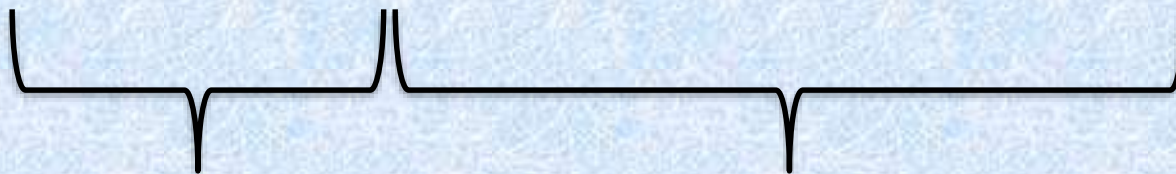
❖ قالب های موجود

▪ فایل های **HTML**، تصاویر **JPEG**، اپلت های جاوا، فایل های صوتی...

❖ صفحه وب شامل یک فایل پایه به زبان **HTML** با انواع مختلف اشیاء ارجاع شده

❖ هر شیء قابل آدرس دهی توسط یک قالب منحصر به فردی برای تعیین محل آن در میزبان سرویس دهنده (**URL**)

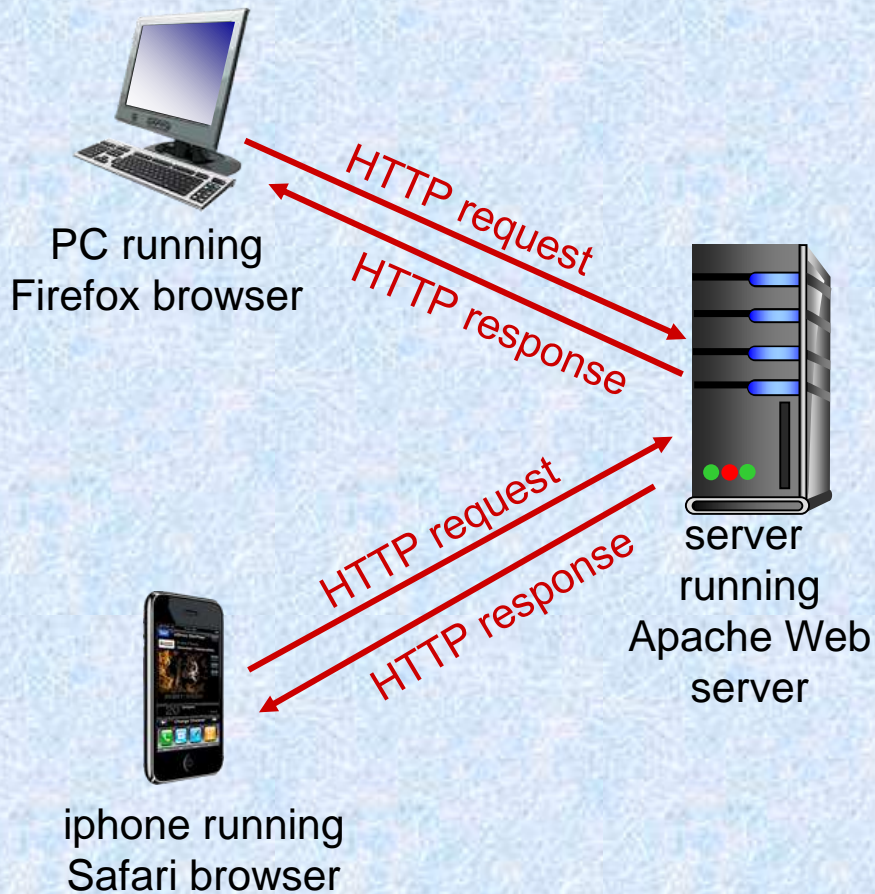
❖ **www.uut.ac.ir/somefolder/someimagefile.jpg**



نام میزبان

مسیر شیء

پروتکل HTTP



❖ پروتکل انتقال ابرمتن

❖ پروتکل لایه کاربرد برای وب

❖ مبتنی بر معماری سرویس دهنده-مشتري

❖ مشتري

▪ مرورگر: برنامه ای برای درخواست، دریافت، اجرا و نمایش اشیاء وب بر اساس پروتکل HTTP

❖ سرویس دهنده

▪ سرویس دهنده وب برای دریافت درخواست، ارسال اشیاء درخواستی بر اساس پروتکل HTTP

...پروتکل HTTP

❖ استفاده از پروتکل انتقال TCP

❖ ایجاد اتصال TCP با ساخت سوکت روی آدرس پردازش تصادفی، ارسال به آدرس پردازش شماره 80 در سرویس دهنده

❖ پذیرش اتصال TCP توسط سرویس دهنده

❖ تبادل پیام های HTTP بین سرویس دهنده (HTTP Server) و مشتری (مرورگر)

❖ خاتمه اتصال TCP

❖ اتصال HTTP پایا

▪ ارسال و دریافت درخواست پاسخ HTTP روی یک اتصال TCP

❖ اتصال HTTP غیرپایا

▪ ارسال یک زوج درخواست/پاسخ (برای دریافت یک شیء) روی یک اتصال TCP

▪ نیاز به اتصال TCP دیگر برای اشیاء دیگر

▪ بستن اتصال پس از هر انتقال و ایجاد اتصال جدید برای انتقال بعدی

تحلیل زمانی اتصال غیرپایا

❖ زمان رفت و برگشت (RTT)

■ زمان لازم برای ارسال یک بسته کوچک به سرویس دهنده و دریافت پاسخ آن

❖ زمان پاسخ HTTP

❖ یک RTT برای برقراری اتصال TCP

❖ یک RTT درخواست HTTP و دریافت چند بایت اول پاسخ

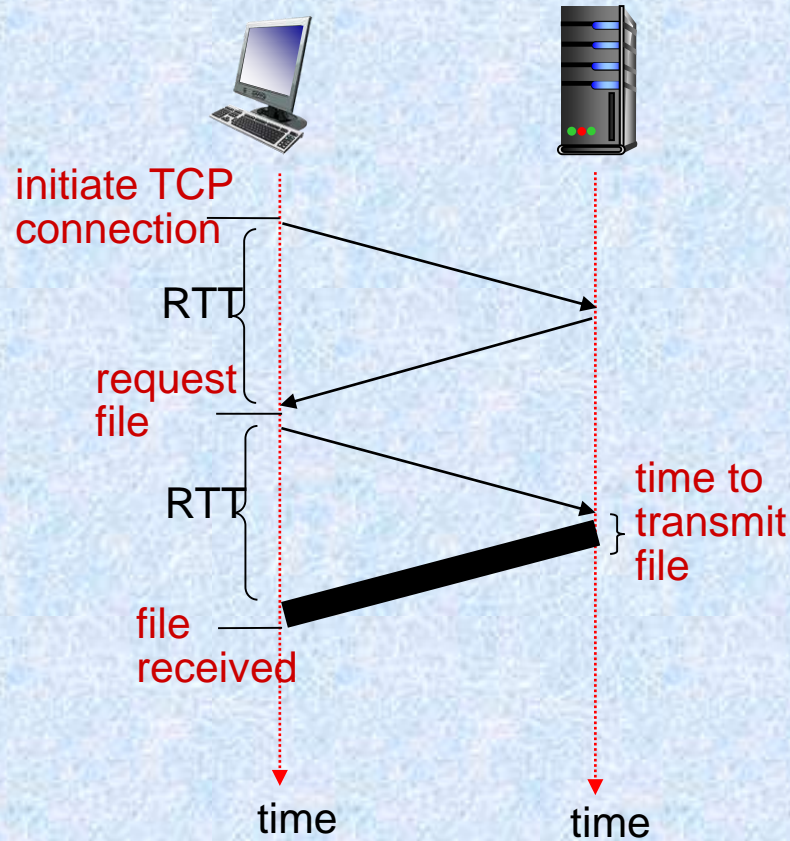
❖ زمان انتقال فایل

❖ زمان کل پاسخ غیرپایا

$2RTT + \text{file transmission time}$

❖ زمان کل پاسخ پایا

❖ نیاز به ۲ زمان RTT برای هر شیء



پیام درخواست HTTP

❖ با کد ASCII

❖ تشکیل شده از ۵ سطر، استفاده از کاراکترهای کنترلی در انتهای هر سطر

request line
(GET, POST,
HEAD commands)

header
lines

carriage return,
line feed at start
of line indicates
end of header lines

carriage return character
line-feed character

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

متدهای درخواست

❖ متد POST

- قابل استفاده برای فرم های قابل پر کردن
- اطلاعات تکمیل شده در مرورگر به سرویس دهنده منتقل می شود
- دارای سطری برای تعیین بدنه موجودیت (entity body)

❖ متد URL

- نام دیگر: متد GET
- قرار گرفتن اطلاعات درخواستی در بدنه URL

❖ متدهای دیگر

▪ PUT

- ارسال یک شیء به سرویس دهنده

▪ DELETE

- حذف یک شیء از سرویس دهنده

▪ HEAD

- درخواست ارسال پاسخ خالی

پیام پاسخ HTTP

status line
(protocol
status code
status phrase)

header
lines

data, e.g.,
requested
HTML file

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02
GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-
1\r\n
\r\n
data data data data data ...
```

کد وضعیت پیام پاسخ

| | |
|---|----------|
| موفقیت آمیز بودن درخواست، شیء در ادامه پاسخ قرار می‌گردد | 200 OK ❖ |
| شیء درخواستی بطور دائمی از سرویس دهنده حذف شده | 301 ❖ |
| عدم تشخیص درخواست توسط سرویس دهنده | 400 ❖ |
| سند درخواستی در سرویس دهنده یافت نشد | 404 ❖ |
| عدم پشتیبانی نسخه مورد استفاده پروتکل HTTP توسط سرویس دهنده | 505 ❖ |

| نوع کد | مفهوم | توصیف |
|--------|------------|---|
| 1yy | اطلاعاتی | شامل اطلاعات عمومی، بدون اطلاعات نتیجه |
| 2yy | موفق | متد توسط سرور دریافت و پذیرش شد |
| 3yy | تغییر مسیر | نیاز به عملیات بیشتر قبل از پذیرش درخواست |
| 4yy | خطای کاربر | درخواست نامعتبر، عدم دریافت کامل یا دارای خطای دستوری |
| 5yy | خطای سرور | درخواست معتبر، عدم پذیرش بدلیل مشکلات سرور |

کدهای وضعیت (موفق)

| <i>Code</i> | <i>Phrase</i> | <i>Description</i> |
|----------------------|---------------|--|
| Informational | | |
| 100 | Continue | The initial part of the request has been received and the client may continue with its request. |
| 101 | Switching | The server is complying with a client request to switch protocols defined in the upgrade header. |
| Success | | |
| 200 | OK | The request is successful. |
| 201 | Created | A new URL is created. |
| 202 | Accepted | The request is accepted, but it is not immediately acted upon. |
| 204 | No content | There is no content in the body. |

کدهای وضعیت (ناموفق)

| <i>Code</i> | <i>Phrase</i> | <i>Description</i> |
|---------------------|-----------------------|---|
| Redirection | | |
| 301 | Multiple choices | The requested URL refers to more than one resource. |
| 302 | Moved permanently | The requested URL is no longer used by the server. |
| 304 | Moved temporarily | The requested URL has moved temporarily. |
| Client Error | | |
| 400 | Bad request | There is a syntax error in the request. |
| 401 | Unauthorized | The request lacks proper authorization. |
| 403 | Forbidden | Service is denied. |
| 404 | Not found | The document is not found. |
| 405 | Method not allowed | The method is not supported in this URL. |
| 406 | Not acceptable | The format requested is not acceptable. |
| Server Error | | |
| 500 | Internal server error | There is an error, such as a crash, at the server site. |
| 501 | Not implemented | The action requested cannot be performed. |
| 503 | Service unavailable | The service is temporarily unavailable, but may be requested in the future. |

کوکی (Cookie)

❖ برای ردیابی کاربران توسط سرویس دهنده (به منظور شناسایی کاربر)

❖ اجزاء

- خط سرآیند کوکی در پاسخ HTTP

- خط سرآیند کوکی در درخواست HTTP

- یک فایل در میزبان مشتری

- پایگاه داده پشتیبان در وب سایت

❖ موارد استفاده

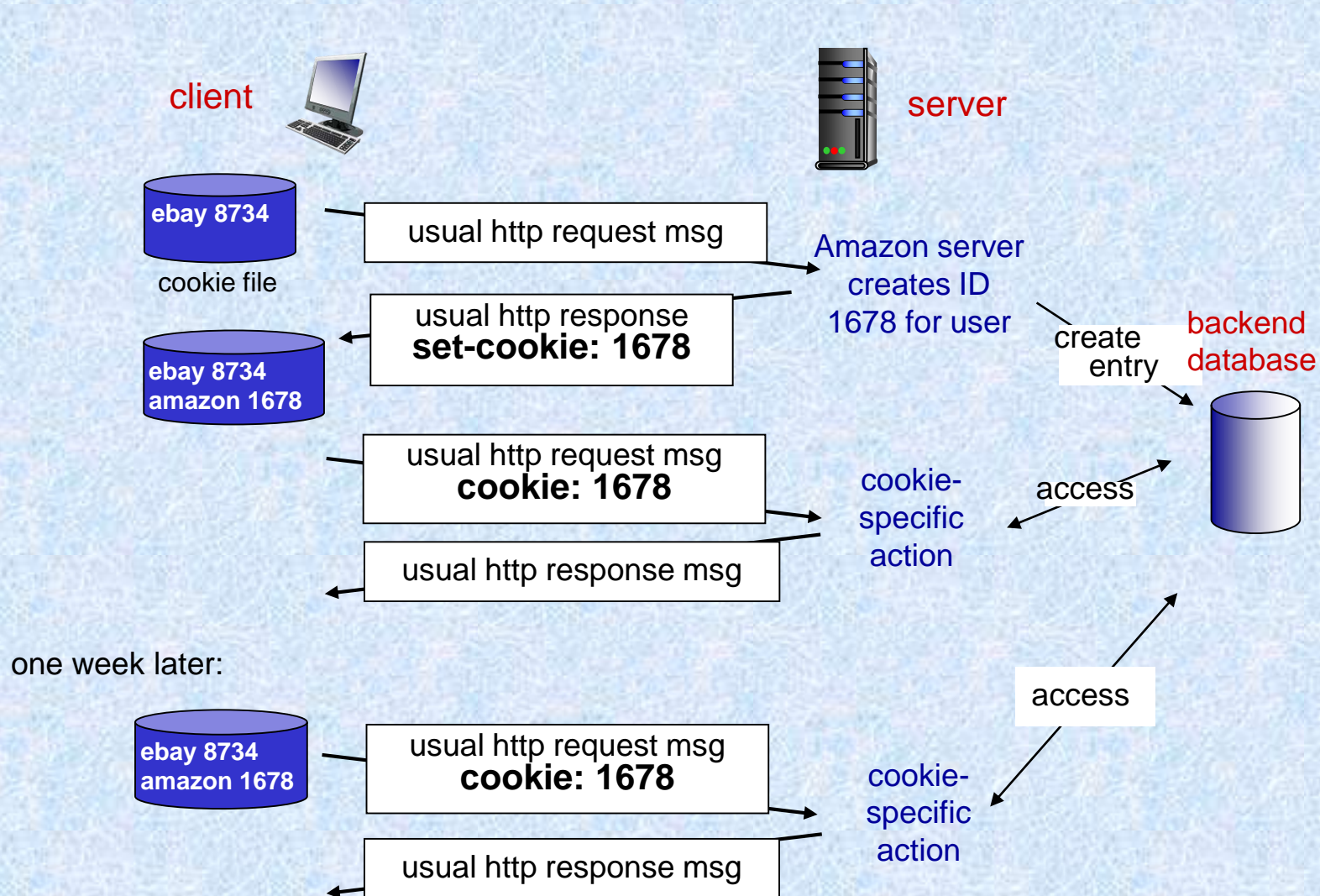
- احراز هویت

- کارت خرید

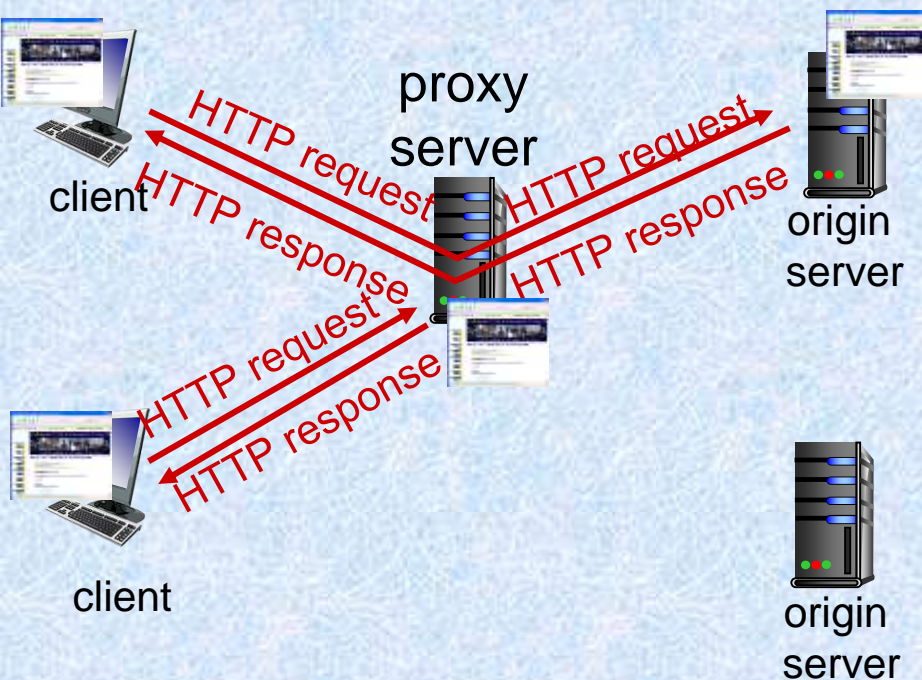
- پیشنهادهای

- وضعیت نشست کاربر

کوکی (Cookie)



حافظه نهان وب



❖ معروف به سرویس دهنده پراکسی

❖ واسطه سرویس دهنده وب برای پاسخ به درخواست ها

❖ اتصال **TCP** توسط مرورگر به حافظه نهان

❖ وجود شیء درخواستی در حافظه
← ارسال پاسخ به مشتری

❖ عدم وجود شیء در حافظه
← درخواست سرویس دهنده واسطه
از سرویس دهنده اصلی، ذخیره و
ارسال به مشتری

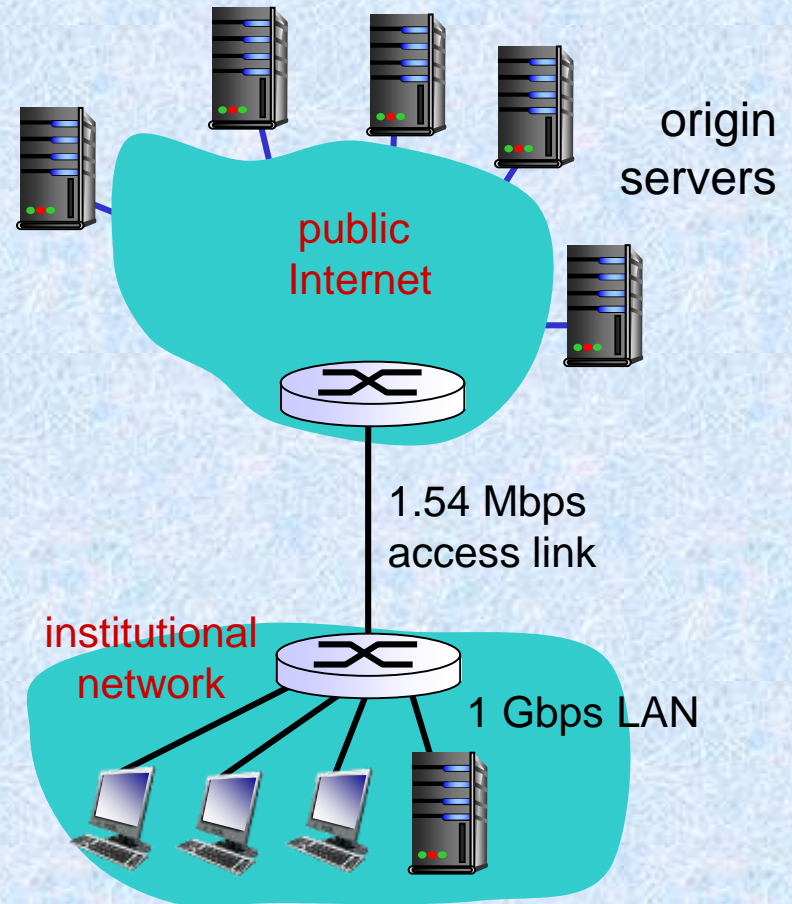
مثال: بدون حافظه ی نهان

assumptions:

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

consequences:

- ❖ LAN utilization: 15%
- ❖ access link utilization = 99% *problem!*
- ❖ total delay = Internet delay + access delay + LAN delay
= 2 sec + minutes + usecs



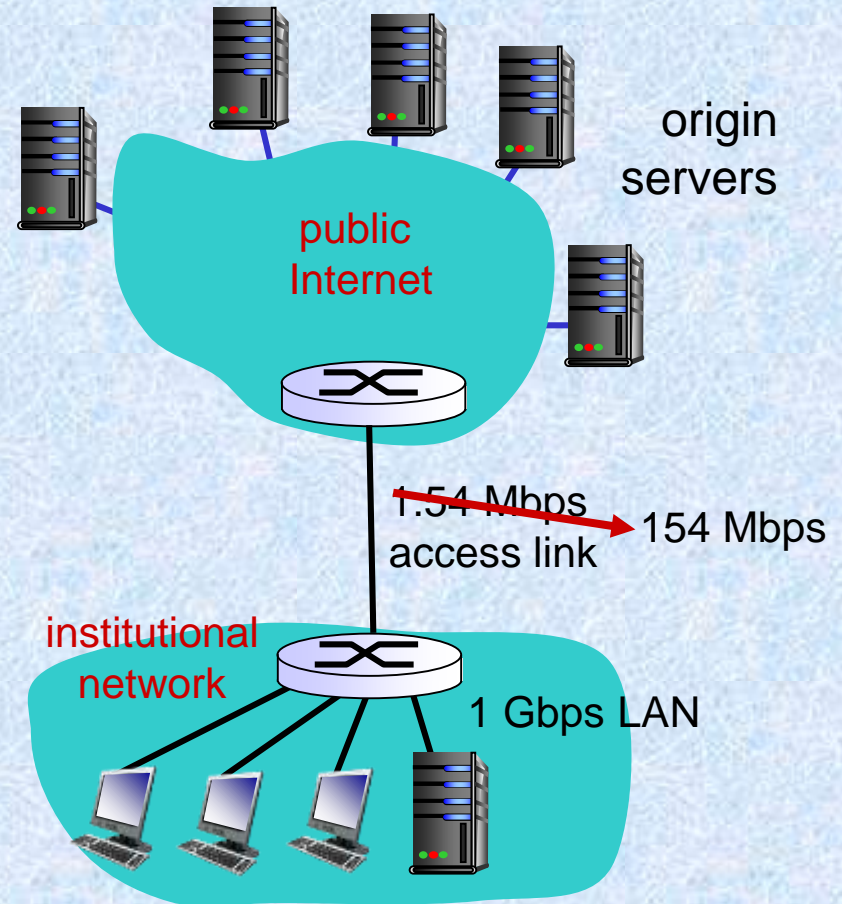
مثال: ارتقاء لینک دسترسی

assumptions:

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

consequences:

- ❖ LAN utilization: 15%
- ❖ access link utilization = ~~99%~~ → 9.9%
- ❖ total delay = Internet delay + access delay + LAN delay
= 2 sec + ~~minutes~~ → usecs
msecs



Cost: increased access link speed (not cheap!)

مثال: نصب حافظه ی نهان

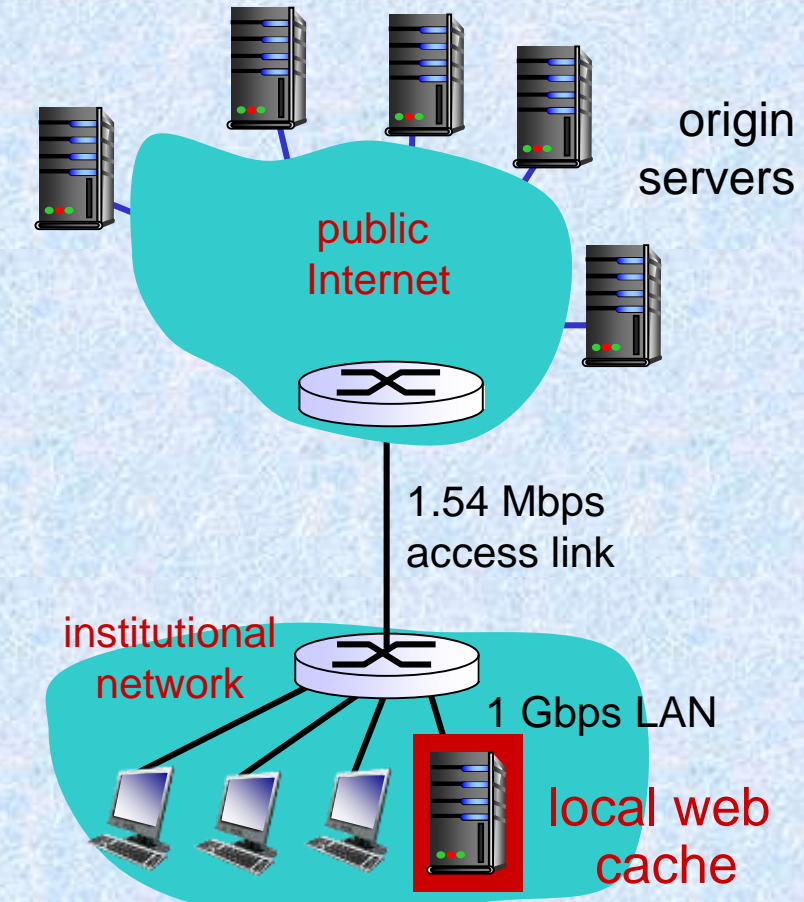
assumptions:

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

consequences:

- ❖ LAN utilization: 15%
- ❖ access link utilization = 100%
- ❖ total delay = Internet delay + access delay + LAN delay
= 2 sec + minutes + usecs

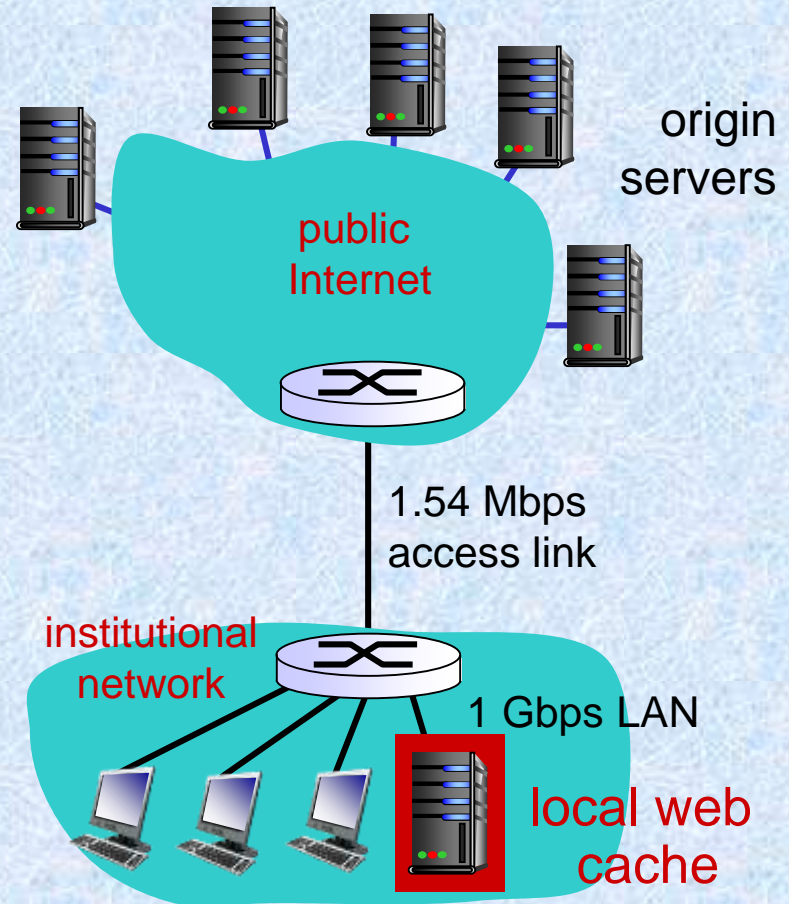
Cost: web cache (cheap!)



مثال: نصب حافظه ی نهان

Calculating access link utilization, delay with cache:

- ❖ suppose cache hit rate is 0.4
 - 40% requests satisfied at cache,
 - 60% requests satisfied at origin
- ❖ access link utilization:
 - 60% of requests use access link
- ❖ data rate to browsers over access link
 $= 0.6 * 1.50 \text{ Mbps} = .9 \text{ Mbps}$
 - utilization $= 0.9 / 1.54 = .58$
- ❖ total delay
 - $= 0.6 * (\text{delay from origin servers}) + 0.4 * (\text{delay when satisfied at cache})$
 - $= 0.6 (2.01) + 0.4 (\sim \text{msecs})$
 - $= \sim 1.2 \text{ secs}$
 - less than with 154 Mbps link (and cheaper too!)



پروتکل انتقال فایل (FTP)

❖ پروتکلی برای انتقال فایل با قابلیت اطمینان و کارائی بالا

❖ امکانات ارائه شده توسط FTP

- فهرست گیری از فایل‌های موجود روی سرور
- حذف، تغییرنام و جابجائی فایل
- ایجاد یا حذف شاخه
- انتقال فایل از سرور به مشتری
- انتقال فایل از مشتری به سرور

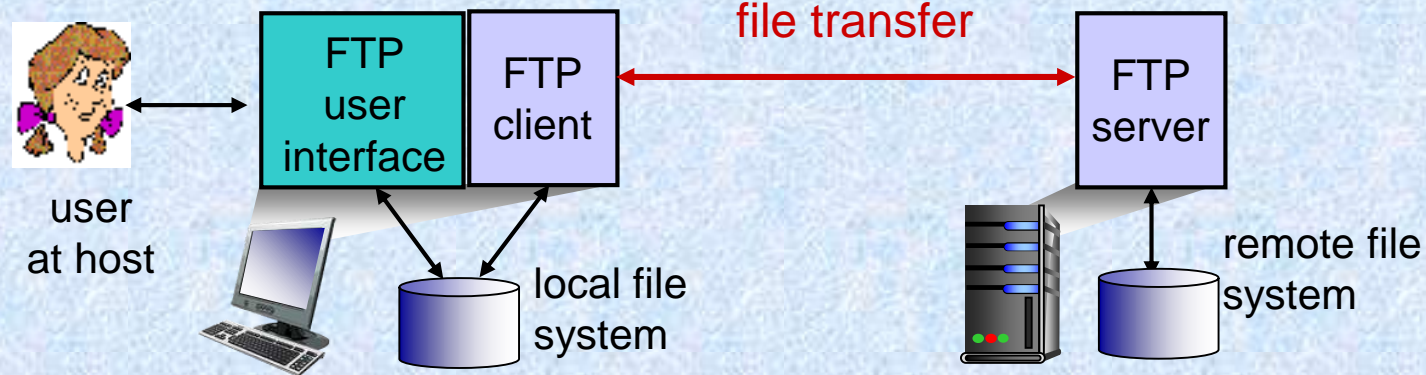
❖ استفاده از دو کانال مجزا برای انتقال داده و فرمان

❖ کانال داده: انتقال داده روی پورت ۲۰

❖ کانال فرمان: مبادله فرامین روی پورت ۲۱

فرآیند اتصال

- ❖ ارسال درخواست به سرویس دهنده روی پورت ۲۱
- ❖ تایید هویت مشتری، مرور فایل ها و ارسال فرمان روی کانال کنترل
- ❖ دریافت فرمان انتقال فایل توسط سرویس دهنده
 - برقراری اتصال دوم روی پورت ۲۰
- ❖ خاتمه اتصال، باز کردن اتصال دوم برای انتقال فایل بعدی توسط سرویس دهنده
- ❖ نگهداری وضعیت توسط سرویس دهنده
 - شاخه جاری، احراز هویت



فرمان های FTP

❖ کدهای پاسخ ساده

- ❖ **331, username OK**
 - Password required
- ❖ **125 data connection already open; transfer starting**
- ❖ **425 can't open data connection**
- ❖ **452 error writing file**

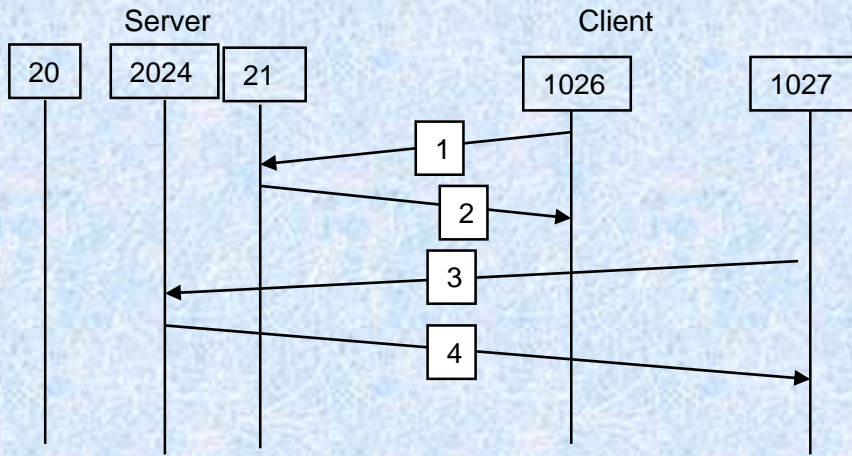
❖ فرمان های ساده

- ❖ **USER username**
- ❖ **PASS password**
- ❖ **LIST**
 - List current directory content
- ❖ **RETR filename**
- ❖ **STOR file**
 - puts file onto remote host

روش های ایجاد نشست

روش غیرفعال (Passive)

- ❖ ایجاد دو سوکت با شماره پورتهای تصادفی
- ❖ اتصال یکی از پورتهای با پورت ۲۱ سرور
- ❖ درخواست نشست غیرفعال توسط مشتری
- ❖ ایجاد یک سوکت با شماره پورت تصادفی در سرور و اعلام آن به مشتری
- ❖ اتصال سوکت دوم مشتری با پورت اعلام شده توسط سرور و شروع نشست



روش معمولی (Normal/Active)

- ❖ ایجاد دو سوکت با شماره پورت تصادفی
- ❖ اتصال یکی از پورتهای به پورت ۲۱ سرور، PI سرور آماده تفسیر فرامین
- ❖ اعلام شماره پورت سوکت دوم به سرور و شروع listen توسط مشتری
- ❖ درخواست اتصال توسط سرور به پورت اعلام شده
- ❖ تصدیق درخواست توسط مشتری و شروع نشست

