شبکه‌های کامپیوتری

مدرس: سعدون عزیزی

s.azizi@uok.ac.ir

گروه مهندسی کامپیوتر

نیم سال دوم ۹۶-۹۵

فصل دوم: لایه کاربرد

نگاه کلی:

- اصول برنامه‌های کاربردی شبکه
- وب و HTTP
- انتقال فایل: FTP
- پست الکترونیک: SMTP
- سرویس دایرکتوری اینترنت: DNS
- برنامه‌های کاربردی نظیر به نظیر: P2P
- برنامه‌نویسی سوکت: ایجاد برنامه‌های کاربردی شبکه

فصل دوم: راهنمای مسیر

- اصول برنامه‌های کاربردی شبکه
- وب و HTTP
- انتقال فایل: FTP
- پست الکترونیک: SMTP
- سرویس دایرکتوری اینترنت: DNS
- برنامه‌های کاربردی نظیر به نظیر: P2P
- برنامه‌نویسی سوکت: ایجاد برنامه‌های کاربردی شبکه

بعضی از برنامه‌های کاربردی شبکه

- ایمیل
- وب
- گروه‌های خبری
- دسترسی از راه دور
- اشتراک فایل P2P
- بازی‌های آنلاین چند نفره
- اشتراک ویدئو مانند یوتیوب
- برنامه‌های صدا روی IP (مانند اسکایپ)
- شبکه‌های اجتماعی (مانند فیسبوک)
- جستجو
- ...

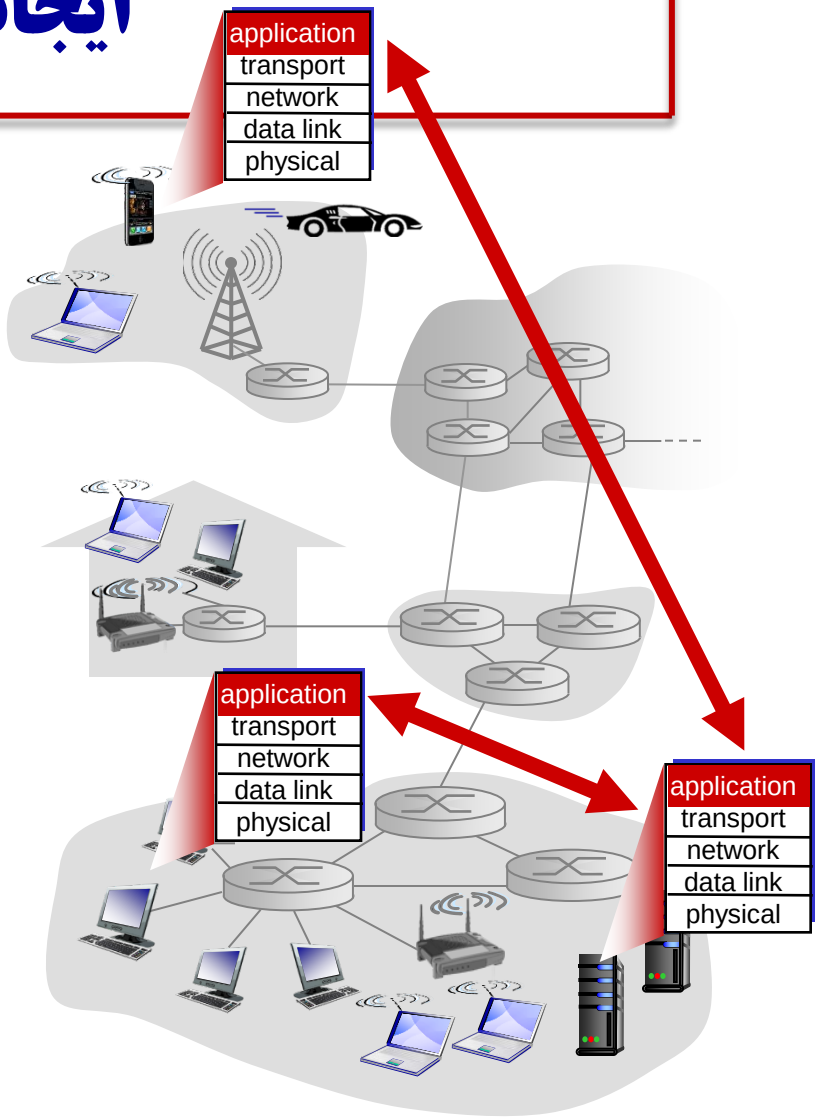
ایجاد یک برنامه شبکه

برنامه‌های شبکه:

- روی سیستم‌های انتهایی (مختلف) اجرا می‌شوند
- از طریق شبکه با هم ارتباط برقرار می‌کنند
- به عنوان مثال، برنامه سرویس‌دهنده وب با برنامه مرورگر ارتباط برقرار می‌کند

نیازی به نوشتن برنامه برای دستگاه‌های هسته شبکه نیست

- دستگاه‌های هسته شبکه برنامه‌های کاربران را اجرا نمی‌کنند
- این طراحی اصولی باعث سرعت و سهولت در توسعه و پخش برنامه‌های کاربردی شبکه می‌شود

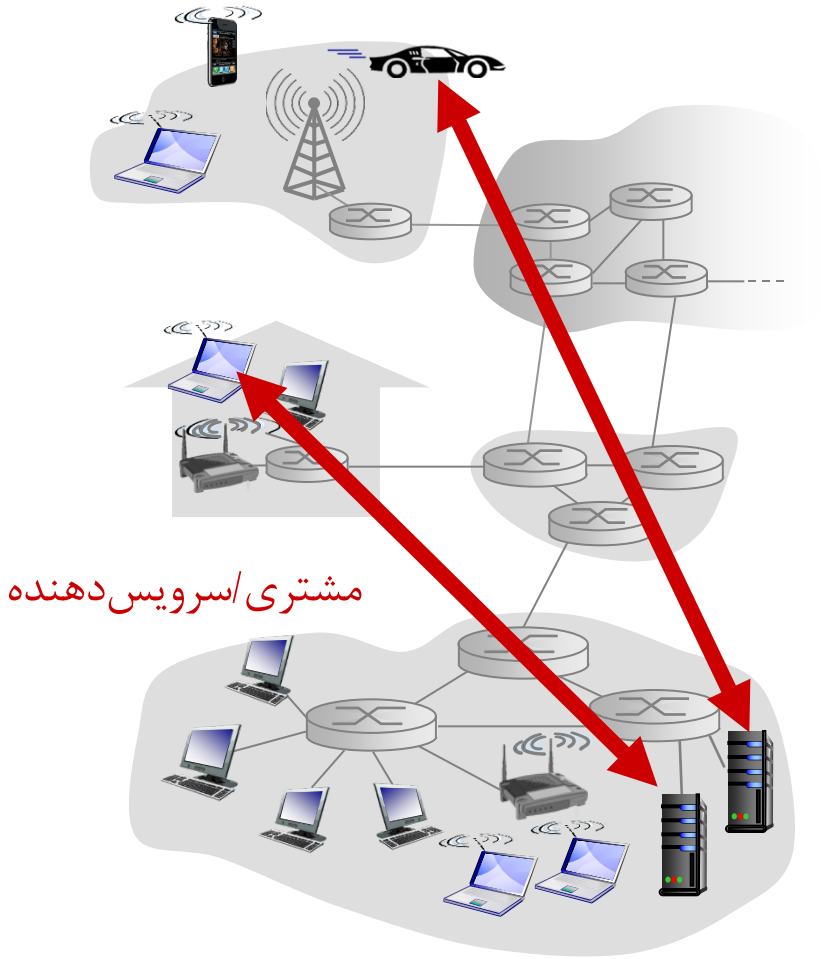


معماری برنامه‌های کاربردی شبکه

دو معماری غالب برای برنامه‌های کاربردی شبکه:

- معماری مشتری-سرویس‌دهنده
- معماری نظیر به نظیر (P2P)

معماری مشتری–سرویس‌دهنده



سرویس‌دهنده:

- میزبان همیشه روشن
- آدرس IP ثابت
- استفاده از مراکز داده برای مقیاس‌های بزرگ

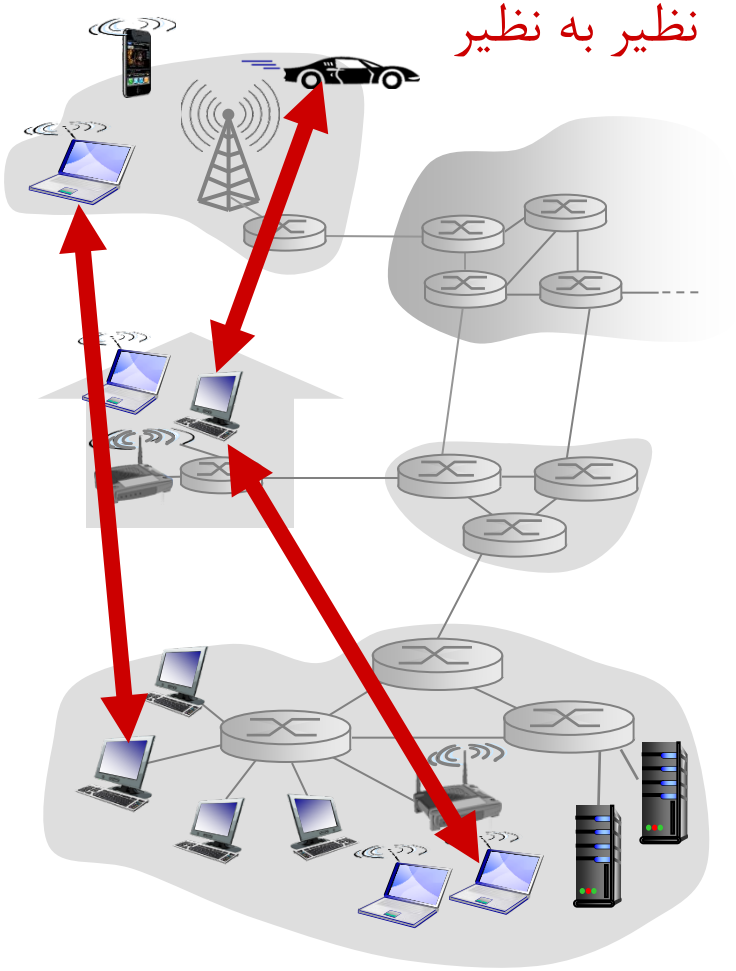
مشتری‌ها:

- با سرویس‌دهنده ارتباط برقرار می‌کنند
- ممکن است که آدرس IP پویا داشته باشند
- به طور مستقیم با یکدیگر ارتباط برقرار نمی‌کنند

معماری P2P

- سرویس‌دهنده همیشه روشن وجود ندارد
- *سیستم‌های انتهایی به طور مستقیم با هم ارتباط برقرار می‌کنند* (درخواست سرویس و ارائه سرویس)
  - *خودمقیاس‌پذیری – کاربران جدید از یک طرف ظرفیت سرویس را افزایش می‌دهند و از طرف دیگر درخواست سرویس جدید دارند*
- از نظر هزینه مقرون به صرفه است
- عدم هماهنگی با سیاست‌های ISP، امنیت، انگیزش‌ها

نظیر به نظیر



ارتباط بین فرآیندها

فرآیند: *برنامه‌ای که روی یک سیستم انتهایی در حال اجراست*

- روی یک سیستم، دو فرآیند از طریق ارتباط بین‌فرآیندی با یکدیگر ارتباط برقرار می‌کنند (تحت کنترل OS)
- فرآیندهای روی سیستم‌های انتهایی مختلف از طریق تبادل پیام روی شبکه با هم ارتباط برقرار می‌کنند

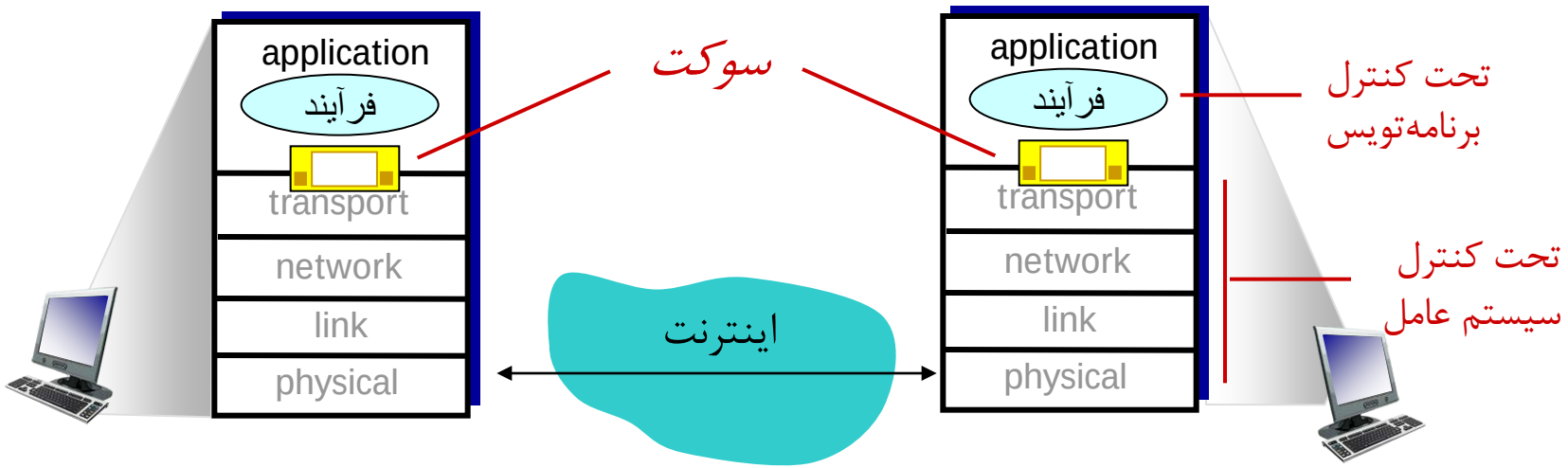
فرآیند مشتری: فرآیندی که ارتباط را آغاز می‌کند

فرآیند سرویس‌دهنده: فرآیندی که منتظر می‌ماند تا برای شروع ارتباط با آن تماس بگیرند

- در معماری‌های P2P، برنامه‌ها هم فرآیند مشتری دارند و هم فرآیند سرویس‌دهنده

سوکت‌ها

- یک فرآیند پیام‌ها را از طریق سوکت ارسال/دریافت می‌کند.
- اگر فرآیند را شبیه یک خانه بدانیم، سوکت مانند درب آن خانه است.



آدرس‌دهی فرآیندها

- برای دریافت پیام‌ها، فرآیند باید شناسه داشته باشد
- در اینترنت هر میزبانی دارای یک آدرس IP ۳۲ بیتی منحصربه‌فردی است
- *سوال:* آیا آدرس IP برای شناسایی فرآیندی که روی یک میزبان اجرا می‌شود، کافی است؟

  - *پاسخ:* خیر، چون ممکن است بیش از یک فرآیند روی یک میزبان اجرا شوند.

- برای شناسایی یک فرآیند دو چیز باید مشخص باشند: آدرس IP و شماره‌ی پورت مرتبط با فرآیند
- مثال (شماره‌ی پورت‌های شناخته شده):
  - سرویس‌دهنده وب: ۸۰
  - سرویس‌دهنده ایمیل: ۲۵
- برای ارسال پیام وب به سرویس‌دهنده وب gaia.cs.umass.edu
  - IP address: 128.119.245.12
  - port number: 80

یک برنامه کاربردی به چه سرویس‌های انتقالی نیاز دارد؟

انتقال داده‌ی قابل اطمینان

- بسیاری از کاربردها (مانند انتقال فایل، تراکنش‌های وب، ایمیل) به انتقال داده قابل اطمینان صددرصدی نیاز دارند.
- برخی از برنامه‌های کاربردهای (مانند کنفرانس صوتی-تصویری) تلفات‌پذیر هستند.

تنظیم وقت (timing)

- برخی از برنامه‌های کاربردی (مانند تلفن اینترنتی، محیط‌های مجازی، بازی‌های چندنفره) برای موثر بودن به تاخیر پایین نیاز دارند.

گذردهی

- برنامه‌های کاربردی مانند برنامه‌های چندرسانه‌ای به گذردهی تضمینی با یک نرخ مشخص نیاز دارند (برنامه‌های حساس به پهنای باند)
- برنامه‌های کشسان (elastic) به پهنای باند حساس نیستند مانند ایمیل وانتقال فایل

امنیت

- رمزگذاری، صحت داده، احراز هویت نقطه انتهایی

ملزومات سرویس انتقال چند برنامه رایج شبکه

application	data loss	throughput	time sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5kbps-1Mbps video:10kbps-5Mbps	yes, 100’s msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few kbps up	yes, 100’s msec
text messaging	no loss	elastic	yes and no

پروتکل‌های انتقال در اینترنت

سرویس‌های TCP:

- انتقال داده قابل اطمینان: همه داده‌ها بدون خطا و با ترتیب صحیح به مقصد می‌رساند
- کنترل جریان: در صورت وجود ازدحام، فرآیند فرستنده را مجبور می‌کند تا بسته‌های خود را با سرعت کمتری ارسال کند
- تنظیم وقت ، تضمین گذردهی و امنیت را فراهم نمی‌کند
- سرویس اتصال‌گرا: بین فرآیندهای مشتری و سرویس‌دهنده تنظیمات اولیه لازم است

سرویس‌های UDP:

- انتقال داده غیرقابل اطمینان: ضمانتی برای تحویل داده‌ها و ترتیب آنها ارائه نمی‌دهد
- کنترل جریان، تنظیم وقت، گذردهی تضمین شده و امنیت را فراهم نمی‌کند
- سرویس غیراتصال‌گرا

سوال: کجا می‌توان از سرویس UDP استفاده کرد؟

مشخصات برنامه‌های معروف اینترنت

application	application layer protocol	underlying transport protocol
e-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),	TCP or UDP
Internet telephony	proprietary (e.g., Skype)	TCP or UDP

امن کردن TCP

TCP & UDP

- هیچ‌گونه رمزنگاری ارائه نمی‌کنند.
- فرآیند فرستنده یک گذرواژه را به صورت متن آشکار به سوکت خود ارسال می‌کند و به همان شکل اینترنت را می‌پیماید.

SSL (لایه سوکت‌های امن)

- سرویس‌های امنیتی کلیدی مانند رمزنگاری، صحت داده و احراز هویت نقطه انتهایی را فراهم می‌کند
- SSL یک پروتکل انتقال اینترنتی مستقل نیست، بلکه یک پیوست تکمیل کننده TCP است که در لایه کاربرد پیاده‌سازی می‌شود

فصل دوم: راهنمای مسیر

- اصول برنامه‌های کاربردی شبکه
- وب و HTTP
- انتقال فایل: FTP
- پست الکترونیک: SMTP
- سرویس دایرکتوری اینترنت: DNS
- برنامه‌های کاربردی نظیر به نظیر: P2P
- برنامه‌نویسی سوکت: ایجاد برنامه‌های کاربردی شبکه

وب و HTTP

مروری بر HTTP

- یک صفحه وب (سند وب) از اشیاء مختلفی تشکیل می‌شود
  - مانند فایل HTML، تصویر JPEG، اپلت جاوا، فایل صوتی و ...
- اصلی‌ترین بخش از یک صفحه‌ی وب یک فایل HTML پایه است که در آن به اشیاء دیگر ارجاع می‌شود.
- ارجاع به هر شیء توسط URL آن صورت می‌گیرد. مانند،

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

نام میزبان — مسیر شیء

مروری بر HTTP

پروتکل انتقال ابرمتن HyperText Transfer Protocol (HTTP)

- پروتکل لایه کاربرد برای وب
- مدل مشتری/سرویس‌دهنده
  - *مشتری: مرورگری که درخواست می‌کند، دریافت می‌کند، و اشیاء وب را نمایش می‌دهد*
  - *سرویس‌دهنده: سرویس‌دهنده وب در پاسخ به درخواست‌ها اشیاء را برای مشتری می‌فرستد*

مروری بر HTTP (ادامه)

HTTP از TCP به عنوان پروتکل انتقال استفاده می‌کند:

- مشتری یک اتصال TCP با سرویس‌دهنده برقرار می‌کند (پورت ۸۰)
- سرویس‌دهنده اتصال TCP را از مشتری می‌پذیرد
- فرآیندهای مرورگر و سرویس‌دهنده از طریق سوکت‌های خود با یکدیگر حرف می‌زنند.
- اتصال TCP بسته می‌شود.

HTTP یک پروتکل بدون حالت (stateless) است

- سرویس‌دهنده هیچ‌گونه اطلاعاتی درباره درخواست‌های گذشته مشتری نگه نمی‌دارد

اتصال‌های TCP

HTTP با اتصال‌های غیرپایا

- حداکثر یک شیء روی هر اتصال TCP فرستاده می‌شود. پس از آن، اتصال بسته می‌شود.
- برای دریافت یک صفحه وب متشکل از ۱۱ شیء بایستی ۱۱ اتصال TCP ایجاد شود.

HTTP با اتصال‌های پایا

- چندین اشیاء می‌توانند روی یک اتصال TCP بین مشتری و سرویس‌دهنده ارسال شوند.

HTTP با اتصال‌های غیرپایا: زمان پاسخ

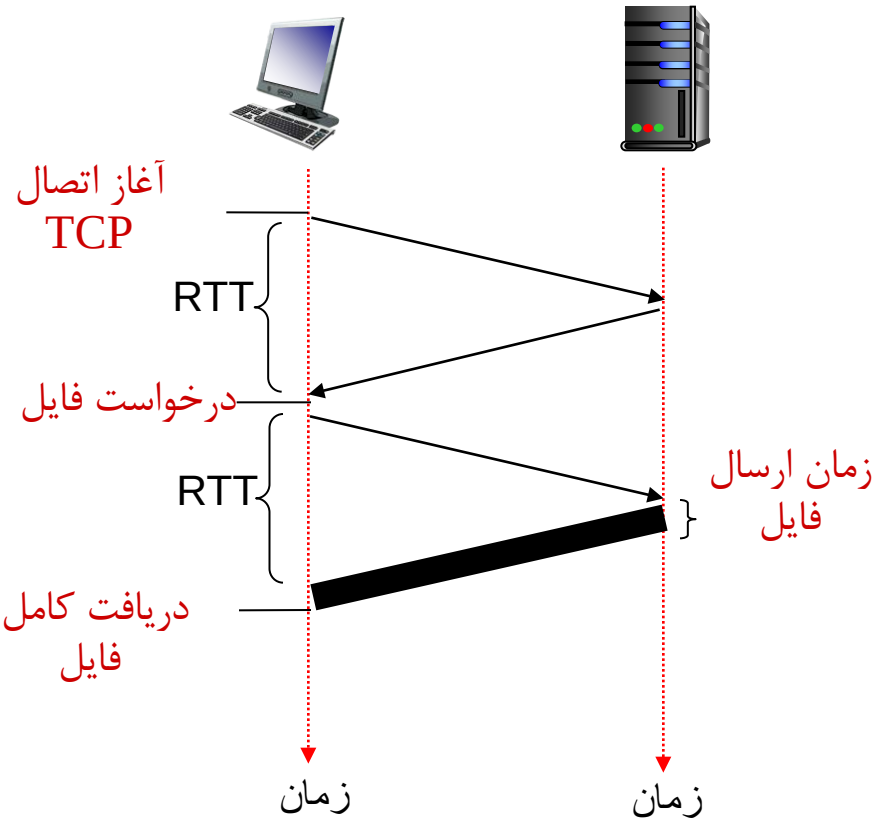
تعریف زمان رفت-برگشت RTT (Round-Trip Time)

زمان طی شده برای سفر یک بسته کوچک از مشتری به سرویس‌دهنده و بازگشت آن از سرویس‌دهنده به مشتری

زمان پاسخ HTTP:

- یک RTT برای آغاز اتصال TCP
- یک RTT برای درخواست HTTP و چند بایت اولیه از پاسخ HTTP برای برگشت
- زمان ارسال فایل
- زمان پاسخ HTTP غیرپایا=

زمان ارسال فایل +2RTT



HTTP با اتصال‌های پایا

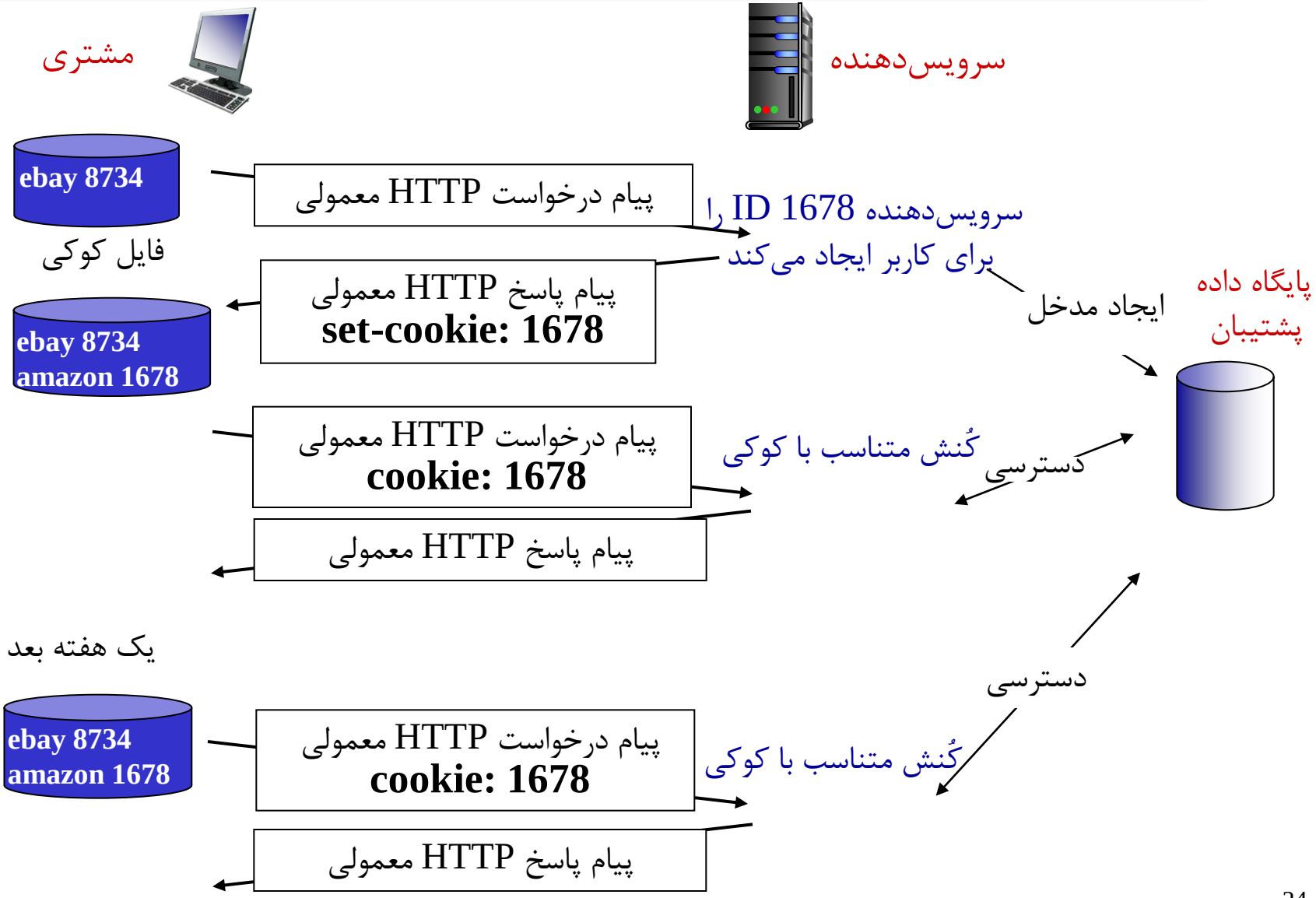
اشکالات اساسی HTTP غیرپایا:

- 2 RTTs به ازای هر شیء لازم دارد
- سربار روی OS برای هر اتصال TCP
- مرورگرها اغلب برای دریافت اشیاء اتصال‌های TCP موازی باز می‌کنند

HTTP پایا:

- سرویس‌دهنده بعد از برگرداندن پاسخ، اتصال TCP را همچنان باز نگه می‌دارد
- درخواست‌ها و پاسخ‌های بعدی بین مشتری و سرویس‌دهنده از طریق همان اتصال صورت می‌گیرد
- نیازی نیست مشتری برای درخواست یک شیء منتظر رسیدن اشیاء قبلی باشد
- سرویس‌دهنده HTTP فقط زمانی یک اتصال TCP را می‌بندد که برای مدتی مشخص هیچ فعالیت روی آن صورت نگیرد

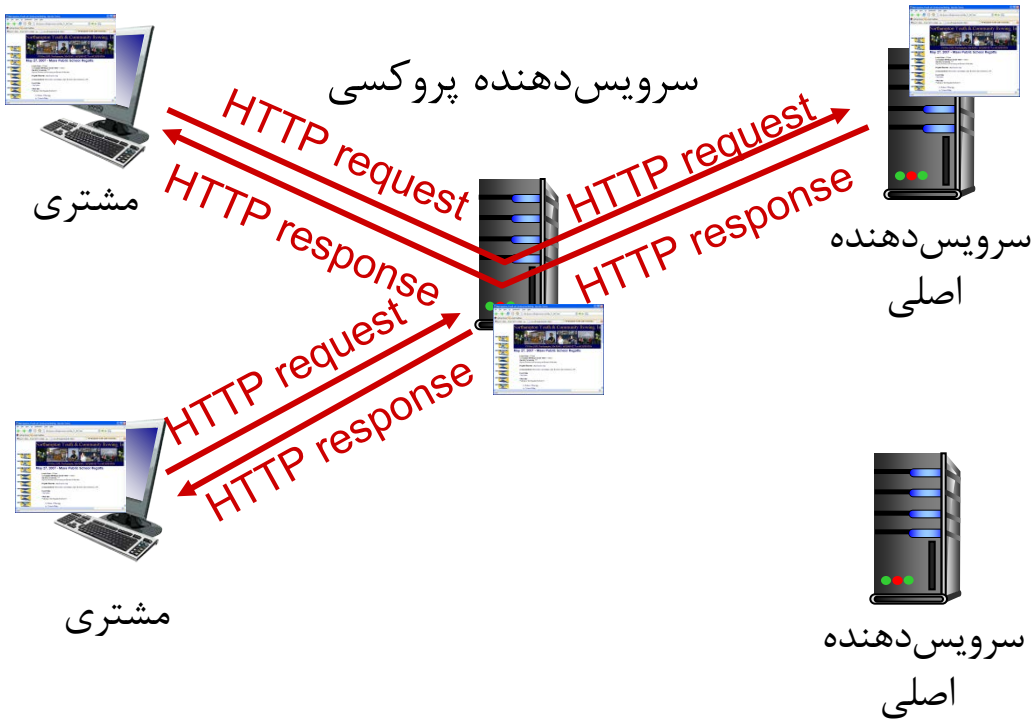
نگهداری اطلاعات حالت کاربر با کوکی



حافظه نهان وب (سرویس‌دهنده پروکسی)

حافظه نهان وب: *موجودیتی روی شبکه که درخواست‌های http را به وکالت از طرف سرویس‌دهنده وب اصلی انجام می‌دهد*

- هر درخواست مرورگر ابتدا به حافظه نهان وب مشخص شده هدایت می‌شود
  - اگر شیء مورد درخواست در در حافظه نهان باشد: حافظه نهان آن را برمی‌گرداند
  - در غیر این صورت، حافظه نهان شیء را از سرویس‌دهنده اصلی درخواست می‌کند، آن را ذخیره کرده و یک نسخه از آن را به مشتری برمی‌گرداند.



حافظه نهان وب (سرویس‌دهنده پروکسی)

- حافظه‌های نهان وب معمولاً توسط ISPها و مراکز بزرگ (مانند سازمان‌ها و موسسات دانشگاهی) نصب و پیاده‌سازی می‌شوند

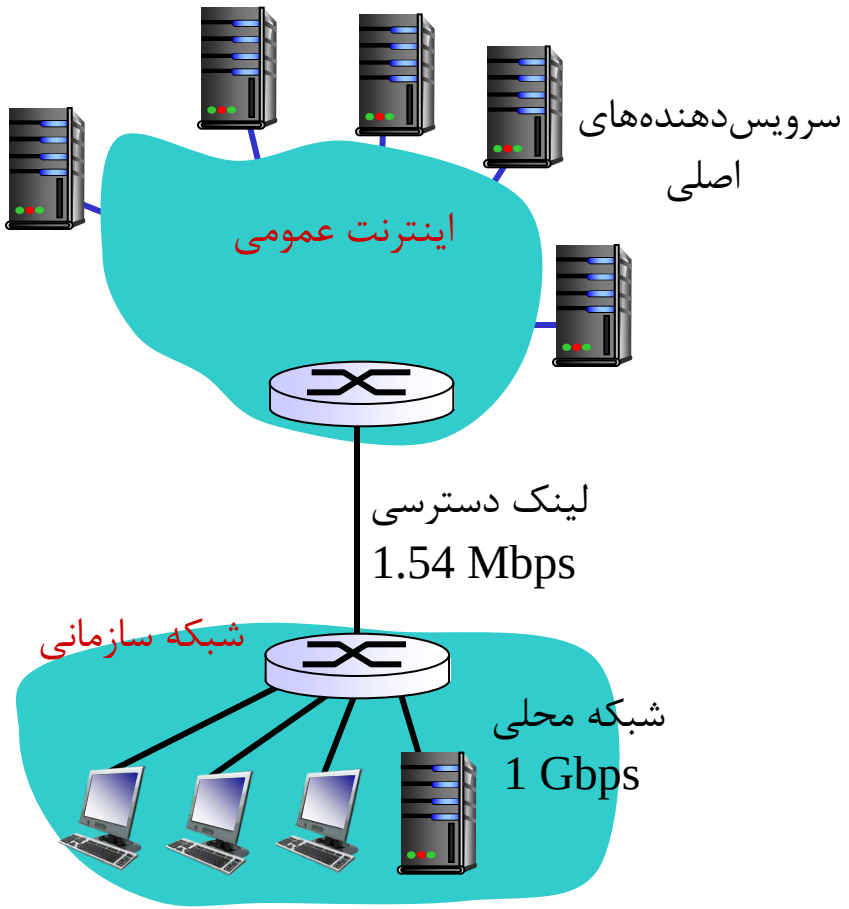
دو دلیل استفاده از حافظه نهان وب

- کاهش زمان پاسخ به مشتری
- کاهش ترافیک لینک‌های دسترسی به اینترنت

مثال

فرضیات:

- متوسط اندازه اشیاء: 100K bits
- متوسط نرخ درخواست از مرورگرها به سرویس‌دهنده‌های اصلی: 15/sec
- متوسط نرخ داده به مرورگرها: 1.5 Mbps
- RTT از مسیریاب سازمانی به هر سرویس‌دهنده‌ی اصلی: 2 sec
- نرخ لینک دسترسی: 1.54 Mbps



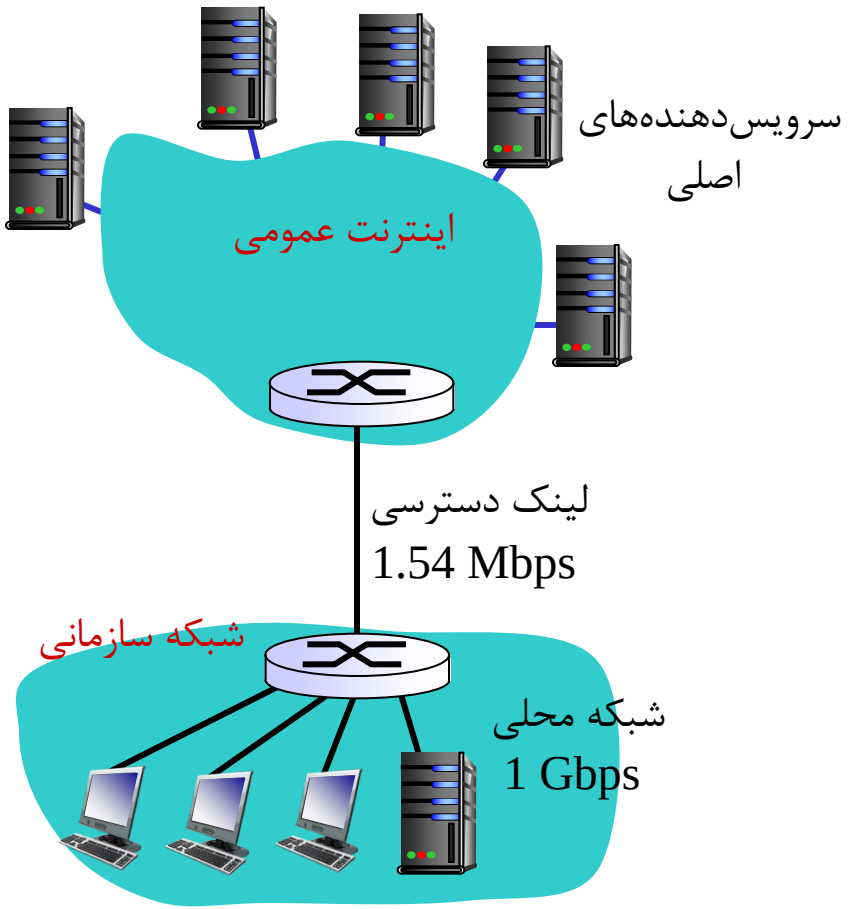
نتایج:

- بهره‌وری شبکه محلی= 15%
- بهره‌وری لینک دسترسی= 99%
- total delay = Internet delay + access delay + LAN delay
  = 2 sec + minutes + usecs

مثال

فرضیات:

- متوسط اندازه اشیاء: 100K bits
- متوسط نرخ درخواست از مرورگرها به سرویس‌دهنده‌های اصلی: 15/sec
- متوسط نرخ داده به مرورگرها: 1.5 Mbps
- RTT از مسیریاب سازمانی به هر سرویس‌دهنده‌ی اصلی: 2 sec
- نرخ لینک دسترسی: ~~1.54 Mbps~~ 154 Mbps



نتایج:

- بهره‌وری شبکه محلی= 15%
- بهره‌وری لینک دسترسی= ~~99%~~ 9.9%
- total delay = Internet delay + access delay + LAN delay
  = 2 sec + ~~minutes~~ + usecs msecs

این مستلزم صرف هزینه هنگفتی است

مثال: نصب حافظه نهان وب

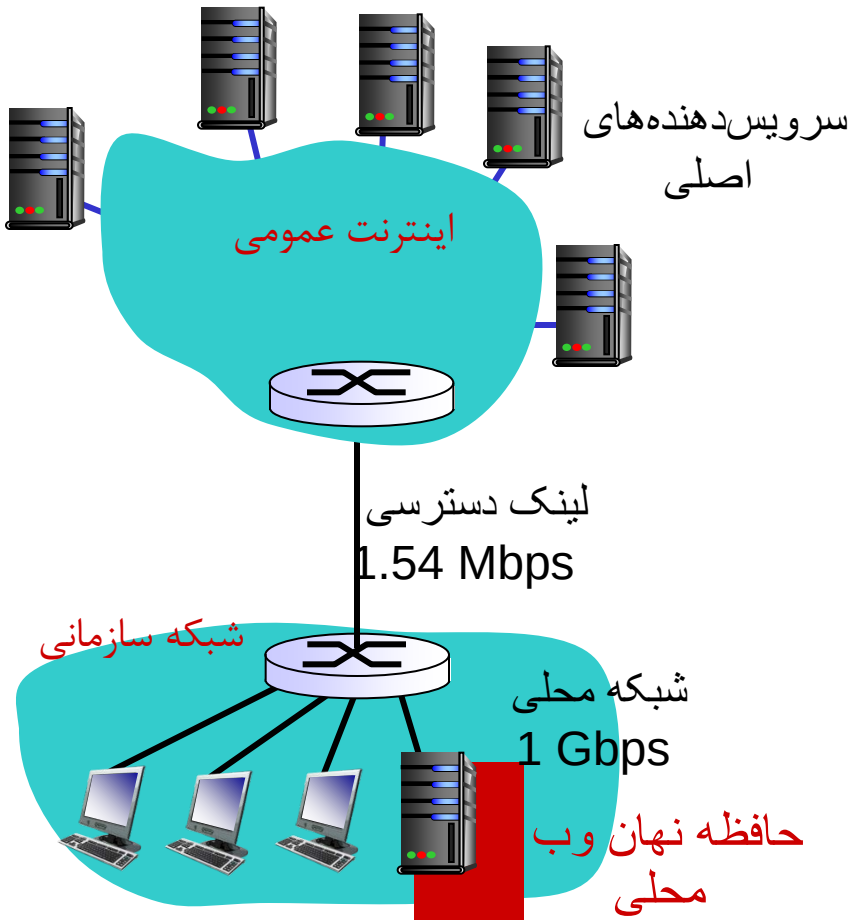
فرضیات:

- متوسط اندازه اشیاء: 100K bits
- متوسط نرخ درخواست از مرورگرها به سرویس‌دهنده‌های اصلی: 15/sec
- متوسط نرخ داده به مرورگرها: 1.5 Mbps
- RTT از مسیریاب سازمانی به هر سرویس‌دهنده‌ی اصلی: 2 sec
- نرخ لینک دسترسی: 1.54 Mbps

نتایج:

- بهره‌وری شبکه محلی= 15%
- بهره‌وری لینک دسترسی= ؟
- تاخیر کل=؟

هزینه نصب حافظه نهان نسبتاً ناچیز است



مثال: نصب حافظه نهان وب

محاسبه بهره‌وری لینک دسترسی و تاخیر کل:

- فرض کنید ضریب اصابت حافظه نهان وب ۰.۴ باشد
  - ۴۰٪ درخواست‌ها توسط حافظه نهان پاسخ داده می‌شوند، ۶۰٪ هم توسط سرویس‌دهنده‌های اصلی
- بهره‌وری لینک دسترسی:

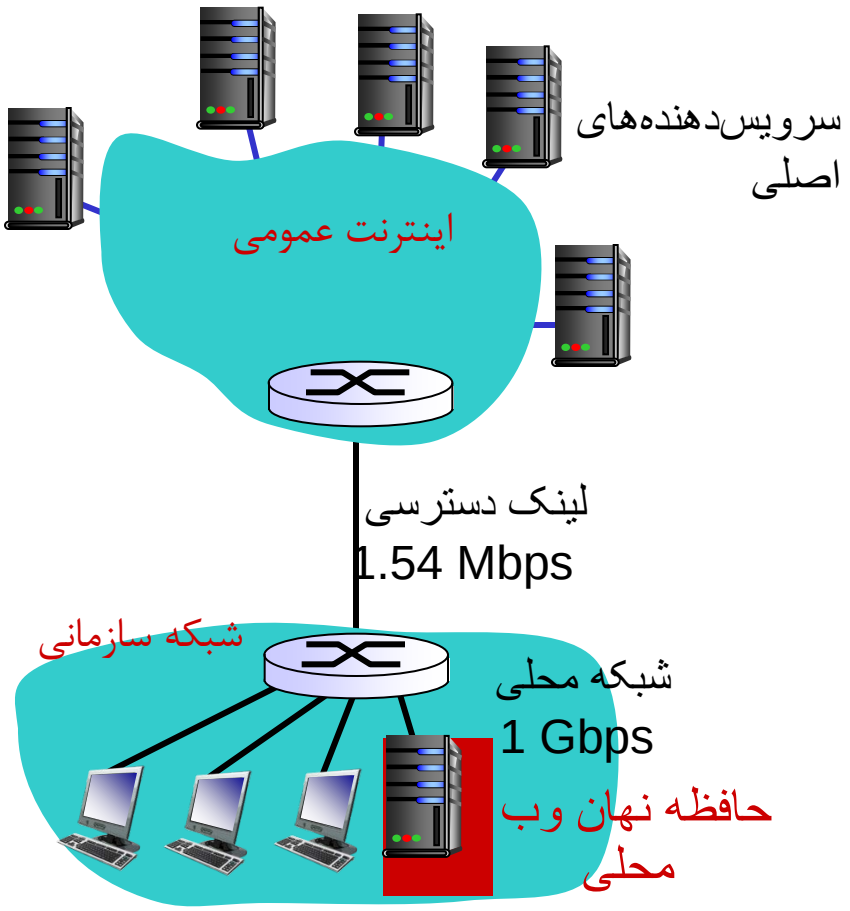
  = 0.6*1.50 Mbps =0 .9 Mbps
  - بهره‌وری

  0.9/1.54 = 0.58
- تاخیر کل:

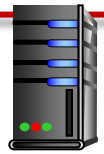
  = 0.6 * (تاخیر ناشی از سرویس‌دهنده‌های اصلی)
  +0.4 * (تاخیر ناشی از حافظه نهان)

  = 0.6 (2.01) + 0.4 (~msecs)

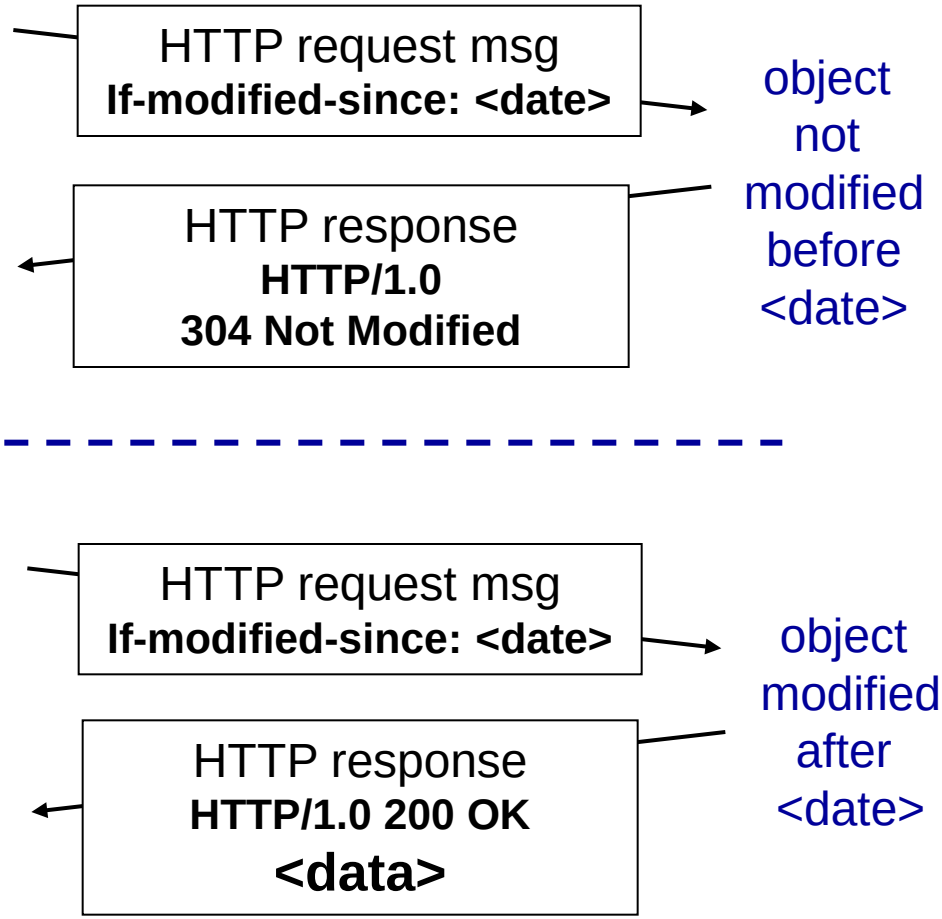
  = ~ 1.2 secs



فرمان GET شرطی



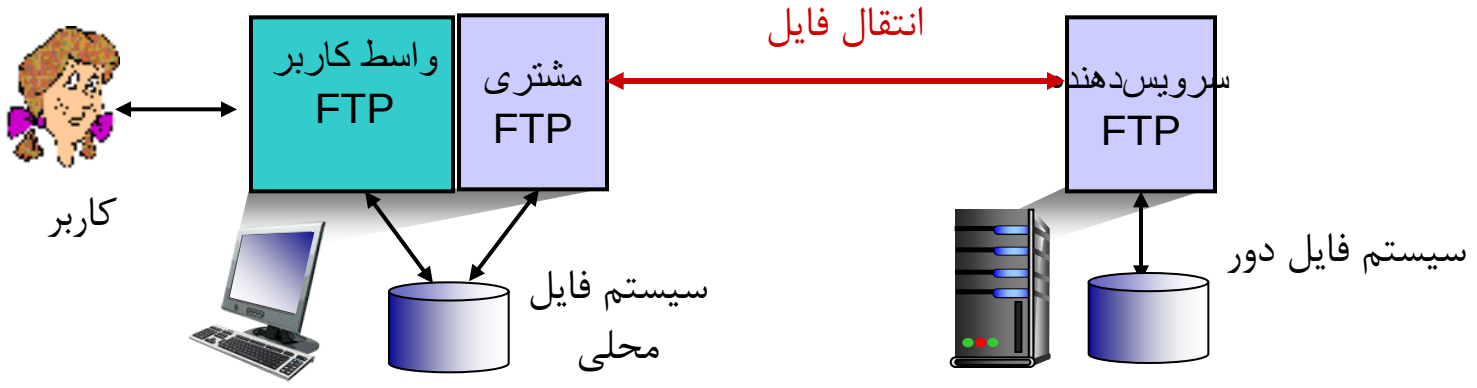
- *هدف: اگر حافظه نهان به روز است شیء را ارسال نکن.*
  - تاخیر ارسال شیء را نخواهیم داشت
  - کاهش بهره‌وری لینک



فصل دوم: راهنمای مسیر

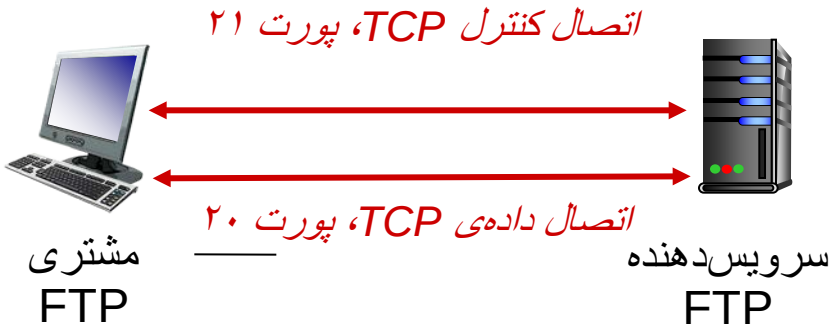
- اصول برنامه‌های کاربردی شبکه
- وب و HTTP
- انتقال فایل: FTP
- پست الکترونیک: SMTP
- سرویس دایرکتوری اینترنت: DNS
- برنامه‌های کاربردی نظیر به نظیر: P2P
- برنامه‌نویسی سوکت: ایجاد برنامه‌های کاربردی شبکه

پروتکل انتقال فایل (FTP)



- انتقال فایل از/به میزبان دور
- مدل مشتری-سرویس‌دهنده
  - *مشتری: طرفی که انتقال را آغاز می‌کند*
  - *سرویس‌دهنده: میزبان دور*
- سرویس‌دهنده ftp: پورت ۲۱

FTP: اتصال کنترل و اتصال داده‌ی جداگانه



- اتصال کنترل: "out-of-band"
- سرویس‌دهنده FTP حالت کاربر را نگه می‌دارد: دایرکتوری جاری، احراز هویت قبلی

- مشتری FTP از طریق TCP روی پور ۲۱ با سرویس‌دهنده FTP تماس می‌گیرد.
- مشتری روی اتصال کنترل احراز هویت می‌شود.
- مشتری روی همین اتصال کنترل فرمان‌ها را ارسال می‌کند.
- وقتی سرویس‌دهنده فرمان انتقال فایل را دریافت کرد، یک اتصال داده‌ی TCP روی پورت ۲۰ به مشتری FTP ایجاد می‌کند.
- بعد از ارسال یک فایل، سرویس‌دهنده اتصال داده را می‌بندد. برای انتقال یک فایل دیگر، سرویس‌دهنده دوباره یک اتصال داده جدید باز می‌کند.

فصل دوم: راهنمای مسیر

- اصول برنامه‌های کاربردی شبکه
- وب و HTTP
- انتقال فایل: FTP
- پست الکترونیک: SMTP
- سرویس دایرکتوری اینترنت: DNS
- برنامه‌های کاربردی نظیر به نظیر: P2P
- برنامه‌نویسی سوکت: ایجاد برنامه‌های کاربردی شبکه

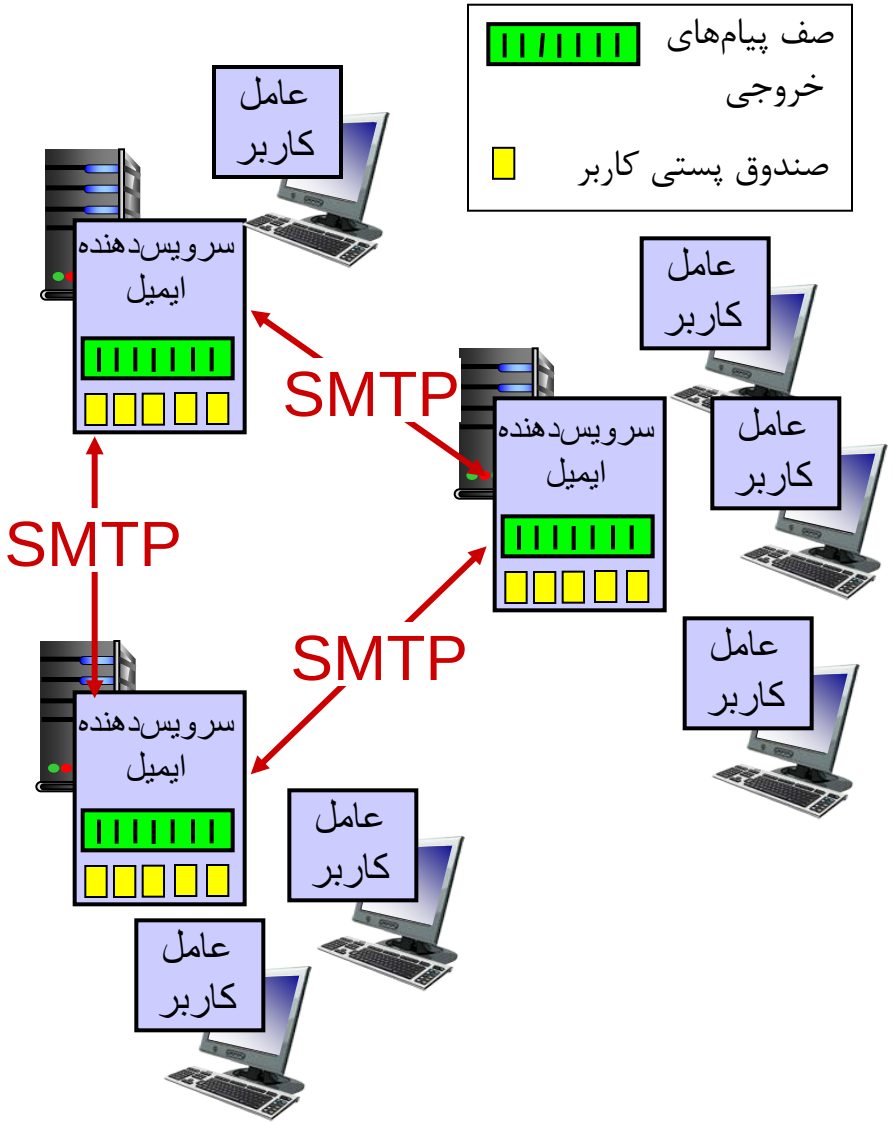
پست الکترونیک

سه جزء اصلی:

- عامل کاربر
- سرویس‌دهنده ایمیل
- پروتکل ساده‌ی انتقال ایمیل (simple mail transfer protocol: SMTP)

عامل کاربر

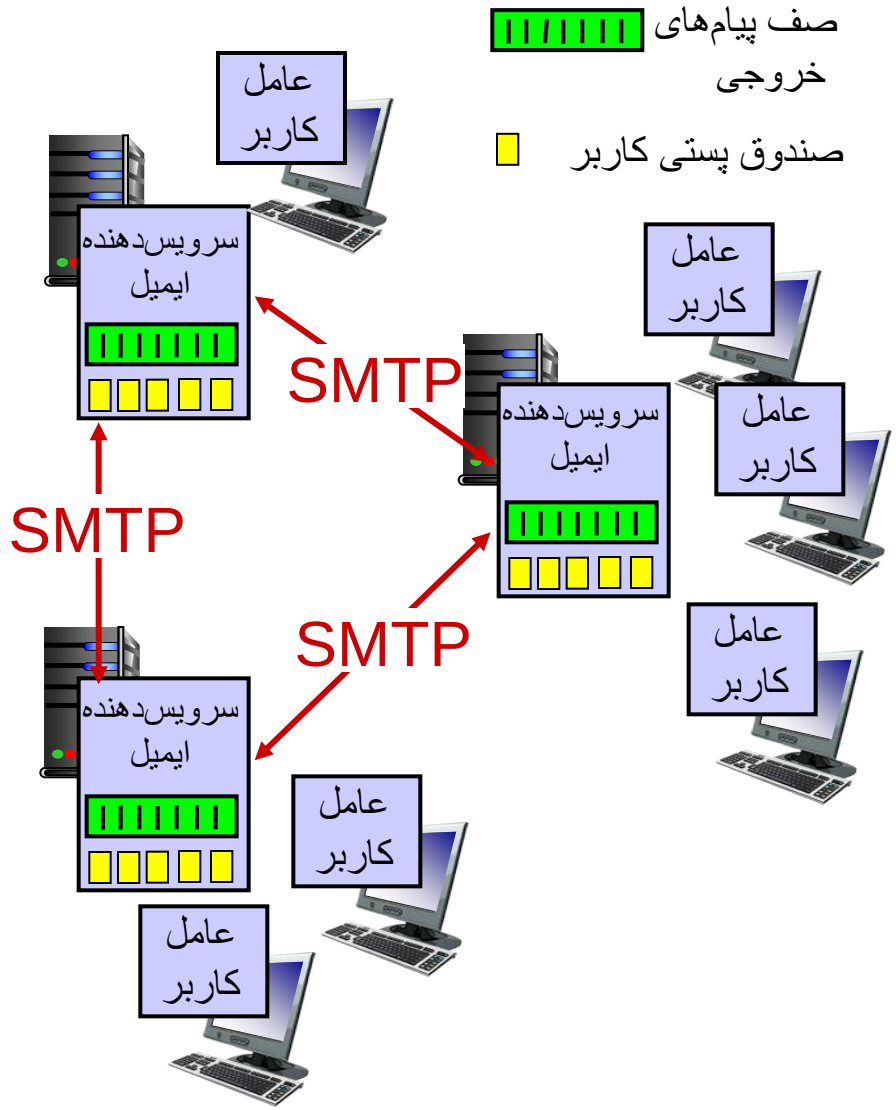
- نوشتن، ویرایش، خواندن پیام‌های ایمیل
- مثال، آوت‌لوک، تاندربرد، انتوراژ
- پیام‌های ارسالی/دریافتی روی سرویس‌دهنده ذخیره می‌شوند



پست الکترونیک: سرویس‌دهنده‌های ایمیل

سرویس‌دهنده‌های ایمیل:

- *صندوق پستی: پیام‌های دریافت شده کاربر را مدیریت و نگهداری می‌کند*
- *صف پیام: برای نگهداری پیام‌هایی که باید ارسال شوند*
- *پروتکل SMTP: برای ارسال پیام‌های ایمیل بین سرویس‌دهنده‌های ایمیل*
  - مشتری: ارسال به سرویس‌دهنده ایمیل
  - سرویس‌دهنده: دریافت از سرویس‌دهنده



پروتکل SMTP

- از TCP استفاده می‌کند
- شماره پورت ۲۵
- انتقال مستقیم: سرویس‌دهنده ارسال به سرویس‌دهنده دریافت
- استفاده از اتصال پایا
- سه مرحله انتقال
  - دستداد
  - انتقال پیام‌ها
  - بستن
- پیام‌ها باید به صورت اسکی ۷-بیتی کدگذاری شوند

سناریو: ارسال یک پیام از آلیس به باب

۱) آلیس با استفاده از عامل کاربر خود پیامی به bob@someschool.edu می‌نویسید

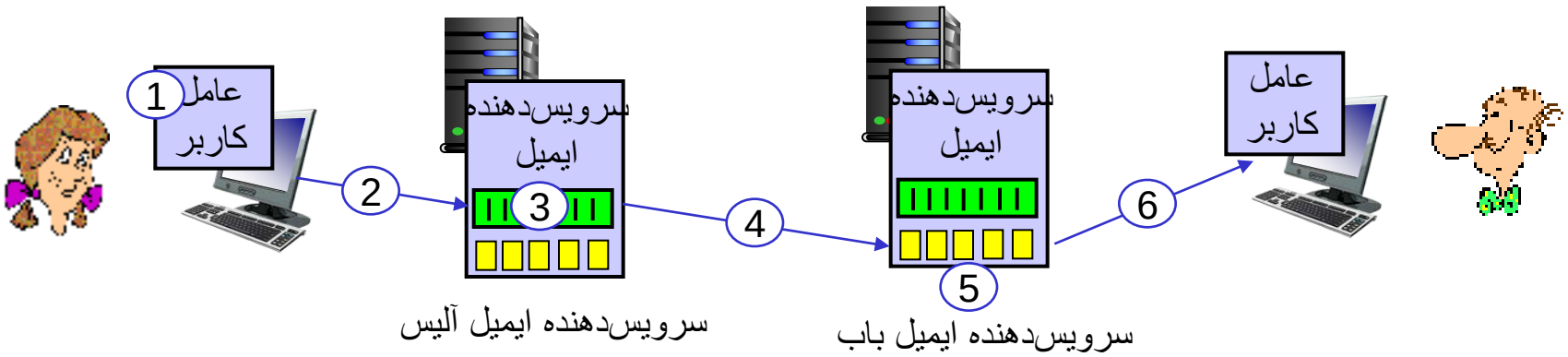
۲) عامل کاربر آلیس پیام را به سرویس‌دهنده‌ی پست او می‌فرستد، که در آنجا در یک صف پیام قرار داده می‌شود

۳) SMTP سمت مشتری (آلیس) یک اتصال TCP با سرویس‌دهنده ایمیل باب باز می‌کند

۴) SMTP مشتری پیام آلیس را روی اتصال TCP می‌فرستد

۵) سرویس‌دهنده ایمیل باب پیام را در صندوق پستی او قرار می‌دهد

۶) در زمان دلخواه، باب عامل کاربر خود را باز می‌کند و پیام آلیس را می‌خواند



فصل دوم: راهنمای مسیر

- اصول برنامه‌های کاربردی شبکه
- وب و HTTP
- انتقال فایل: FTP
- پست الکترونیک: SMTP
- سرویس دایرکتوری اینترنت: DNS
- برنامه‌های کاربردی نظیر به نظیر: P2P
- برنامه‌نویسی سوکت: ایجاد برنامه‌های کاربردی شبکه

سیستم نام دامنه (domain name system) DNS

- افراد شناسه‌های مختلفی دارند:
  - نام
  - شماره ملی
- میزبان‌ها و مسیریاب‌های اینترنت:
  - نام
  - آدرس IP

سوال: چگونه نام را به آدرس IP (و برعکس) نگاشت کنیم؟

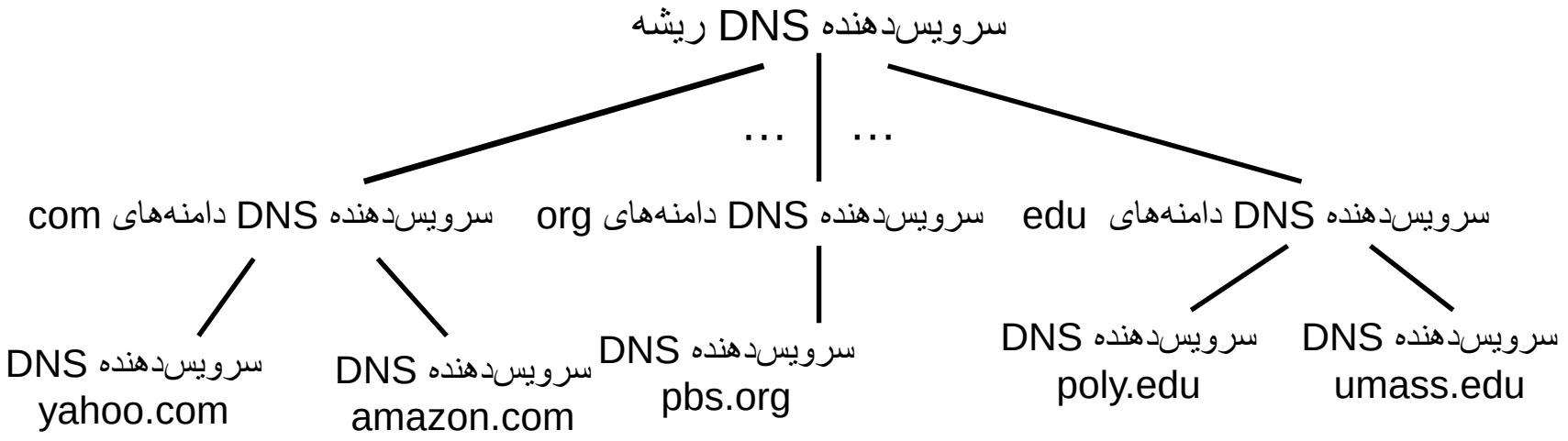
DNS:

- پایگاه داده توزیع شده: ساختاری سلسله مراتبی از سرویس‌دهنده‌های DNS
- پروتکل لایه کاربرد: برای پرس و جو در پایگاه داده توزیع شده

سه طبقه از سرویس‌دهنده‌های DNS

- سرویس‌دهنده‌های DNS ریشه
  - در اینترنت ۱۳ سرویس‌دهنده وجود دارد (آمار سال ۲۰۱۲)
- سرویس‌دهنده‌های DNS دامنه سطح-بالا (TLD (top-level domain
  - مسئول دامنه‌های سطح بالا مانند com، org، net، edu، gov و همچنین تمام دامنه‌های سطح بالای کشوری مانند uk، jp، ir هستند
- سرویس‌دهنده‌های DNS سرپرستی
  - سازمان‌ها می‌توانند یک سرویس‌دهنده‌ی DNS سرپرستی اختصاصی پیاده‌سازی کنند.

DNS: یک پایگاه داده‌ی سلسه مراتبی توزیع شده



فرض کنید یک مشتری آدرس IP نام میزبان www.amazon.com را بخواهد:

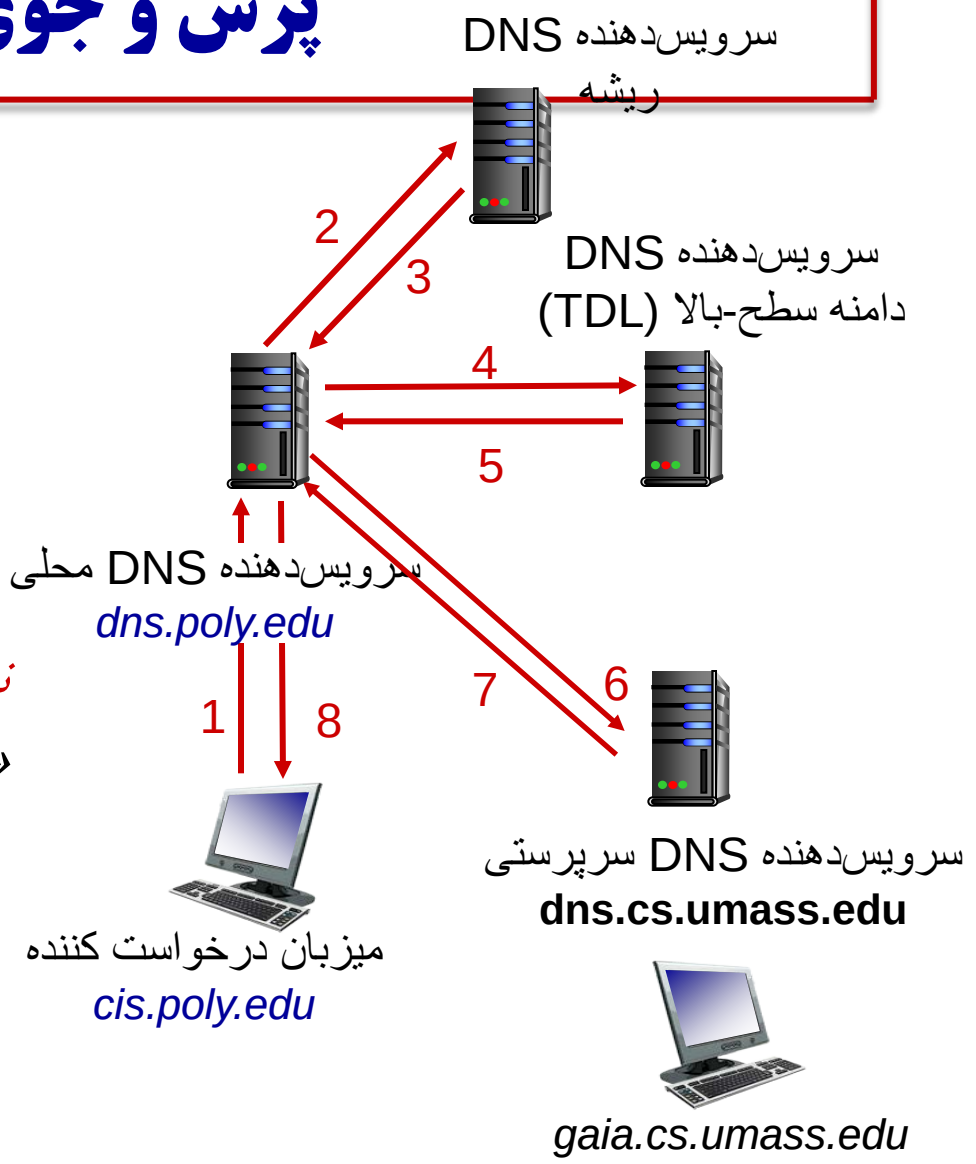
- مشتری از ریشه پرس و جو می‌کند تا سرویس‌دهنده DNS دامنه‌های com را پیدا کند
- مشتری از سرویس‌دهنده دامنه‌های com پرس و جو می‌کند تا سرویس‌دهنده DNS نام amazon.com را بدست آورد
- در پایان، مشتری با یکی از سرویس‌دهنده‌های DNS دامنه amazon.com تماس می‌گیرد، این سرویس‌دهنده آدرس (یا آدرس‌های) IP میزبان amazon.com را به آن برمی‌گرداند.

سرویس‌دهنده‌ی DNS محلی

- هر ISP، مانند دانشگاه‌ها، سازمان‌ها و شرکت‌های بزرگ، و همچنین ISPهای محلی دارای یک سرویس‌دهنده DNS محلی هستند
  - به آن سرویس‌دهنده‌ی نام پیش‌فرض نیز گفته می‌شود
- وقتی میزبان یک پرس و جوی DNS انجام می‌دهد، این پرس و جو (درخواست) به سرویس‌دهنده DNS محلی‌اش فرستاده می‌شود
  - سرویس‌دهنده DNS محلی در نقش یک پروکسی ظاهر می‌شود

پرس و جوی تکراری در DNS

- میزبان cis.poly.edu می‌خواهد آدرس IP میزبان gaia.cs.umass.edu را بداند



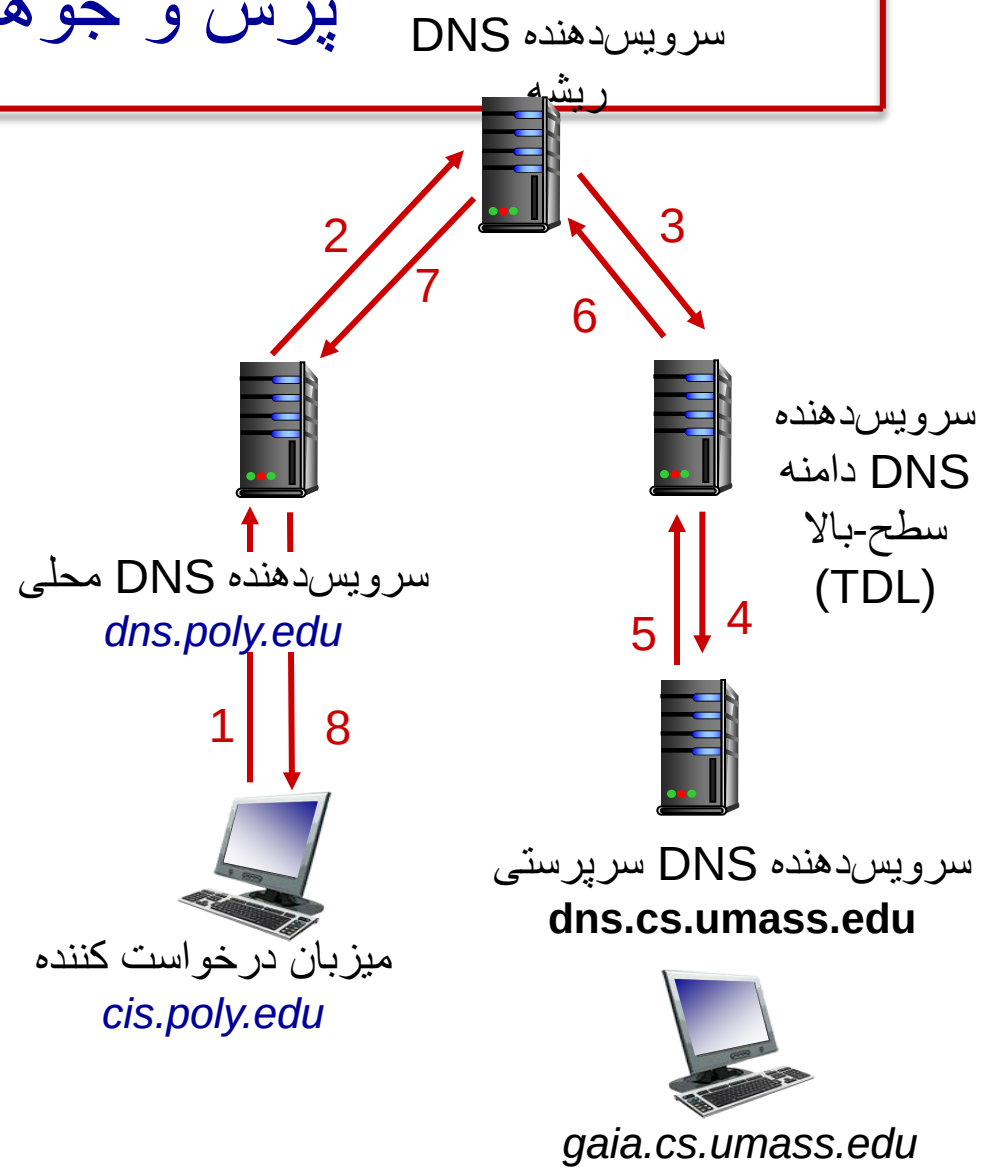
نحوه راهنمایی سرویس‌دهنده‌های DNS:

«من این نام را نمی‌شناسم، اما می‌توانید از این سرویس‌دهنده بپرسید»

پرس و جوهای بازگشتی در DNS

پرس و جوی بازگشتی:

- مسئولیت پرسیدن نام بر عهده سرویس‌دهنده تماس گرفته‌شده است
- بار سنگین روی سطوح سلسله مراتبی



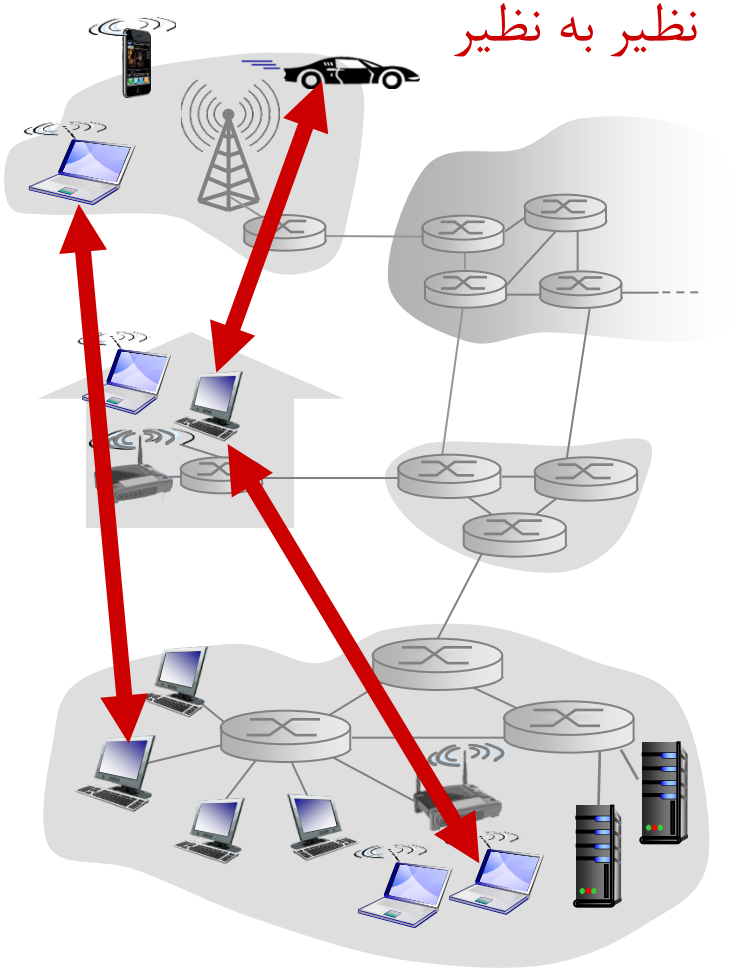
فصل دوم: راهنمای مسیر

- اصول برنامه‌های کاربردی شبکه
- وب و HTTP
- انتقال فایل: FTP
- پست الکترونیک: SMTP
- سرویس دایرکتوری اینترنت: DNS
- برنامه‌های کاربردی نظیر به نظیر: P2P
- برنامه‌نویسی سوکت: ایجاد برنامه‌های کاربردی شبکه

معماری P2P (مرور)

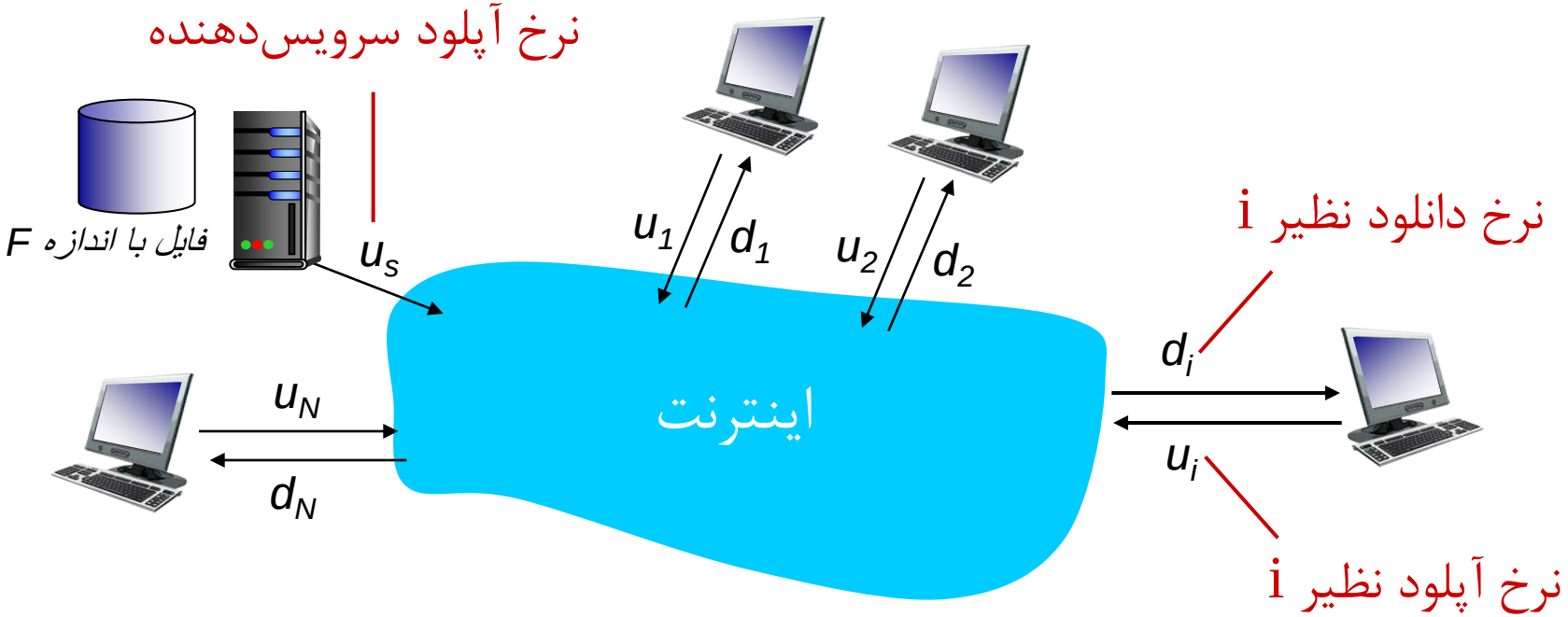
- سرویس‌دهنده همیشه روشن وجود ندارد
- *سیستم‌های انتهایی به طور مستقیم با هم ارتباط برقرار می‌کنند* (درخواست سرویس و ارائه سرویس)
  - *خودمقیاس‌پذیری – کاربران جدید از یک طرف ظرفیت سرویس را افزایش می‌دهند و از طرف دیگر درخواست سرویس جدید دارند*
- از نظر هزینه مقرون به صرفه است
- عدم هماهنگی با سیاست‌های ISP، امنیت، انگیزش‌ها

نظیر به نظیر



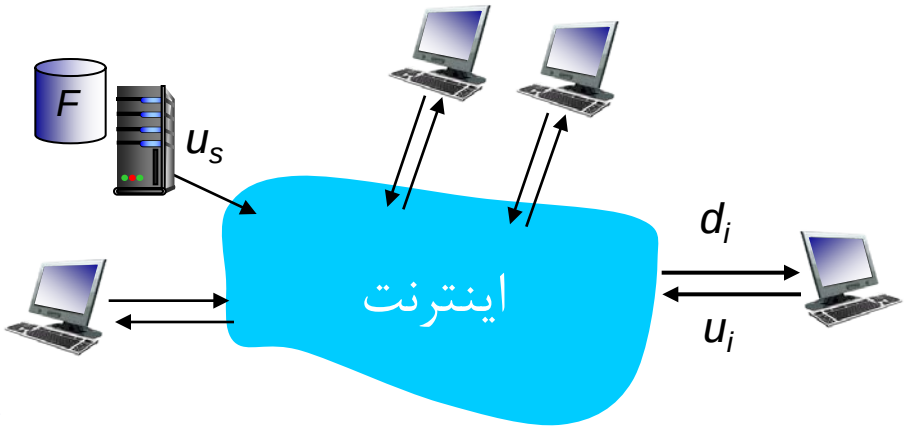
توزیع فایل: مدل مشتری–سرویس‌دهنده در مقابل مدل P2P

سوال: چقدر زمان لازم است تا فایلی با اندازه F (بیت) از یک سرور به N نظیر توزیع شود؟



زمان توزیع فایل: مدل مشتری–سرویس‌دهنده

- *سرویس‌دهنده باید یک کپی از فایل را به هر یک از N نظیر ارسال کند:*
  - زمان ارسال برای یک کپی: F/u_s
  - زمان ارسال برای N کپی: NF/u_s



- *هر مشتری باید یک کپی از فایل را دریافت کند:*
  - کمترین نرخ دانلود مشتری: dmin
  - کمترین زمان دانلود مشتری: F/dmin

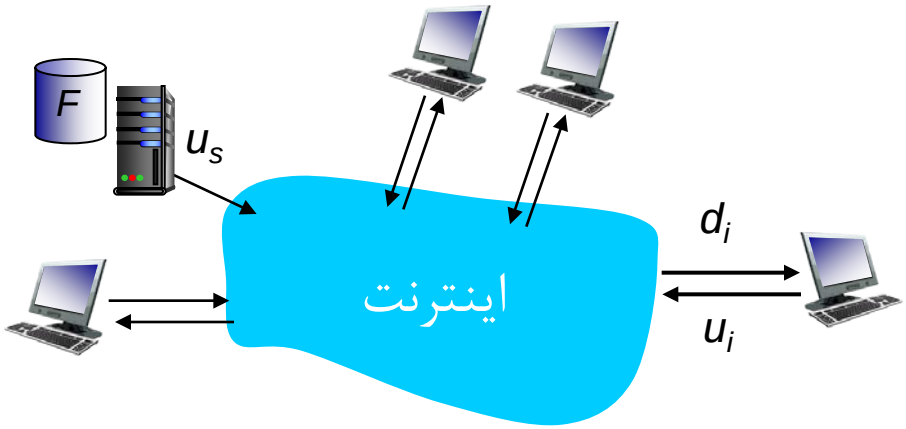
زمان لازم برای توزیع F به N مشتری با استفاده از رویکرد مشتری-سرویس‌دهنده

$$D_{c\text{-}s} \geq max\{NF/u_s, F/d_{min}\}$$

- *با افزایش N به صورت خطی افزایش می‌یابد*

زمان توزیع فایل: مدل P2P

- سرویس‌دهنده حداقل باید یک کپی آپلود کند
  - زمان ارسال یک کپی: F/u_s
- هر مشتری باید یک کپی از فایل را آپلود کند
  - حداقل زمان دانلود مشتری: $F/dmin$



- مشتری‌ها در مجموع باید NF بیت دانلود کنند
  - بیشترین نرخ آپلود: $u_s + \Sigma u_i$

زمان لازم برای توزیع F به N مشتری با استفاده از رویکرد P2P

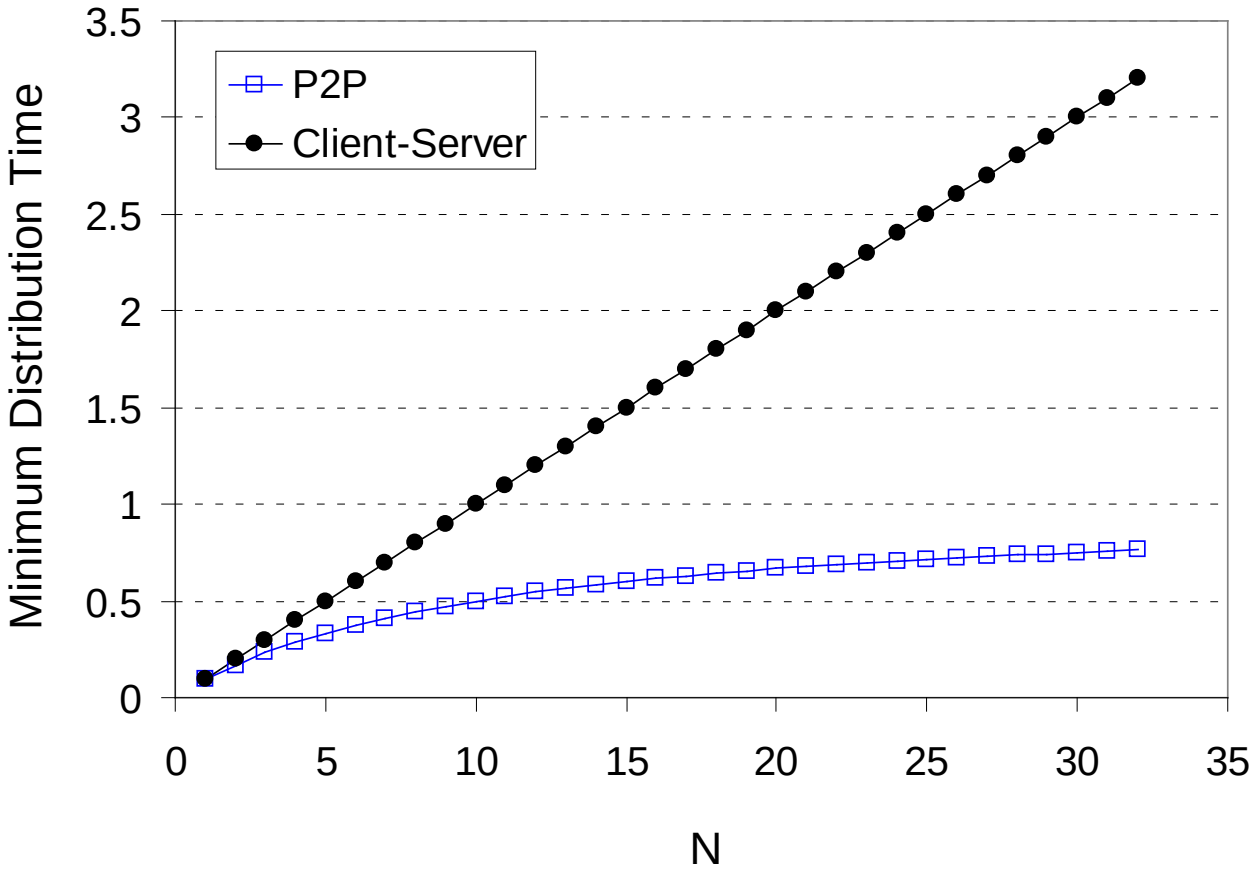
$$D_{P2P} \geq max\{F/u_s, F/d_{min}, NF/(u_s + \Sigma u_i)\}$$

با افزایش N به صورت خطی افزایش می‌یابد

اما این مقدار نیز افزایش پیدا می‌کند (هر نظیر ظرفیت سرویس را افزایش می‌دهد)

مشتری–سرویس‌دهنده در برابر P2P: مثال

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

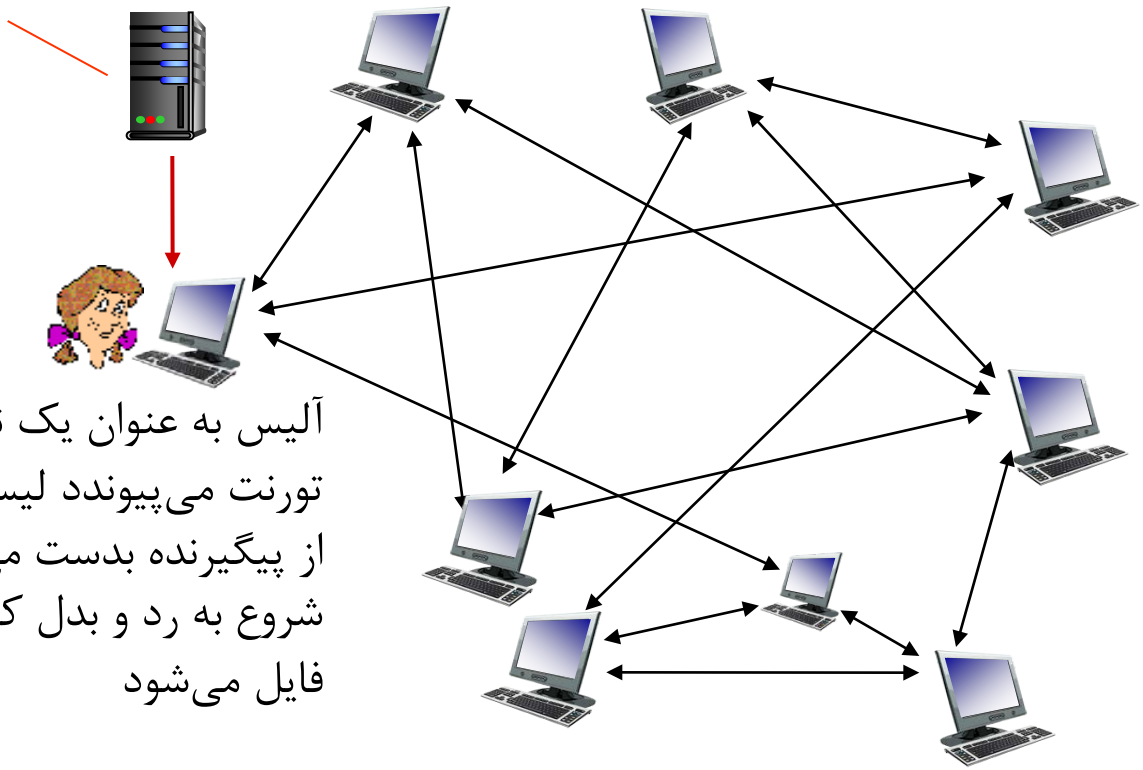


بیت تورنت

- فایل به قطعات 256 KB تقسیم می‌شوند
- نظیرها در تورنت قطعات فایل را ارسال/دریافت می‌کنند

تورنت: گروهی از نظیرها که قطعات یک فایل را مبادله می‌کنند

پیگیرنده: نظیرهای فعال در تورنت را دنبال می‌کند



آلیس به عنوان یک نظیر جدید به تورنت می‌پیوندد لیستی از نظیرها را از پیگیرنده بدست می‌آورد و با آنها شروع به رد و بدل کردن قطعات فایل می‌شود

خلاصه فصل ۲

- معماری برنامه‌های کاربردی
  - مشتری-سرویس‌دهنده
  - نظیر به نظیر (P2P)
- نیازمندی‌های سرویس برنامه‌های کاربردی
  - قابلیت اطمینان، پهنا باند، تاخیر
- مدل سرویس انتقال در اینترنت
  - سرویس اتصال‌گرا، قابلیت اطمینان: TCP
  - سرویس غیراتصال‌گرا، نامطمئن: UDP

- پروتکل‌های لایه کاربرد
  - HTTP
  - FTP
  - SMTP
  - DNS
  - P2P: بیت تورنت