بسم الله الرحمن الرحیم

# مهندسی اینترنت
# Internet Engineering


**دانشگاه پیام نور کـــــرمان**

رشته مهندسی نرم افزار

**مدرس : فرزانه دباغیان**

f.dabaghian20@gmail.com
f.dabaghian@aut.ac.ir


darsamooz.blog.ir

# منبع درس

## مهندسی اینترنت

مؤلف: عباسعلی رضایی
نشر : خط اول

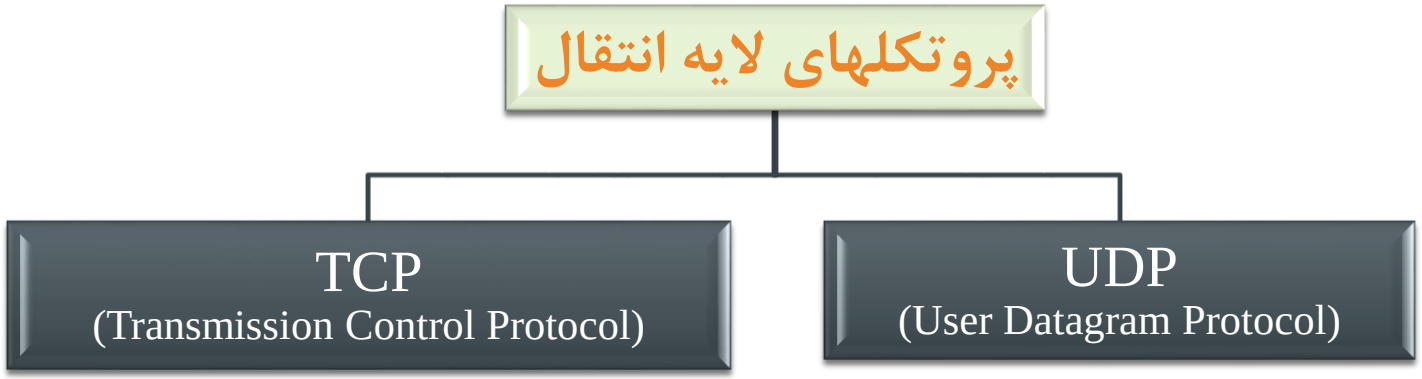
# فصل پنجم

## لایه انتقال در اینترنت

# فهرست موضوعات

# مقدمه

- ✓ لایه‌ی سوم(شبکه) نمی‌تواند مشکلاتی که در طی مسیر ممکن است برای یک بسته‌ی IP اتفاق بیفتد، حل کند.
- ✓ لایه‌ی انتقال که بالای لایه‌ی شبکه قرار دارد مشکلات و ناکارامدی لایه IP را جبران کرده و یک ارتباط انتها به انتها و مطمئن را برقرار می‌کند.



# UDP
## مشخصات

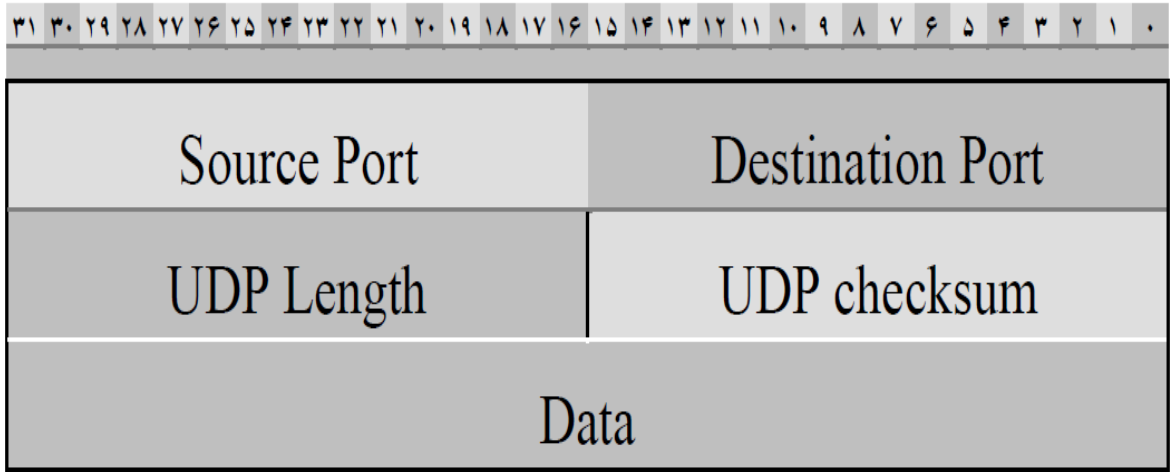
- **بدون اتصال (Connectionless) :**

  **قبل از ارسال هرگونه داده، اقدامی جهت برقراری تماس و هماهنگی بین مبدا و مقصد صورت نمی‌گیرد و بسته ارسال می‌شود.**

- **نامطمئن**
- **بدون کنترل جریان**
- **بدون خطایابی**

# UDP
## قالب بسته

- **Source Port : (شماره پورت مبدأ)**
یک شماره‌ی ۱۶ بیتی به عنوان آدرس پورت پروسه‌ی مبدأ که این بسته را جهت ارسال، تولید کرده است قرار خواهد گرفت.
- **Destination Port : (شماره پورت مقصد)**
یک شماره‌ی ۱۶ بیتی به عنوان آدرس پورت پروسه‌ی مبدأ که این بسته را جهت ارسال، تولید کرده است قرار خواهد گرفت.
- **UDP Length : (طول کلی)** برحسب بایت درج می‌شود.
- **UDP Checksum : (جمع کنترلی)** بکارگیری این فیلد اختیاری است و برای کنترل خطا بکار می‌رود.



# TCP
## مشخصات

- **اتصال‌گرا :**

  **قبل از ارسال هرگونه داده، جهت برقراری تماس و هماهنگی بین مبدا و مقصد اقداماتی صورت می‌گیرد.**

- **تصحیح خطا**
- **با کنترل جریان**
- **قابلیت اعتماد بالا و با اطمینان**

# TCP
## آدرس سوکت

- **آدرس پورت (Port number)** : هر پروسه برای تقاضای برقراری یک ارتباط با پروسه ای دیگر روی شبکه، یک شماره شناسایی برای خود برمی گزینند. به این شماره شناسایی «آدرس پورت» گفته می شود.
- **آدرس IP** : یک ماشین یکتا را در کل شبکه مشخص می کند.
- **آدرس سوکت : زوج آدرس IP و آدرس پورت** را گویند.

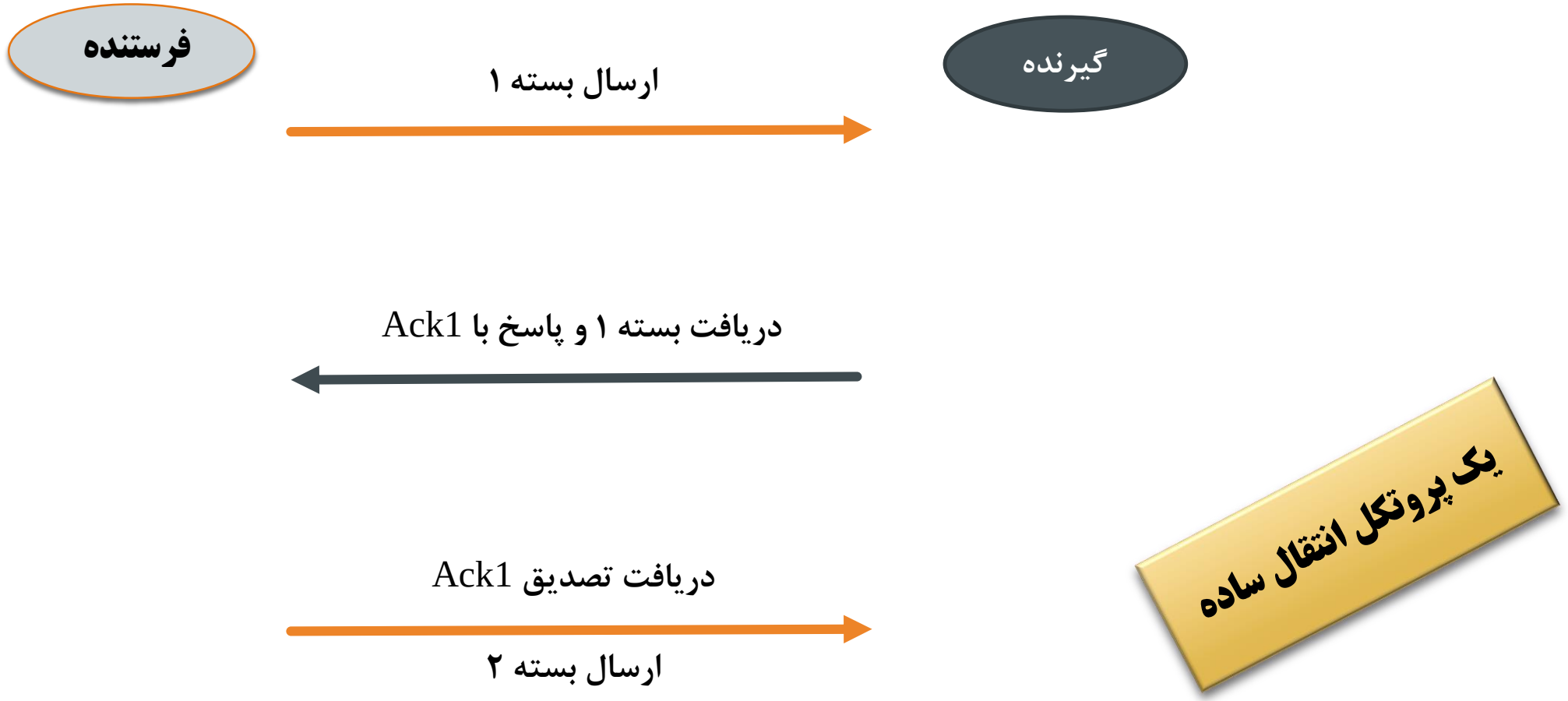
**IP Address: Port Number = Socket Address**

**مثال: 193.142.22.121 : 80**

# TCP
## طرز کار

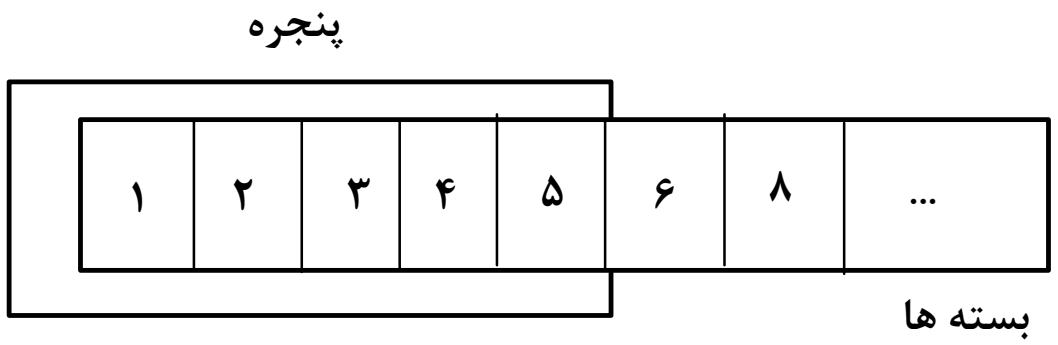
- با استفاده از شماره های ترتیب وتصدیق ها با دیگر ایستگاهها در شبکه ارتباط مطمئن برقرار می کند.
- TCPمبتنی بربایت است. (به ازای هر بایت یک شماره ترتیب در نظر گرفته می شود.)
  - ✓ چند بایت را در قالب یک سگمنت و با یک شماره ترتیب به لایه IP می دهد تا به مقصد ارسال شود.
  - ✓ تصمیم گیری در مورد سگمنت بندی برعهده ی TCP می باشد.

# TCP
## مدیریت پنجره لغزان – طرز کار پنجره




یک پروتکل انتقال ساده

# TCP
## مدیریت پنجره لغزان




- فرستنده پنج بسته را در قالب یک پنجره پیش از دریافت تصدیق ها باهم ارسال نموده و برای هر یک زمان سنجی روشن می کند.
- گیرنده با دریافت هر بسته آن را تصدیق کرده و شماره آخرین بسته ی دریافتی را نگه می دارد.
- فرستنده با دریافت هر تصدیق پنجره ی خود را یک واحد می لغزاند.

# TCP
## مشکل بسته کوچک

- اگر بسته ی فرستنده یک بایتی باشد :
  - ✓ ۲۰ بایت سرآیند IP + ۲۰ بایت سرایندTCP + ۱ بایت بسته ارسالی = ۴۱ بایت می فرستد
- ارسال بسته ی تصدیق توسط گیرنده : ۴۰ بایت
- به روز رسانی پنجره ی فرستنده : ۴۰ بایت
- چاپ کاراکتر روی صفحه کاربر (فرستنده) توسط گیرنده : ارسال بسته ۴۱ بایتی

جمعاً ۱۶۲ بایت هزینه برای چاپ هر کاراکتر

برای خطوط کم سرعت مناسب نیست.

# TCP
## حل مشکل بسته کوچک توسط ناگل

هدف این است که فرستنده بسته های کوچک را نفرستد

- فقط بایت اول ارسال می شود.
- بقیه تا زمان رسیدن تصدیق بایت اول در بافر فرستنده ذخیره می شوند.
- بایت های ذخیره شده در قالب یک بسته TCP ارسال می شود.
- تا زمان رسیدن تصدیق آن دوباره شروع به ذخیره سازی بایت ها می کند.

# TCP
## مشکل سندروم پنجره (Silly Window Syndrome)

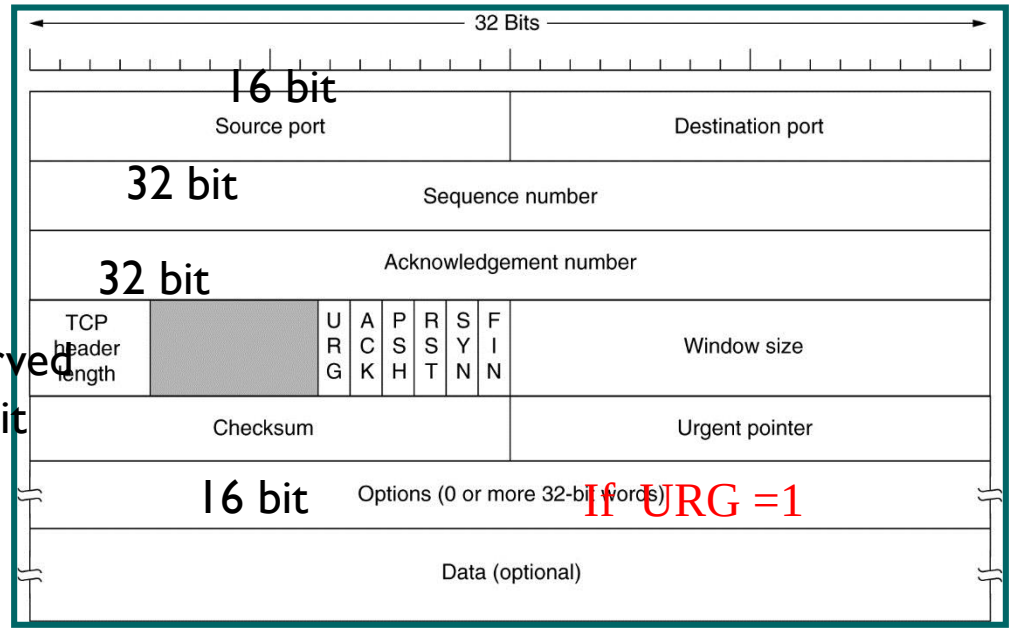
- داده ها در فرستنده بصورت بسته های بزرگ باشند.
- گیرنده تنها یک بایت را در هر بار بخواند.

# TCP
## حل مشکل سندروم پنجره توسط کلارک

هدف این است که گیرنده بسته های کوچک را دریافت نکند

- گیرنده نمی تواند مقدار پنجره ی بروز شده ی خود را برای یک بایت ارسال کند.
- منتظر می ماند تا فضای قابل ملاحظه ای ایجاد شود و بعد اینکار را انجام می دهد.

# TCP
## قالب بسته




16 bit

If SYN =1

If ACK =1

4 bit

Reserved
6 bit

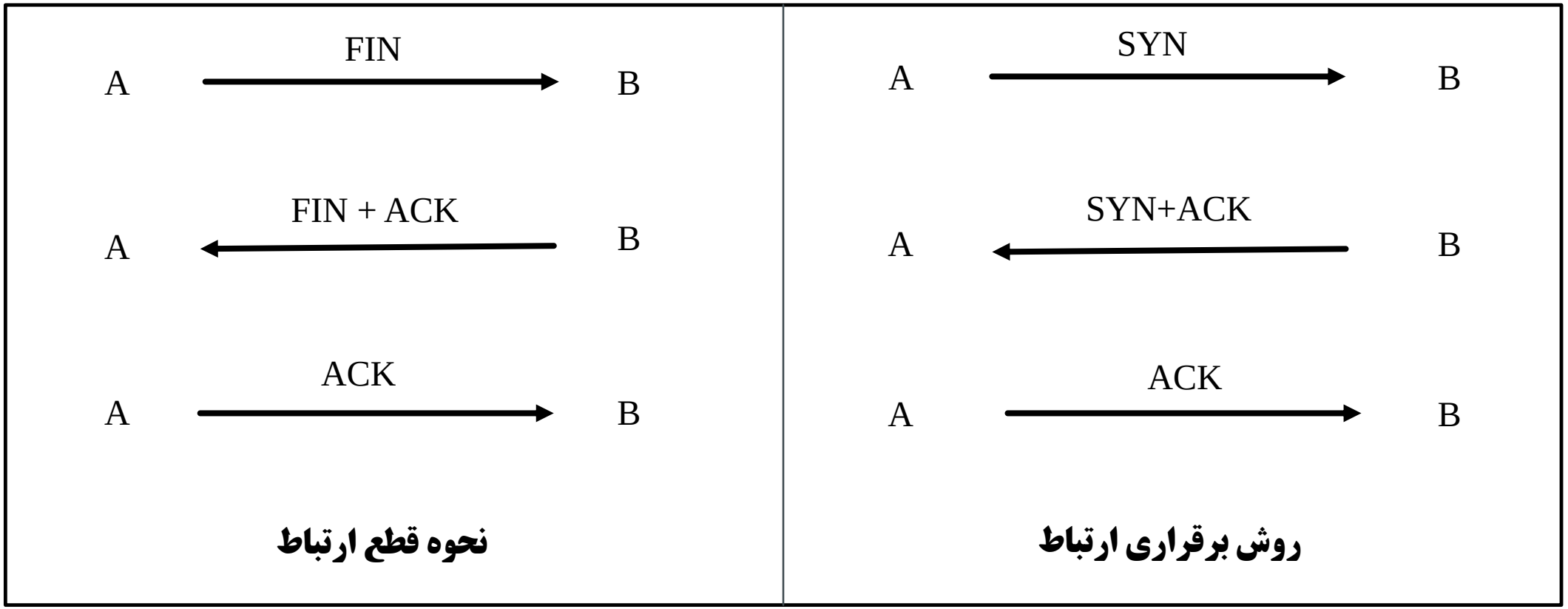
16 bit

16 bit

If URG =1

معرفی فیلدها در داخل کتاب ...

# TCP
## نحوه برقراری و قطع ارتباط (دست دهی سه مرحله ای)



# TCP
## کنترل ازدحام

- وجود مسیریاب ها و لینک هایی با سرعت پایین بین فرستنده و گیرنده
  - ✓ گم شدن تعدادی از بسته ها
  - ✓ کاهش کارایی
- TCP می تواند نرخ فرستنده را با ظرفیت شبکه تنظیم کند و از شرایط ازدحام جلوگیری کند.

# TCP
## کنترل ازدحام – پنجره ازدحام

- پنجره ازدحام : جهت جلوگیری از ازدحام توسط شبکه تعیین می شود.
- اندازه واقعی پنجره فرستنده (نرخ تولید بسته) :
  - ✓ حداقل اندازه بین پنجره گیرنده(میزان بافر خالی گیرنده) و پنجره ازدحام (Congestion Window)

Actual window size = minimum ( rwnd , cwnd)

میزان بافر گیرنده : receiver window

# TCP
## کنترل ازدحام – مدیریت ازدحام

- سیاست کلی TCP برای مدیریت ازدحام در **سه فاز** انجام می شود:
  - ✓ شروع آرام
  - ✓ جلوگیری از ازدحام
  - ✓ تشخیص ازدحام

در **فاز شروع آرام** : فرستنده با یک **نرخ خیلی کم** شروع به ارسال می کند و تا رسیدن به یک آستانه (Threshold) نرخ را زیاد می کند. وقتی سرعت به مقدار آستانه رسید برای

**جلوگیری از ازدحام**: نرخ داده را کاهش می دهد. سرانجام اگر

**ازدحام تشخیص داده شد** : فرستنده به فاز شروع آرام یا جلوگیری از ازدحام بر می گردد.

# TCP
## کنترل ازدحام – مدیریت ازدحام – الگوریتم شروع آرام

در الگوریتم شروع آرام:

- نرخ ارسال بسته در شبکه به نرخ دریافت تصدیق ها از طرف گیرنده بستگی دارد.
- نحوه ی افزایش پنجره ازدحام در قسمت فرستنده تعیین می شود.
  - ✓ پنجره ازدحام توسط فرستنده، براساس ازدحام شبکه، جریان را کنترل می کند.
  - ✓ پنجره اعلام شده توسط گیرنده، کنترل جریان انجام می دهد.(براساس مقدار بافر خالی گیرنده برای فرستنده اعلام میشود)

# TCP
## کنترل ازدحام – مدیریت ازدحام – الگوریتم شروع آرام

- در این فاز TCP از عمل افزایش نمایی استفاده می کند.
- ابتدا مقدار اولیه پنجره ازدحام توسط فرستنده به اندازه یک سگمنت تنظیم می شود. $cwnd = 2^0=1$
- ارسال یک بسته توسط فرستنده
- انتظار دریافت تصدیق توسط فرستنده
- دریافت تصدیق
- افزایش یک واحدی پنجره ازدحام در فرستنده
- ارسال دو بسته $cwnd = 2^1=2$
- پس از انتظار و دریافت تصدیق دو بسته
- افزایش چهار واحدی پنجره ازدحام $cwnd = 2^2=4$
- مقدار آستانه الگوریتم شروع آرام ۶۵۵۳۵ بایت است.
  - ✓ بعد از رسیدن به مقدار آستانه مدیریت ازدحام وارد فاز جلوگیری از ازدحام می شود.

**شکل صفحه ۲۰۷ الگوریتم شروع آرام**

# TCP
## کنترل ازدحام – مدیریت ازدحام – الگوریتم جلوگیری از ازدحام

- در این فاز TCP از عمل افزایش جمعی (Additive Increase) استفاده می کند.
- از زمانیکه اندازه پنجره ازدحام به مقدار آستانه رسید(در الگوریتم شروع آرام ) به ازای هربار دریافت تصدیق، اندازه پنجره یکی اضافه می شود. این عمل تا وقوع ازدحام ادامه می یابد.

- نرخ گم شدن بسته ها بدلیل خرابی بسیار کم است : کمتر از یک درصد
- علت اصلی گم شدن بسته ها : ازدحام
- دو عامل نشان دهنده ازدحام (یا گم شدن بسته ها):
  - ✓ انقضای زمانی
  - ✓ دریافت تصدیق های تکراری

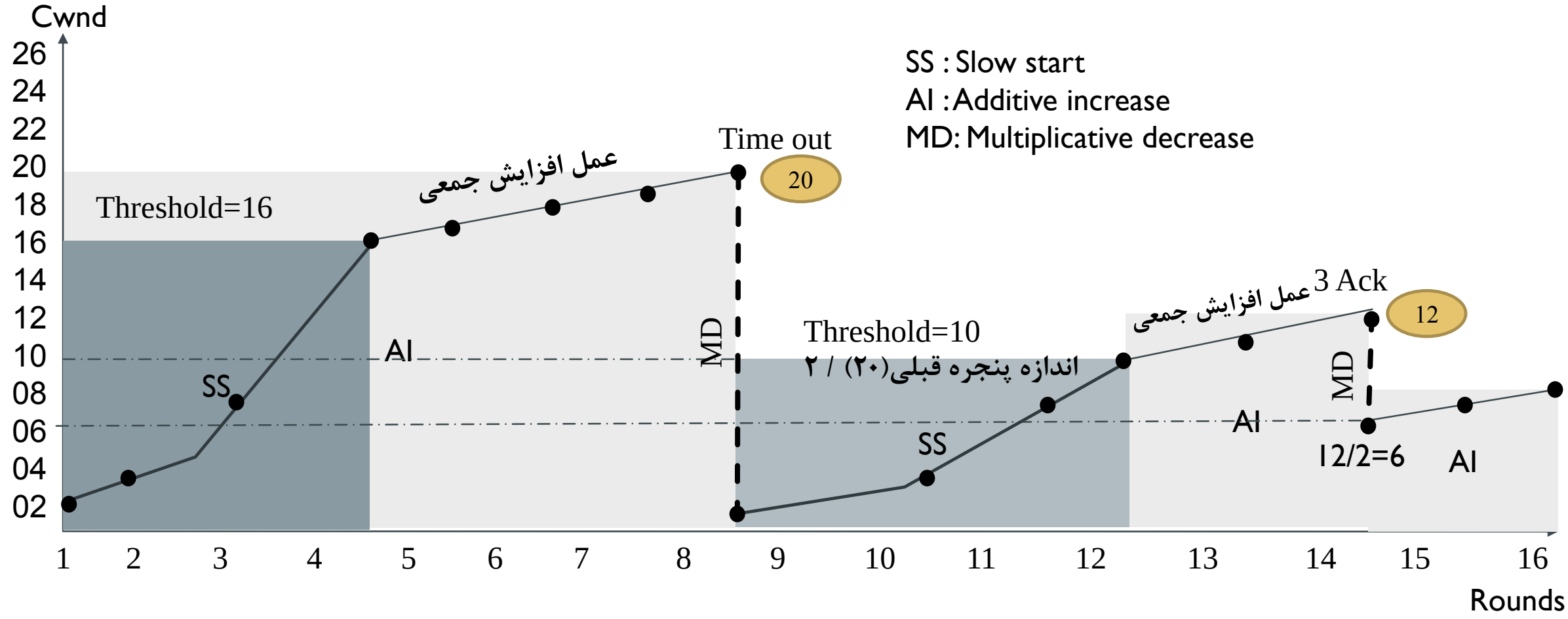
**شکل صفحه ۲۰۸ الگوریتم جلوگیری از ازدحام**

# TCP
## کنترل ازدحام – مدیریت ازدحام – تشخیص ازدحام

- در صورت وقوع ازدحام، اندازه پنجره باید تغییر کند.
  - ✓ تغییر بصورت کاهش ضربی (Multiplicative Decrease) و به اندازه نصف آستانه می باشد.
- اگر در اثر انقضای زمانی : (احتمال ازدحام)
  - ✓ نصف اندازه پنجره قبلی = مقدار آستانه (Threshold)
  - ✓ Cwnd = 1
  - ✓ دوباره وارد فاز شروع آرام (افزایش نمایی) می گردد.
- اگر در اثر تصدیق های تکراری : (احتمال می رود یک سگمنت گم شده باشد)
  - ✓ Threshold = (window size)/2
  - ✓ Cwnd = Threshold
  - ✓ دوباره فاز جلوگیری از ازدحام (عمل افزایش جمعی) را اجرا می کند.
- ➢ اگر cwnd <= Threshold : در حالت شروع آرام است
- ➢ اگر cwnd > Threshold : در حال اجرای الگوریتم جلوگیری از ازدحام

# TCP

## کنترل ازدحام – مدیریت ازدحام – تشخیص ازدحام



# فصل ششم

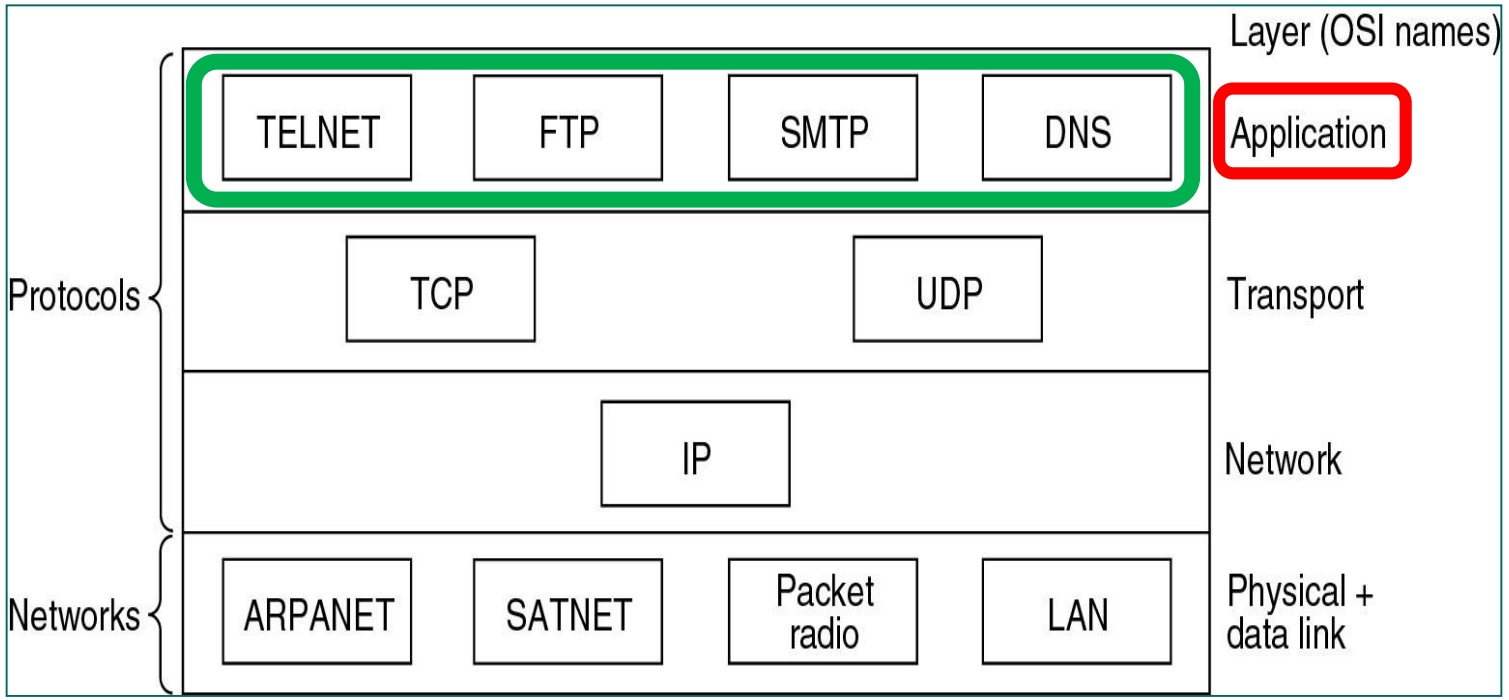
## لایه کاربرد در اینترنت

# فهرست موضوعات

# لایه کاربرد

## مقدمه – پروتکل های مهم لایه کاربرد

- برنامه های کاربردی در لایه کاربرد قرار دارند.
- از دید کاربر درنهایت این برنامه های لایه کاربرد هستند که در شبکه با یکدیگر کار می کنند.




HTTP SNMP

پشته قرارداد TCP/IP

# قرارداد HTTP

از فصل ششم – لایه کاربرد در اینترنت

# لایه کاربرد
## قرارداد HTTP (Hyper Text Transfer Protocol)

- زبان یا قراردادی برای صحبت کردن برنامه مشتری با سرویس دهنده وب است.
- اجازه می دهد مستندات HTML انتقال یابند.
  - ✓ مستندات حاوی : عکس، صدا، ویدئو و غیره
- این قرارداد مبتنی بر درخواست و پاسخ است.
  - ✓ درخواست مشتری از سرور
  - ✓ پاسخ سرور به مشتری
- یک قرارداد بدون حالت است.
  - ✓ ردپایی از اتصالات ذخیره نمی کند.
  - ✓ اگر در خواستی به اطلاعات مبادله شده در ارتباط قبلی وابسته باشد، این قرارداد کاری انجام نمی دهد.
    - برای رفع این مشکل از کوکی ها استفاده می شود.

**کوکی** : یکسری اطلاعات است که بین مرورگر مشتری و سرویس دهنده، در طول تراکنش HTTP مبادله می شود . حداکثر اندازه کوکی ۴ کیلوبایت است.(این اطلاعات در یک فایل در مسیر مرورگر ذخیره میشود- توسط سرویس دهنده چک می شود.)

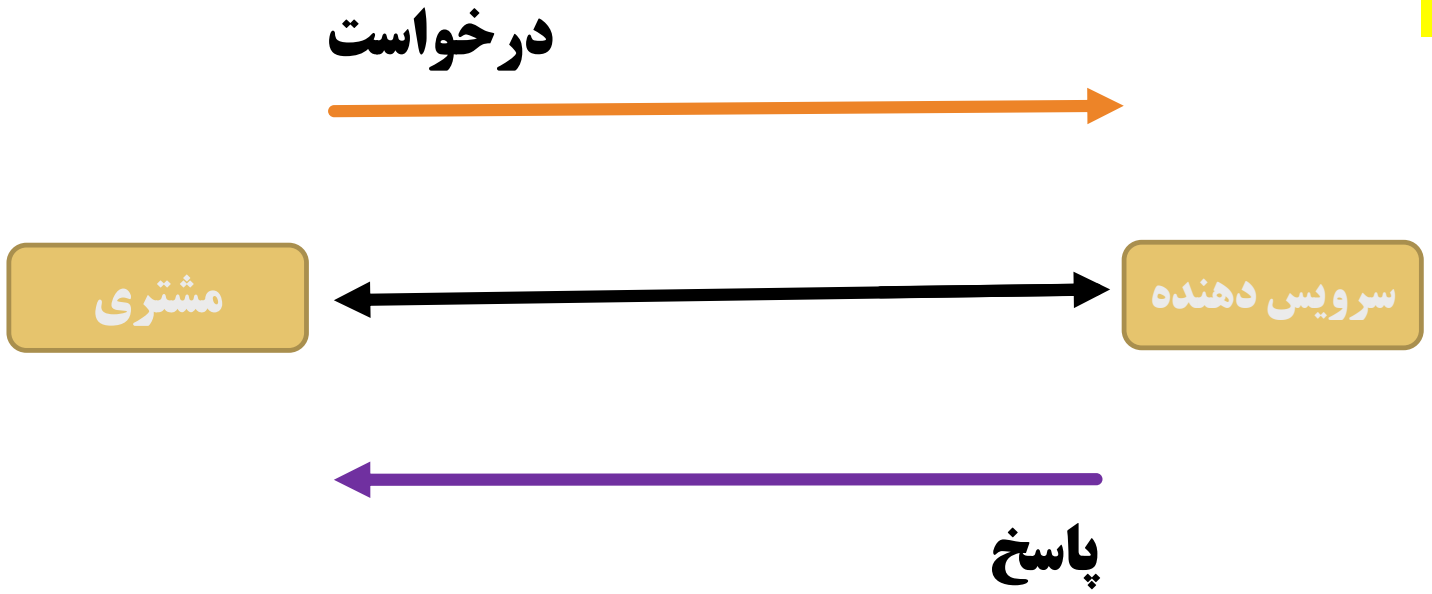
# لایه کاربرد
## قرارداد HTTP (Hyper Text Transfer Protocol)

- **بطور کلی یک تراکنش HTTP به چهار مرحله تقسیم می شود:**

1. مشتری توسط مرورگر یک اتصال TCP روی پورت ۸۰ با سرویس دهنده (سرور) باز می کند.
2. مرورگر یک درخواست HTTP شامل آدرس URL به سرویس دهنده می فرستد.
3. سرویس دهنده با دریافت درخواست و خواندن اطلاعات از حافظه خود یک پیام پاسخ به مرورگر مشتری می فرستد.
4. بعد از دریافت پیام توسط مشتری اتصال TCP بسته می شود.

# لایه کاربرد
## قرارداد HTTP – طرز کار

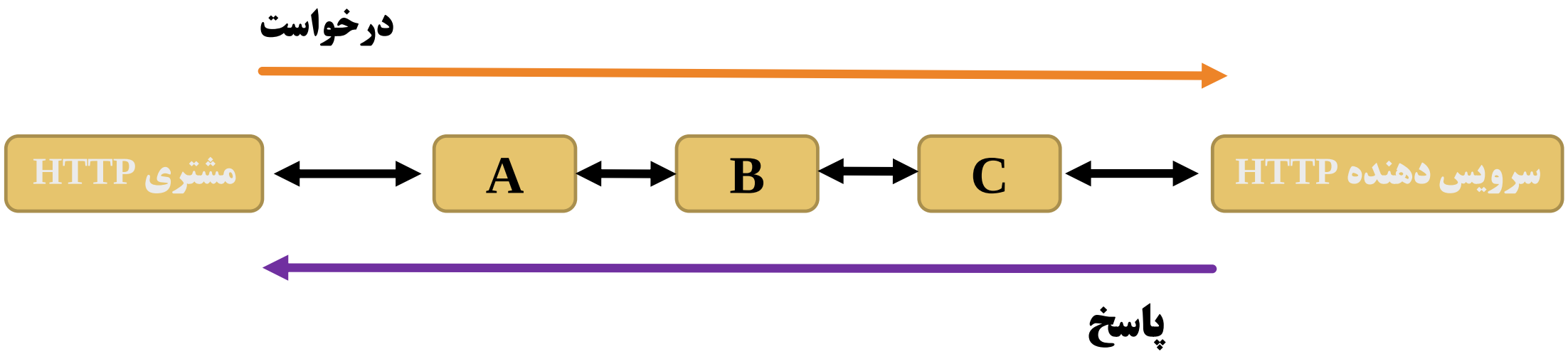
**ارتباط مستقیم**




**ساده ترین حالت اتصال HTTP بین مشتری و سرویس دهنده**

# لایه کاربرد
## قرارداد HTTP – طرز کار

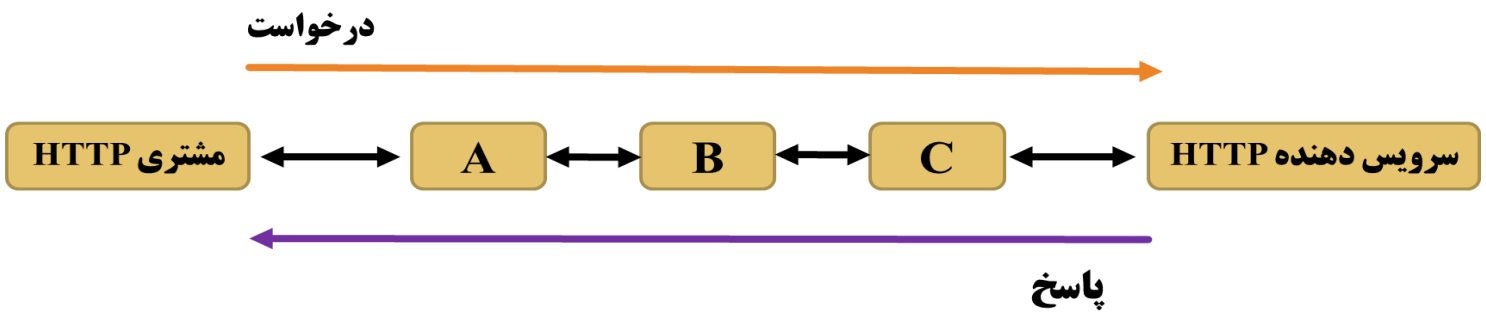
**ارتباط به همراه چندین واسط**



# لایه کاربرد
## قرارداد HTTP – طرز کار

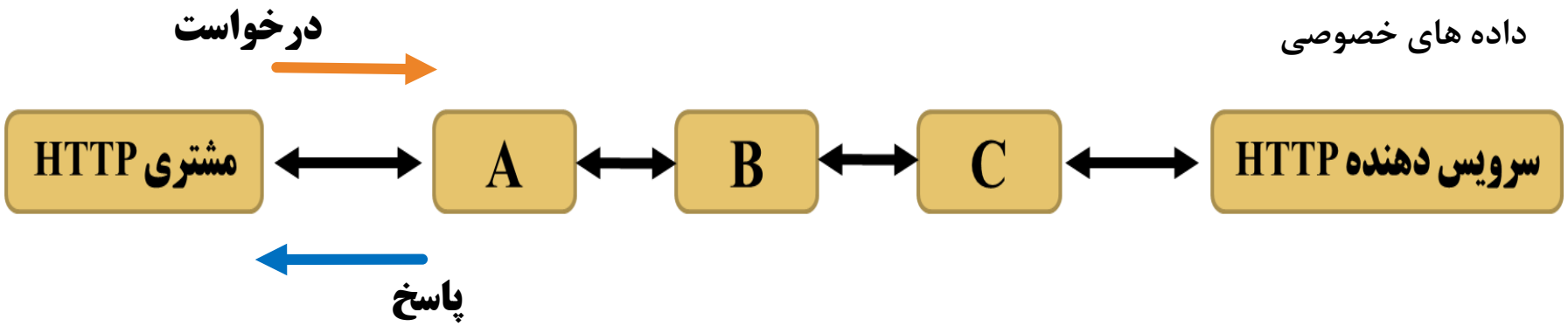
ارتباط به همراه چندین واسط

- **هیچ ارتباط مستقیمی بین سرویس دهنده و کاربر وجود ندارد.**
  - ✓ واسط هایی مانند پروکسی، دروازه، تونل ....
  - ✓ **پروکسی**: به محتویات داده دسترسی داشته و می تواند آنها را تغییر دهد (بازنویسی درخواست).
  - ✓ **دروازه** : ارسال پیام به قراردادهای زیرین با قالب مناسب
  - ✓ **تونل** : با محتویات پیام کاری ندارد، آنها را بدون تغییر ارسال می کند.
- **درخواست ها و پاسخ ها توسط واسط، ارزیابی و بعد به سمت مقصد یا دیگر واسط های احتمالی هدایت می شوند.**



# لایه کاربرد
## قرارداد HTTP – طرز کار

- **پروکسی ها و دروازه ها می توانند داده ها (صفحات عبوری ) را cache کنند.**
  - ✓ این کار زمان پاسخ و ترافیک شبکه را کاهش می دهد.
  - ✓ درخواست بعدی مشتری وقتی به جایی برسد که داده ی مورد درخواست **cache** شده است از همانجا پاسخ خود را دریافت می کند. (لازم به طی کردن مسیر تا سرویس دهنده نیست)
  - ✓ البته همه صفحات قابل cache شدن نیستند.
    1. داده ها ی غیرقابل cache
    2. داده های عمومی
    3. داده های خصوصی




**A یک کپی از داده موردنظر را در پاسخ ها قبلی ذخیره داشته است**

# لایه کاربرد
## قرارداد HTTP (Hyper Text Transfer Protocol)

شکل صفحه ۲۱۵ – قالب کلی پیام HTTP

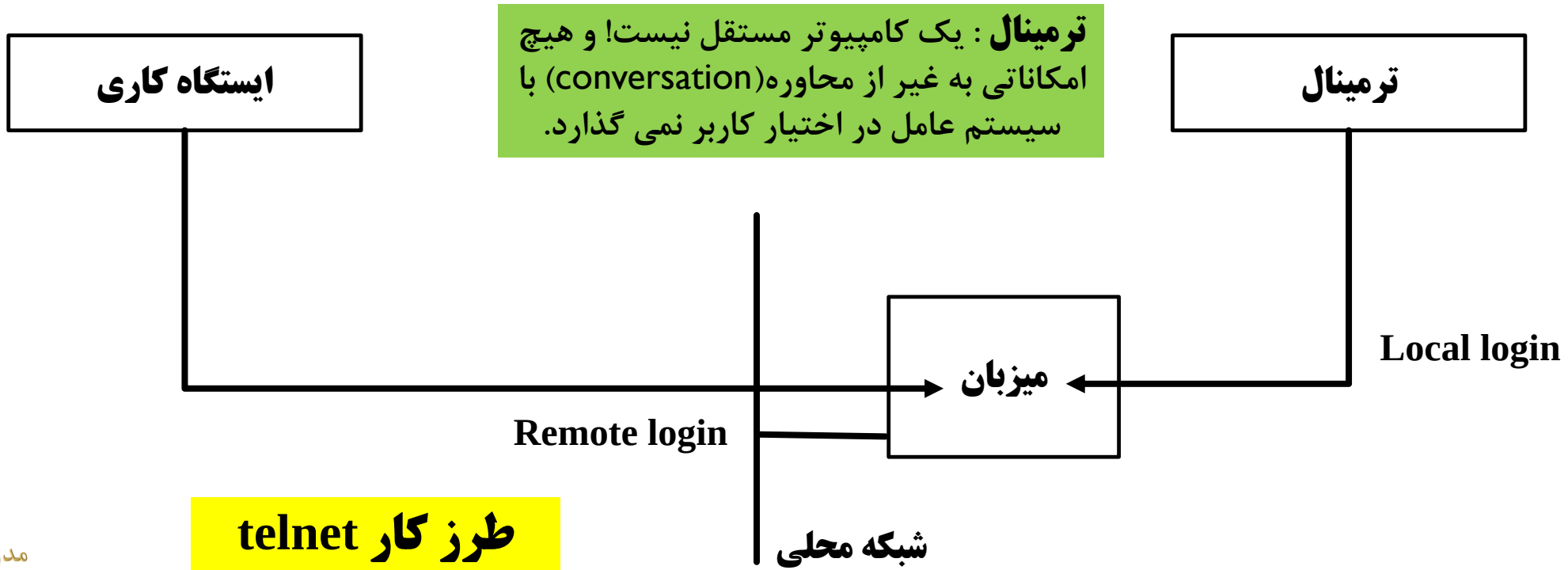
جداول صفحه ۲۱۶ ، ۲۱۷ ، ۲۱۸ و ۲۱۹

# قرارداد Telnet

از فصل ششم – لایه کاربرد در اینترنت

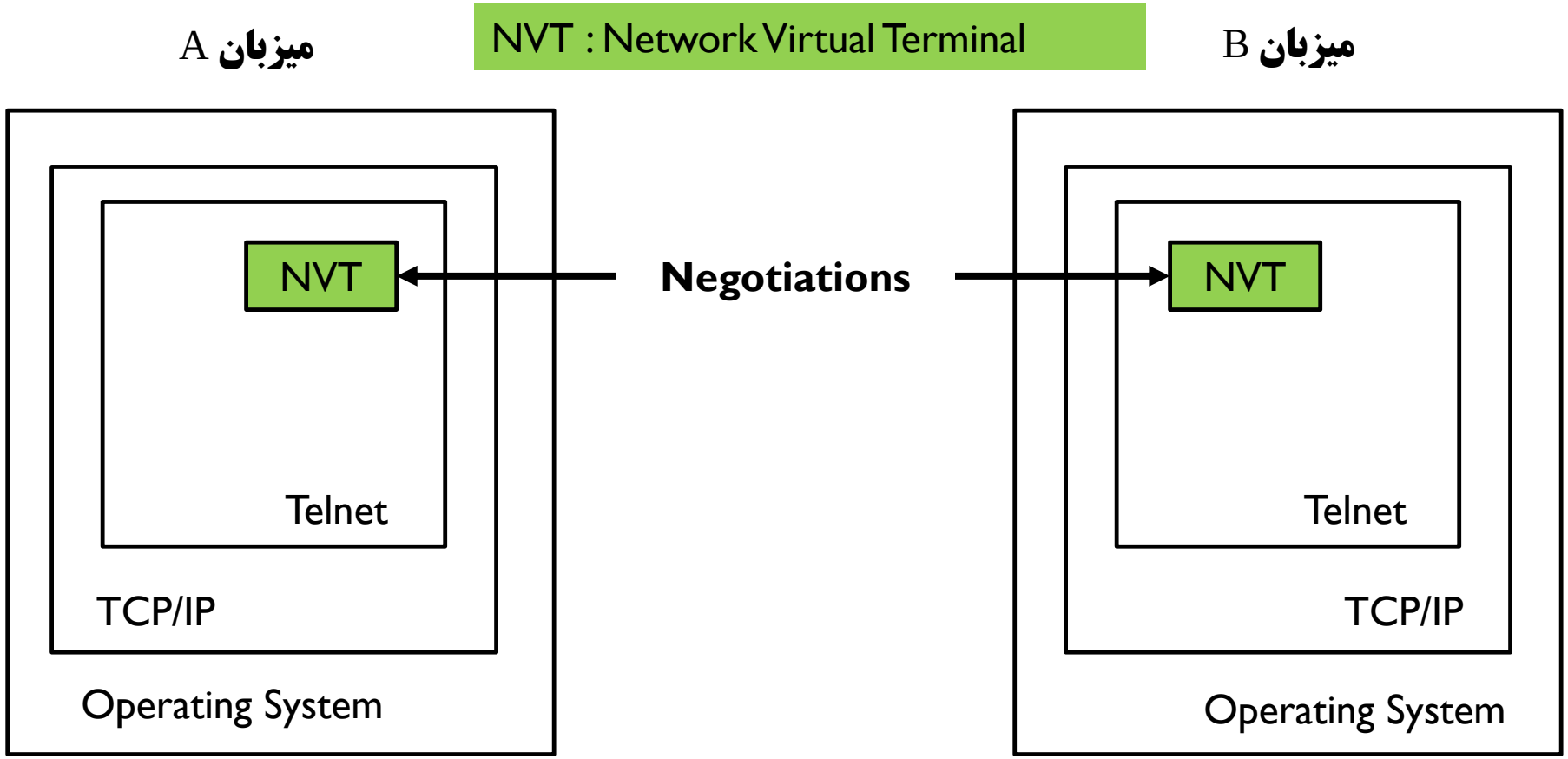
# لایه کاربرد
## قرارداد Telnet

- قرارداد Telnet یک رابط استاندارد فراهم می کند که از طریق آن یک برنامه روی یک میزبان (مشتری telnet) می تواند به منابع میزبان دیگر (سرویس دهنده telnet) روی پورت ۲۳ دسترسی پیدا کند.



# لایه کاربرد
## قرارداد Telnet

NVT : Network Virtual Terminal




**مذاکرات telnet**

# لایه کاربرد
## قرارداد Telnet

- ترمینال مجازی (NVT) دارای یک صفحه نمایش و یک صفحه کلید است برای دریافت و نمایش داده.
- مشخصات پایه ای یک NVT :
  - ✓ نمایش داده بصورت کد اسکی ۷ بیتی است که در بایت های ۸ بیتی ارسال می شود.
  - ✓ یک دستگاه نیمه دو طرفه است (Half duplex)
  - ✓ با یک تابع echo محلی می تواند داده ها را روی ترمینال محلی نیز نمایش دهد.

# لایه کاربرد
## قرارداد Telnet – ساختار دستور

- ارتباط بین مشتری و سرویس دهنده با دستورهای داخلی اداره می شود که کاربر به آنها دسترسی ندارد.
- دستورهای داخلی telnet براساس نوع دستور از ۲ یا ۳ بایت تشکیل می شود.

| Interpret As Command(IAC) | کد دستور | گزینه های مذاکره شده |
|---|---|---|
| بایت اول | بایت دوم | بایت سوم |

WILL : اعلام شروع بـه کـار یـا تایید اینکه در حال کار هستید.

| 255 | 253 | 24 |
|---|---|---|
| IAC | WILL | نوع ترمینال |

# لایه کاربرد
## قرارداد Telnet – دستورهای آن

- telnet مشتری دو مد عملیاتی دارد.
  - ✓ مد دستور
  - ✓ مد نشست
- **مد دستور** به ترمینال اجازه می دهد تا
  - ✓ یک اتصال با میزبان راه دور برقرار نموده
  - ✓ پارامترهای عملیاتی را نشان دهد،
  - ✓ تنظیمات را انجام دهد،
  - ✓ وضعیت را چاپ کند
  - ✓ و از برنامه خارج شود.

- پس از اتصال، telnetمشتری وارد **مد نشست** شده

از طریق نشست خط فرمان، کاربر می تواند:

- ✓ برنامه های روی کامپیوتر راه دور را اجرا کند.
- ✓ با [+Ctrl] از مد نشست وارد مد دستور شود:
  - ✓ تنظیمات ترمینال را تغییر دهد.
  - ✓ با Enter دوباره وارد مد نشست شود.
- دستورهای موردنظر خود را بصورت زیر به سرویس دهنده بفرستد :

Send [\\RemoteServer][دستور]

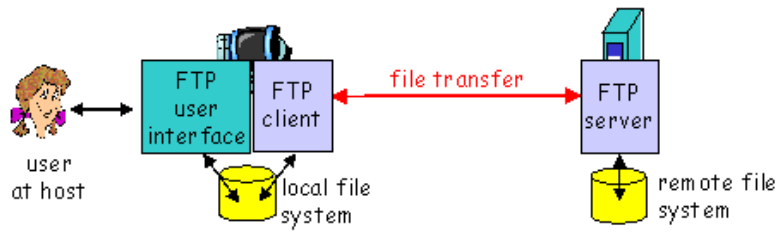
جدول صفحه ۲۲۴

# قرارداد FTP

از فصل ششم – لایه کاربرد در اینترنت

# لایه کاربرد
## قرارداد FTP

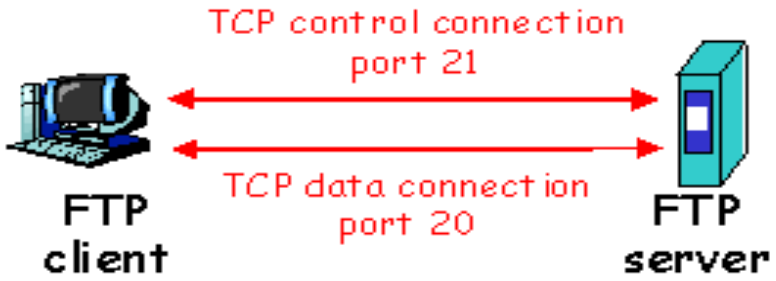
## ftp: the file transfer protocol




- transfer file to/from remote host
- client/server model
  - *client:* side that initiates transfer (either to/from remote)
  - *server:* remote host
- ftp: RFC 959
- ftp server: port 21

# ftp: separate control, data connections

- ftp client contacts ftp server at port 21, specifying TCP as transport protocol
- two parallel TCP connections opened:
  - control: exchange commands, responses between client, server.
    "out of band control"
  - data: file data to/from server
- ftp server maintains "state": current directory, earlier authentication



# ftp commands, responses

## Sample commands:

- sent as ASCII text over control channel
- `USER` *username*
- `PASS` *password*
- `LIST` return list of file in current directory
- `RETR` filename retrieves (gets) file
- `STOR` filename stores (puts) file onto remote host

## Sample return codes

- status code and phrase (as in http)
- `331 Username OK, password required`
- `125 data connection already open; transfer starting`
- `425 Can't open data connection`
- `452 Error writing file`

# قرارداد TFTP

از فصل ششم – لایه کاربرد در اینترنت

# لایه کاربرد

## قرارداد TFTP ( Trivial File Transfer Protocol)

- یک قرارداد(پروتکل) انتقال فایل ساده است.
- دلیل سادگی معماری آن پیاده سازی راحت است.
- TFTP در لایه انتقال از قرارداد UDP و پورت ۶۹ استفاده می کند.
  - ✓ ابتدا مشتری TFTF درخواست نوشتن یا خواند را روی این پورت می فرستد سپس
  - ✓ سرویس دهنده و مشتری پورتی که برای ادامه کار بکار می برند را تعیین می کنند.
- TFTP احراز هویت نداشته و لذا یک قرارداد ناامن است.
- مزایای زیادی نسبت به FTP سنتی دارد :
  - ✓ استفاده توسط دستگاههای بدون دیسک جهت دریافت اطلاعات لازم در زمان بوت
  - ✓ استفاده توسط هر فرآیندی که در آن نمی توان شناسه کاربری و رمز عبور تعریف نمود
  - ✓ در برنامه های کوچک، که باعث پیاده سازی ارزان آنها شده و همچنین در محیط هایی که مشکل منابع وجود دارد.

# لایه کاربرد
## قرارداد TFTP – طرز استفاده

- برای این قرارداد دستورهای استانداردی تعریف نشده و تنها تعامل مستقیم مشتری و سرویس دهنده استاندارد شده است.
- بطور کلی میتوان دستورهای زیر را مطرح نمود:
  - **Connect <host>** : شناسه میزبان مقصد را مشخص می کند.
  - **Mode <ASCII | Binary>** : نوع مد انتقال را مشخص می کند.
  - **Get < remote filename> | [<local filename]** : فایل را دریافت می کند.
  - **Put <remote filename> | [<local filename]** : فایل را ذخیره می کند.
  - **Verbose**: اطلاعات اضافی در حین انتقال را نمایش می دهد.
  - **Quit** : از TFTP خارج می شود.

# لایه کاربرد
## قرارداد TFTP – طرز استفاده

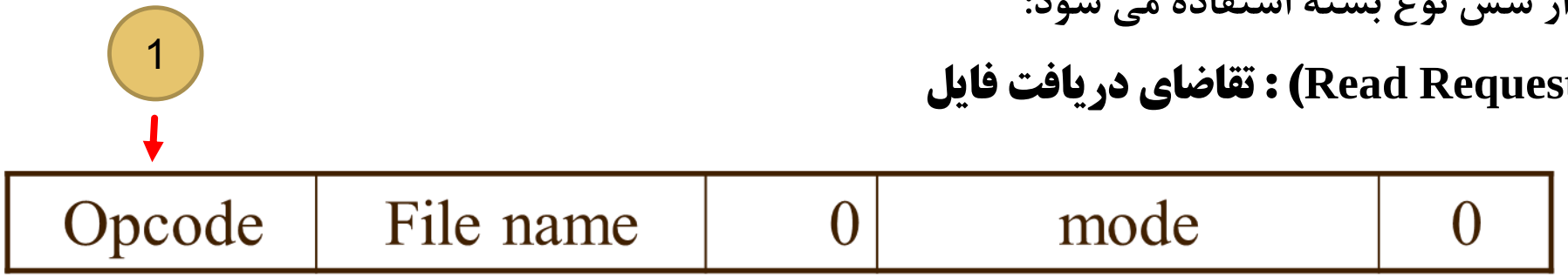
- هر انتقال TFTP با یک درخواست خواندن و نوشتن شروع می شود.
- اگر سرویس دهنده آن را قبول نمود، اتصال باز شده و فایل در بلوکهای ۵۱۲بایتی منتقل می شود.
  - ✓ این بلوکها از شماره ۱ شروع شده و هر بسته تنها یک بلوک را حمل می کند.
- پیش از ارسال بسته جدید، باید تصدیق قبلی بیاید.
- اگر بسته ای کوچکتر از ۵۱۲ بایت ارسال شود ارتباط قطع می شود.
  - ✓ بدلیل ارتباط نامطمئن، اکثر خرابیها منجر به قطع ارتباط می شوند.
- TFTP می تواند بسته های گم شده را با روشن نمودن زمان سنج و ارسال مجدد بسته بازیابی کند.
  - ✓ ارسال مجدد هم بخاطر گم شدن بسته و هم گم شدن تصدیق ها انجام می شود.

سندورم SAS (Apprentice Syndrome Sorcerer's): شبکه هایی که تاخیر زیاد دارند (توقف و انتظار زیاد)

- ✓ رفع مشکل با بسته های OACK (Option Acknowledgment)

# لایه کاربرد
## قرارداد TFTP – بسته ها

- در این قرارداد از شش نوع بسته استفاده می شود:
- **بسته RRQ (Read Request) : تقاضای دریافت فایل**

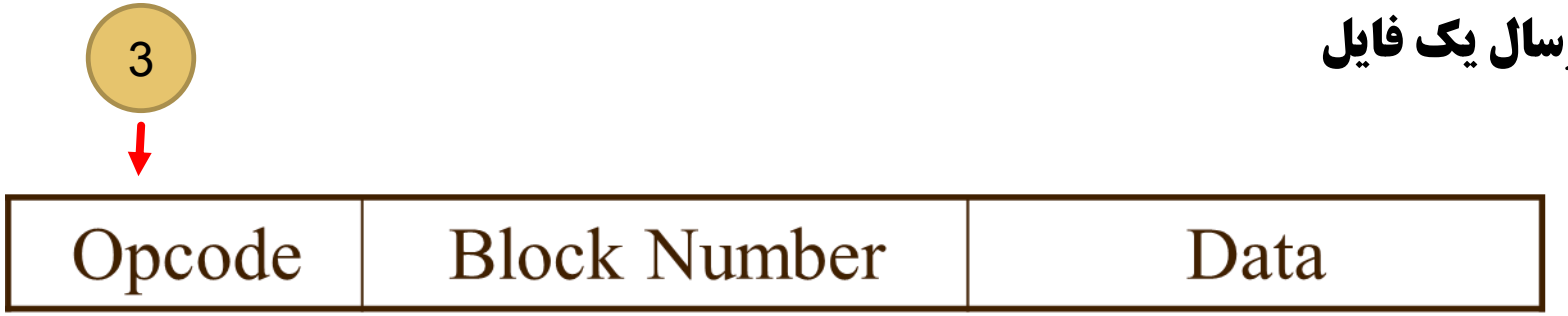


# لایه کاربرد
## قرارداد TFTP – بسته ها

- بسته WRQ (Write Request): تقاضای ارسال یک فایل

# لایه کاربرد
## قرارداد TFTP – بسته ها

- بسته DATA : تقاضای ارسال یک فایل

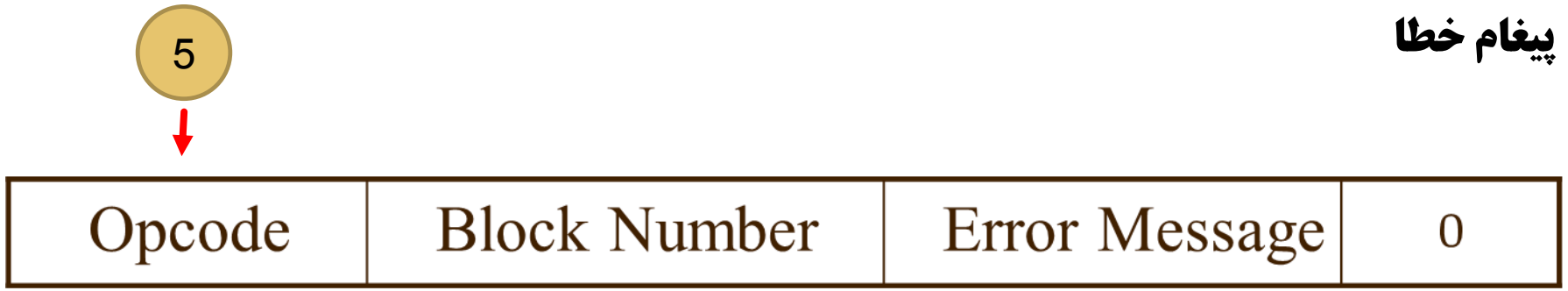


# لایه کاربرد
## قرارداد TFTP – بسته ها

- بسته ACK : پیغام تصدیق و پذیرش

# لایه کاربرد
## قرارداد TFTP – بسته ها

- بسته ERROR: پیغام خطا

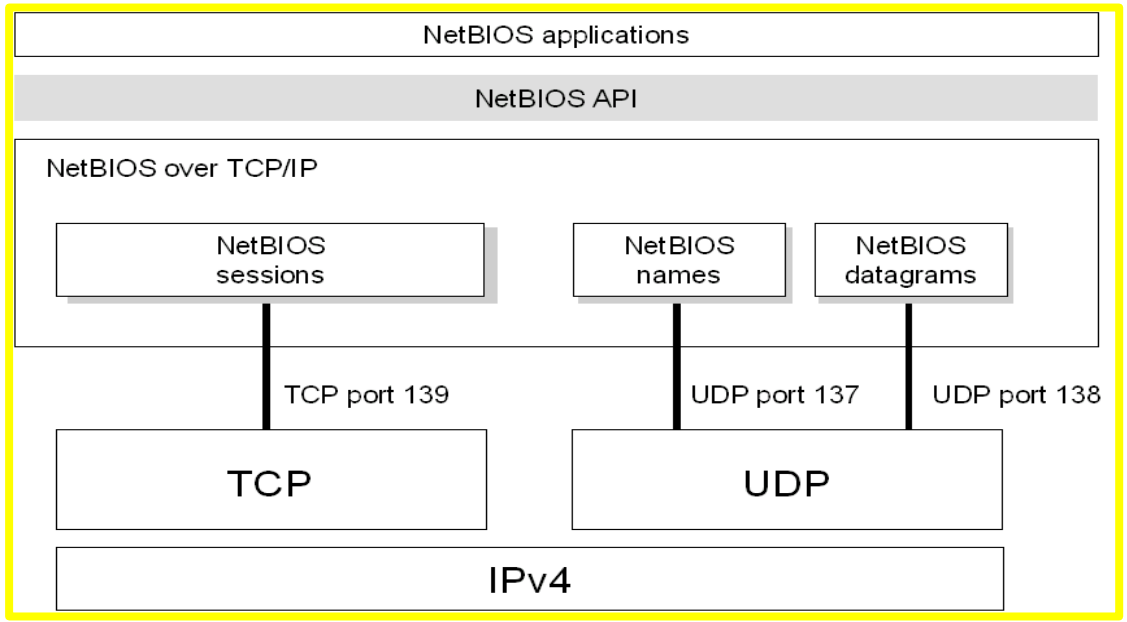


# سرویسهای تحلیل نام

از فصل ششم – لایه کاربرد در اینترنت

# لایه کاربرد
## سرویس های تحلیل نام (Name Resolution) – NetBIOS

- **تحلیل نام** : به فرآیند تبدیل نام به آدرس و برعکس تحلیل نام گفته می شود.
- **NetBIOS** : قرارداد(پروتکل) اصلی شبکه در DOS و ویندوز محسوب می شود.
  - ✓ در ابتدا برای سادگی مثلاً بجای آدرس ۱۰.۱۲.۷.۱۴ اسم computer1 را قرار می دادند. سپس به روشی دوباره به آدرس ۱۰.۱۲.۷.۱۴ تبدیل می شود.
  - ✓ نام ها ۱۶ بایتی است و بایستی در کل شبکه یکتا باشند.
  - ✓ بسته های NetBIOS دارای آدرس شبکه(NetID) نیست
  - ✓ عمل مسیریابی روی این بسته ها انجام نمی گیرد.
  - ✓ بسته ها به صورت پخشی روی شبکه ارسال می شوند.




**ساختار منطقی قرارداد NetBIOS**

# سرویس WINS

از فصل ششم – لایه کاربرد در اینترنت

# لایه کاربرد
## WINS (Windows Internet Name Service)

- اسامی NetBIOS در فایلی به نام Hosts.txt نگهداری می شد که به ایـن سیسـتم Local Manager Host میگفتند (LMHOST) .
- با گسترده شدن اینترنت این فایل دیگر جوابگوی نیازها نبود.
- مایکروسافت سرویسی را به نام WINS معرفی کرد که یک Database توزیع شده بود و این اسامی را شامل مـی شد.

✓ روش کار به این صورت بود که کامپیوترها درخواستی را برای تبدیل نام می دادند کامپیوتر ابتـدا بـه Cache خود رجوع میکرد اگر این آدرس در Cache کامپیوتر وجود نداشت یک request را به صورت Unicast بـه نرم افزار Wins ارسال می کرد و از آنجا پاسخ مناسب را دریافت می کرد.

# لایه کاربرد
## WINS – مزایا

- مدیریت بهتر اسامی
- عدم وجود نام های تکراری
- ارسال پیام مبنی بر حضور به نرم افزار Wins

# لایه کاربرد
## WINS – معایب

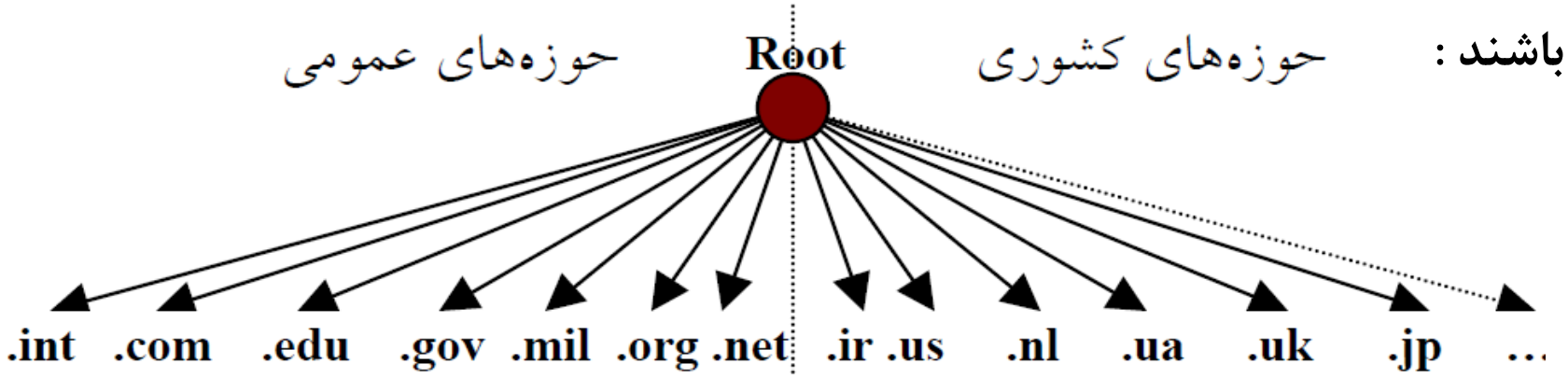
- بالا رفتن ترافیک شبکه (پیام های پخشی)
  - ✓ کند شدن سرعت شبکه
- نبود یک ساختار اسمی مناسب

# سرویس DNS

از فصل ششم – لایه کاربرد در اینترنت

# لایه کاربرد
## سرویس DNS (Domain Name Service)


- شرکت مایکروسافت سرویس DNS را در سال ۱۹۸۴ معرفی نمود.
- DNS یک Database توزیع شده است.
- دارای یک ساختار درختی است.
- دامنه های سطح بالا دو نوع می باشند :
  - ✓ عمومی
  - ✓ کشوری



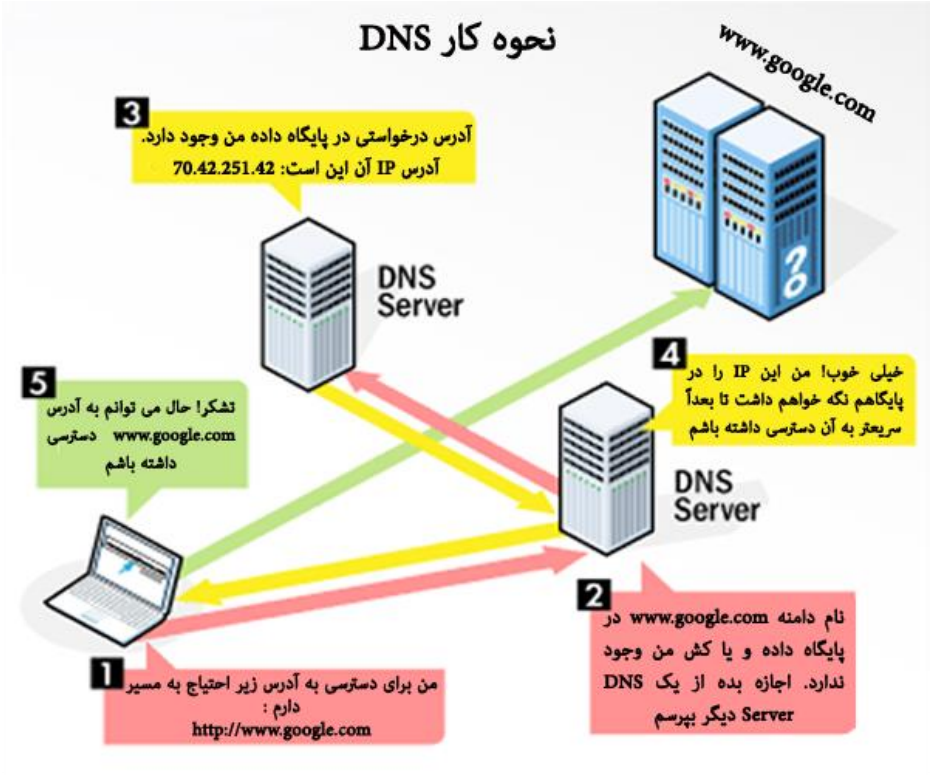
# لایه کاربرد
## سرویس DNS (Domain Name Service) – فرایند تحلیل نام

برای تبدیل نام سایت به IP :

- ابتدا به تحلیلگر حافظه پنهان خود (Resolver Cache) که همان فایل Hosts است مراجعه می کند.
  - ✓ مسیر در ویندوز : C:\ WINDOWS\System32\drivers\etc\
- اگر کامپیوتر نتوانست در مراجعه به این فایل آدرس موردنظر را بیابد:
  - ✓ یک درخواست آماده
  - ✓ درخواست را به سرویس دهنده DNS شبکه خود ارسال می کند.
  - ✓ این سرویس دهنده بسته را دریافت و ابتدا به قسمت نواحی (zones) تعریف شده در قرارداد مراجعه می کند.
  - ✓ اگر در Zones جوابی پیدا نشد، شروع به جستجو در DNS Server Cache می کند.
  - ✓ بعد از اتمام کار در صورتیکه همچنان جوابی پیدا نشده باشد درخواست به ریشه ها (Root Hints) ارسال می شود که در حقیقت همان سرویس دهنده های DNS اینترنتی است.

# لایه کاربرد

## سرویس DNS (Domain Name Service) – فرایند تحلیل نام

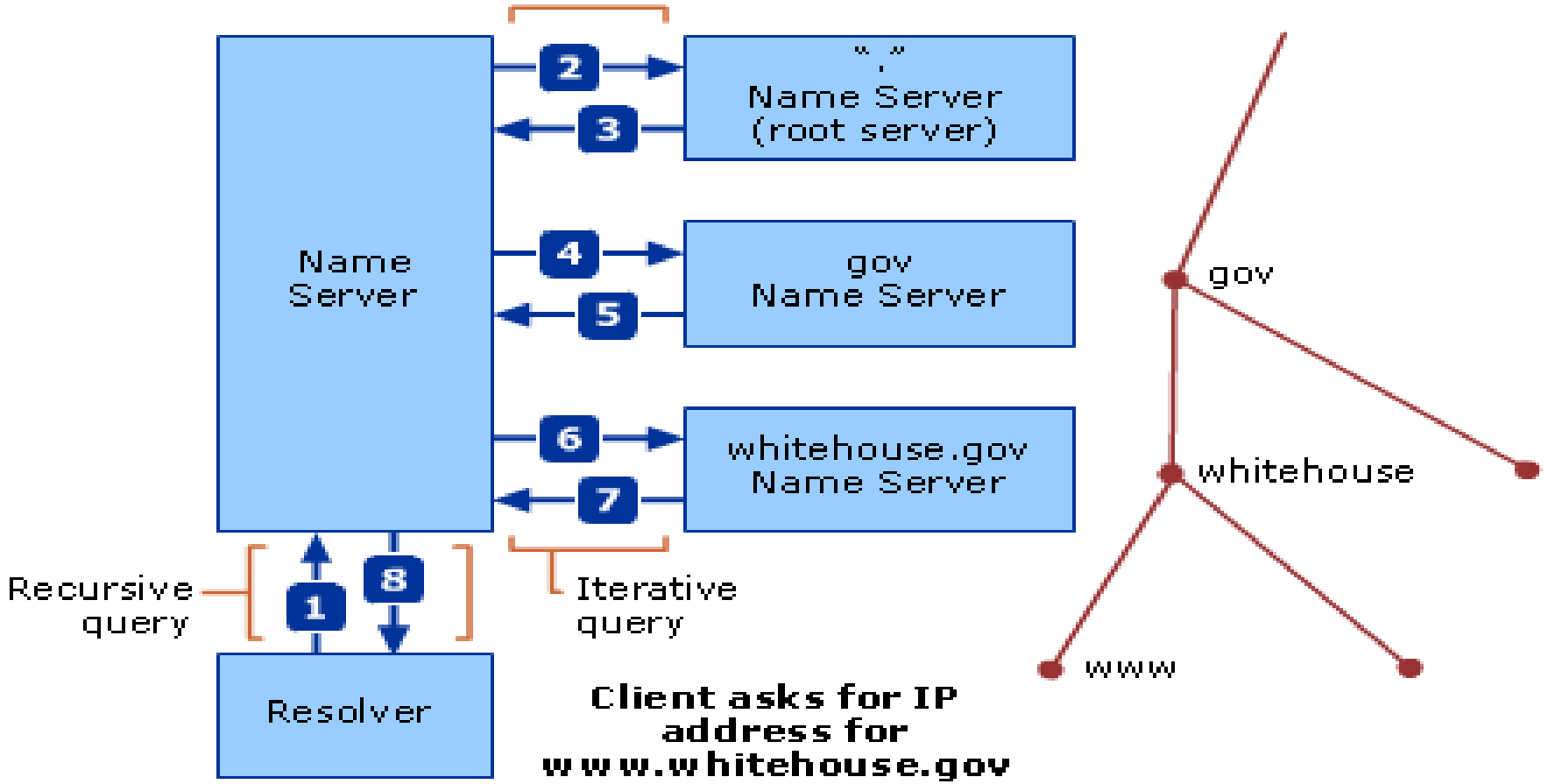


# لایه کاربرد
## سرویس DNS (Domain Name Service) – روش جستجوی تکراری

- روند ترجمه آدرس Microsoft.com را در نظر بگیرید:
- ۱-برنامه کاربردی با فراخوانی تابع تحلیل گر نام تقاضای ترجمه آدرس را برای DNS محلی ارسال می کند.
- ۲-سرویس دهنده محلی از Root، آدرس ماشینی را که دربرگیرنده دامنه Com. است را سوال میکند.
- ۳-آدرس ماشین هایی که در برگیرنده دامنه Com. هستند ارسال میشود.
- ۴- Local DNS از ماشین معرفی شده در مرحله قبل آدرس سرور مربوط به دامنه Microsoft.com را سوال میکند.
- ۵-فهرستی از سرور های مربوط به Microsoft.com را برمیگرداند.
- ۶- Local DNS این تقاضا را به Microsoft.com ارسال میکند.
- ۷-معادل Ip سایت Microsoft.com برمیگردد.
- و در مرحله آخر آدرس Ip خواسته شده در اختیار مرورگر قرار میگیرد.

# لایه کاربرد

## سرویس DNS (Domain Name Service) – روش جستجوی تکراری

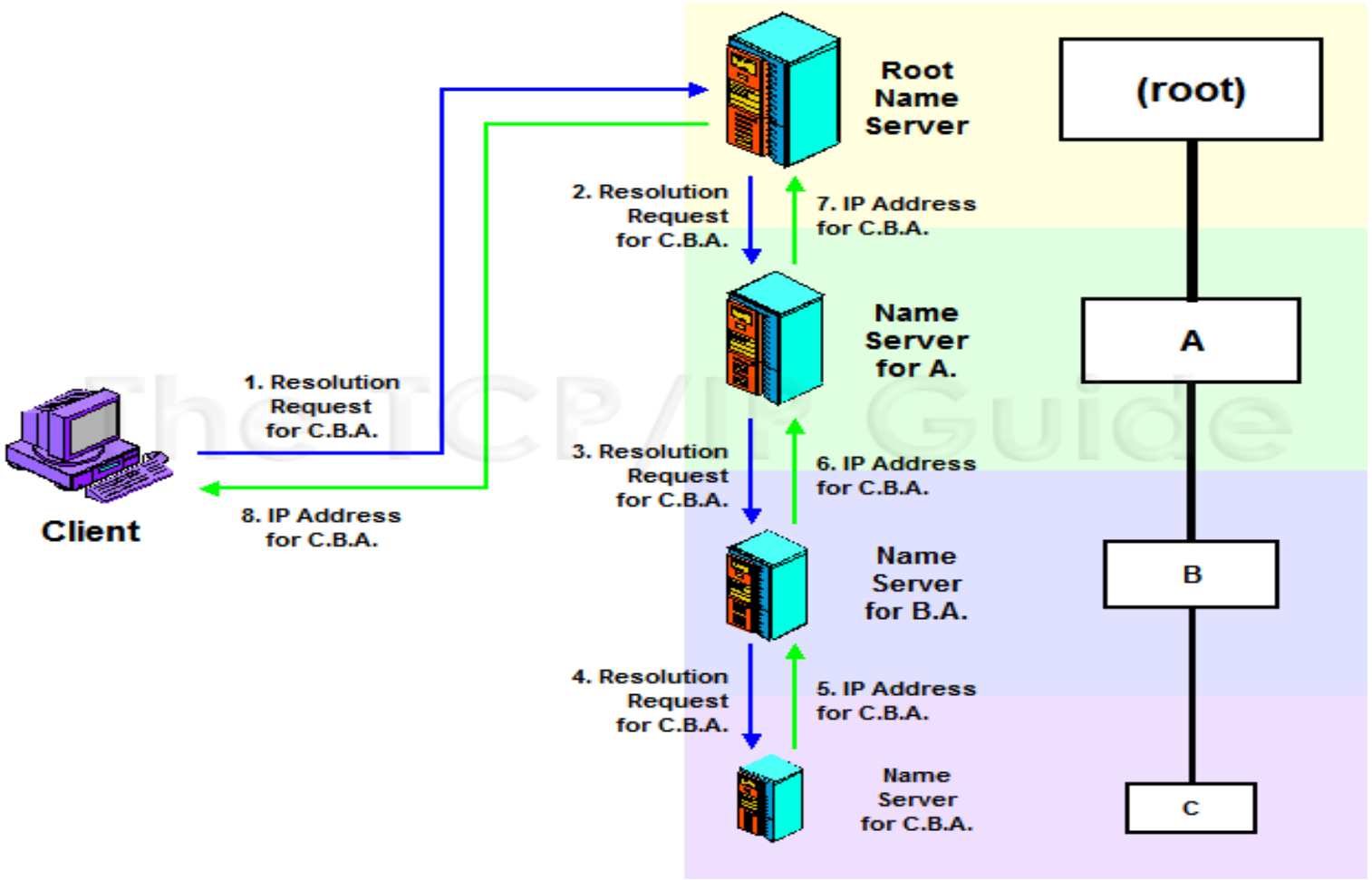


# لایه کاربرد
## سرویس DNS (Domain Name Service) – روش جستجوی بازگشتی

- برنامه کاربردی با فراخوانی تابع تحلیل گر نام تقاضای ترجمه آدرس را برای DNS محلی ارسال میکند.
- Local DNS این تقاضا را به یک DNS ریشه میفرستد.در این حالت Local DNS موظف است بدون آن که به تقاضا دهنده خبر دهد خودش این تقاضا را به یک سرور ریشه ارسال کند.
- Root DNS به همین ترتیب درخواست را پیگیری میکند اگر این ادرس را داشته باشد به Local DNS ارسال میکند در غیر اینصورت از سرویس دهنده های سطح پایین تقاضای ترجمه آدرس میکند پس در این پرس و جو دیگر عمده کار برعهده Local DNS نمی باشد. و در این جا Local DNS عملی جز ارسال request را انجام نمیدهد.

# لایه کاربرد

## سرویس DNS (Domain Name Service) – روش جستجوی بازگشتی



# لایه کاربرد
## سرویس DNS (Domain Name Service) – روش جستجوی معکوس

- این پرس و جو در حالتی مطرح میشود که IP در دسترس باشد ولی نام نمادین آن را نداشته باشیم.
- با توجه به NetID موجود در IP Address این جستجو انجام شده که یک جستجوی کاملا وقت گیر مـی باشـد و صفحه مناسب توسط مرورگر در اختیار کاربر قرار می گیرد.

# موتورهای جستجو

از فصل ششم – لایه کاربرد در اینترنت

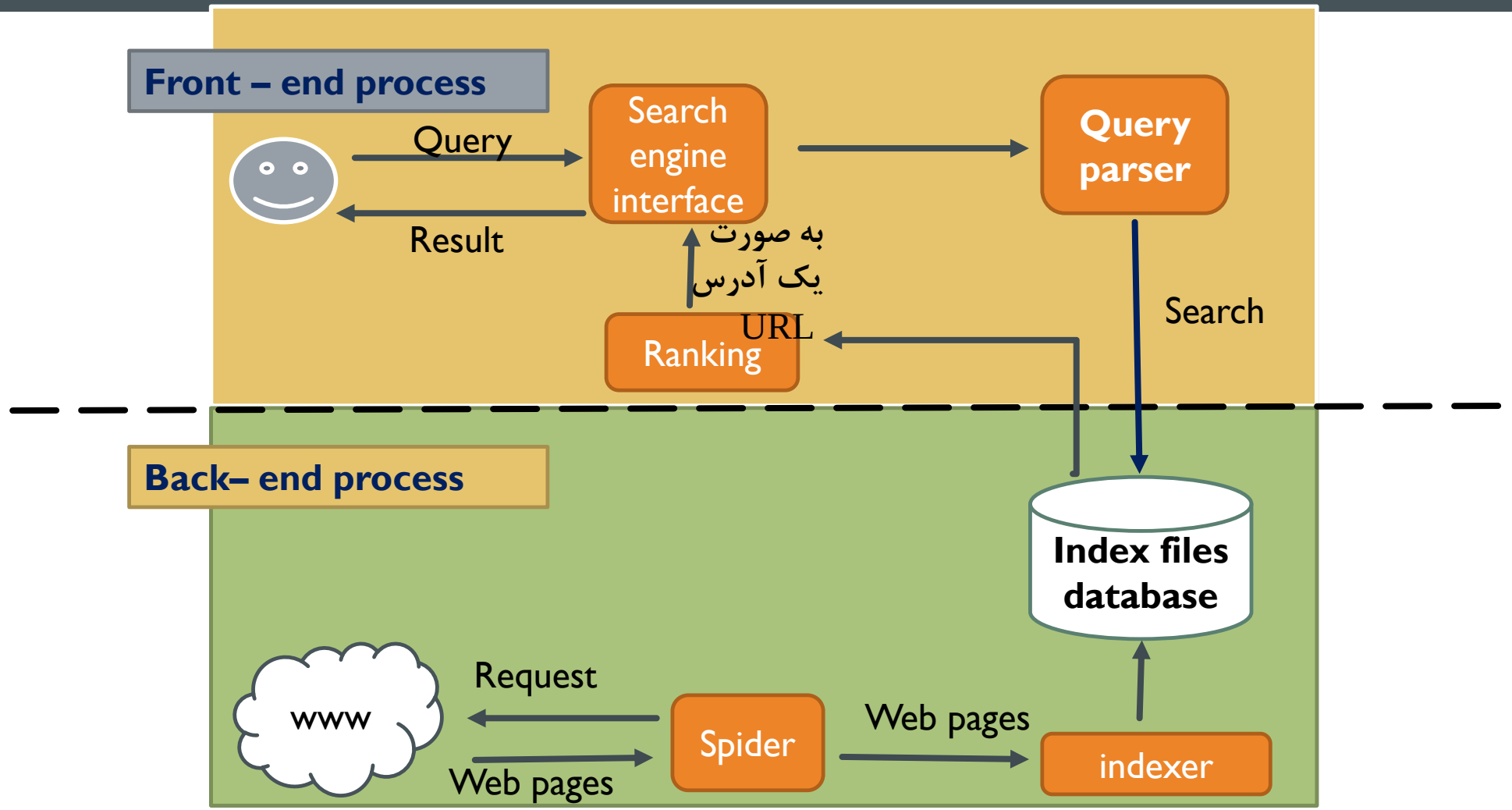
# لایه کاربرد
## موتورهای جستجو

- یافتن اطلاعات در سریعترین زمان از موتورهای جستجوی رایج : گوگل، یاهو
- **موتور جستجو :** به برنامه ای گفته می شود که موضوعات مورد نظر کاربران در قالب **کلمات کلیدی** در یک سند یا بانک اطلاعاتی جستجو می کند و نتایج بدست آمده را بصورت **آدرس محل ذخیره** نشان می دهد.
  - ✓ کلمات کلیدی در میان انبوه اطلاعات وجود در فایلها، سندهای وب جهانی، گروههای خبری، آرشیوهای FTP
  - ✓ امروزه بیشتر موتورهای جستجو اطلاعات طبقه بندی شده ای مانند دایرکتوری ها دارند.

# لایه کاربرد
## موتورهای جستجو

- **انواع موتورهای جستجو:**
- **مبتنی بر پیمایش (Crawler-Based) :**
  - ✓ مانند گوگل ، فهرست خود را بصورت خودکار تشکیل می دهند.
  - ✓ کل وب را پیمایش کرده و مجموعه اسناد و اطلاعات را رده بندی و ذخیره می کنند.
  - ✓ سپس کاربران از میان اطلاعات ذخیره شده آنچه را که می خواهند جستجو می کنند.
  - ✓ در کل صفحه عمل جستجو انجام میشود و هر گونه تغییر اگر در صفحه وبی وجود داشته باشد آن را پیدا میکند و تغییـرات را در فهرست خود اعمال میکند.
- **مبتنی بر فهرست (Directory-Based) :**
- این نوع جستجو ها فقط در قسمت خاصی انجام میشود. مثل عنوان یک سایت یا کلمات کلیدی موجود در آن و اگر تغییراتی در خود صفحه ایجاد شود روی فهرست تغییری به وجود نمی آید.
- **ترکیبی (Hybride) :** بیشتر، نتایج مبتنی بر فهرست را در نظر میگیرد ولی از یافته های مبتنی بر پیمایش نیز استفاده مـی کند. مثل موتورهای جستجوی MSN

# لایه کاربرد
## معماری پایه ای یک موتور جستجو



# لایه کاربرد
## موتورهای جستجو – فایل robots.txt

- استفاده از فایل Robot.txt باعث میشود که صفحات وب سایت توسط موتور جستجو دنبال نشود.
- اگر کاربری بخواهد هیچ قسمتی از وب سایت او دیده نشود از این ۲ خط استفاده میکند:
- User-agent:*
- Disallow:/

# لایه کاربرد
## بهینه سازی موتورهای جستجو

- صفحات ۲۵۱ تا ۲۵۵ مطالعه شود.

# فصل هفتم

# قراردادهای پست الکترونیکی

# فهرست موضوعات

# مقدمه

- امروزه اینترنت و نامه الکترونیکی به عنوان عمده ترین وسیله ارتباطی در بیشتر تجارت ها بکار گرفته می شوند.
- در این فصل قراردادهای TCP/IP راجع به نامه های الکترونیکی بررسی می شوند، که شامل :
  - ✓ قرارداد انتقال نامه الکترونیکی ساده (SMTP)
  - ✓ برنامه SendMail
  - ✓ سیستم نامه الکترونیکی چندمنظوره (MIME)
  - ✓ قرارداد اداره پست (POP)
  - ✓ قرارداد دسترسی به پیام های اینترنتی (IMAP)

# قرارداد SMTP

از فصل هفتم – قراردادهای پست الکترونیکی

# (Simple Mail Transfer Protocol) SMTP

- این قرارداد جهت ارسال پیام، بین دو سیستم، بصورت انتها به انتها بکار می رود.
  - ✓ میزبان به دروازه
  - ✓ دروازه به دروازه
  - ✓ دروازه به میزبان مقصد
- از کدهای اسکی ۷بیتی برای داده های متنی استفاده می کند.
- روی پورت ۲۵ تماس برقرار می کند.
- سیستم SMTP تضمین نمی کند که نامه حتما به مقصد برسد بلکه به یک دروازه می رساند.

# SMTP – پیام ها

- در قرارداد SMTP هر پیام حاوی دو قسمت سرآیند و محتویات می باشد:
  - ✓ سرآیند : سرآیند نامه با یک خط null خاتمه می یابد.
  - ✓ محتویات : هرچیزی بعد از خط null بدنه برنامه می باشد که متشکل از کارکترهای اسکی است. (کارکترهایی با کداسکی زیر ۱۲۸)

# SMTP – قالب سر آیند نامه

- سرآیند پیام توسط SMTP مدیریت می شود.
- سرآیند بصورت لیستی از مشخصات به شکل Keyword : Value می باشد.

لیستی از فیلدهای رایج سر آیند SMTP

| کلمه کلیدی | مقدار |
|---|---|
| TO | آدرس گیرنده های اولیه پیام |
| CC | آدرس گیرنده های ثانویه پیام (کاربن کپی) |
| BCC | آدرس گیرنده های ثانویه بدون اطلاع از آدرس یکدیگر |
| From | آدرس فرستنده پیام |
| Reply-to | جهت پاسخ به فرستنده پیام |
| Return-path | آدرس ها و مسیری که می توان از آن ها به فرستنده رسید. این فیلد توسط آخرین سیستمی که نامه را تحویل می دهد اضافه شود. |
| Subject | خلاصه ای از پیام فرستاده شده که توسط فرستنده مشخص می شود. |

# SMTP – سرآیند نامه– یک نمونه

**From :** myemail@gmail.com

**To :** “your email” <youremail@yahoo.com>

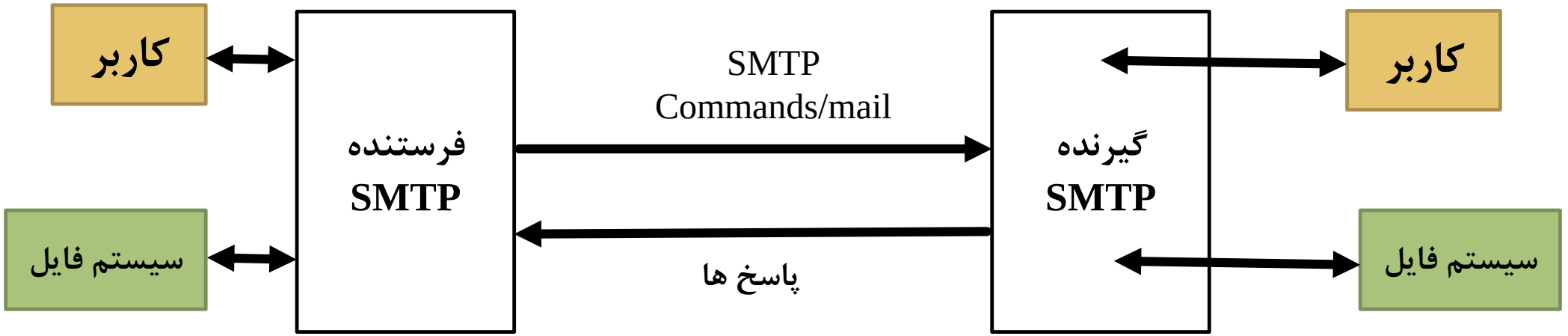
**CC :** “your Boss” <yourBoss@Hotmail.com>

**Reply-to :** myemail@gmail.com

**Subject :** This is a sample SMTP header

# SMTP – تبادل نامه

- بعد از درخواست کاربر، فرستنده SMTP یک ارتباط دو طرفه با گیرنده SMTP برقرار می کند.
- گیرنده می تواند مقصد نهایی یا یک دروازه میانی باشد.
- فرستنده با ارسال دستور و دریافت پاسخ از جانب گیرنده کار را انجام می دهد.

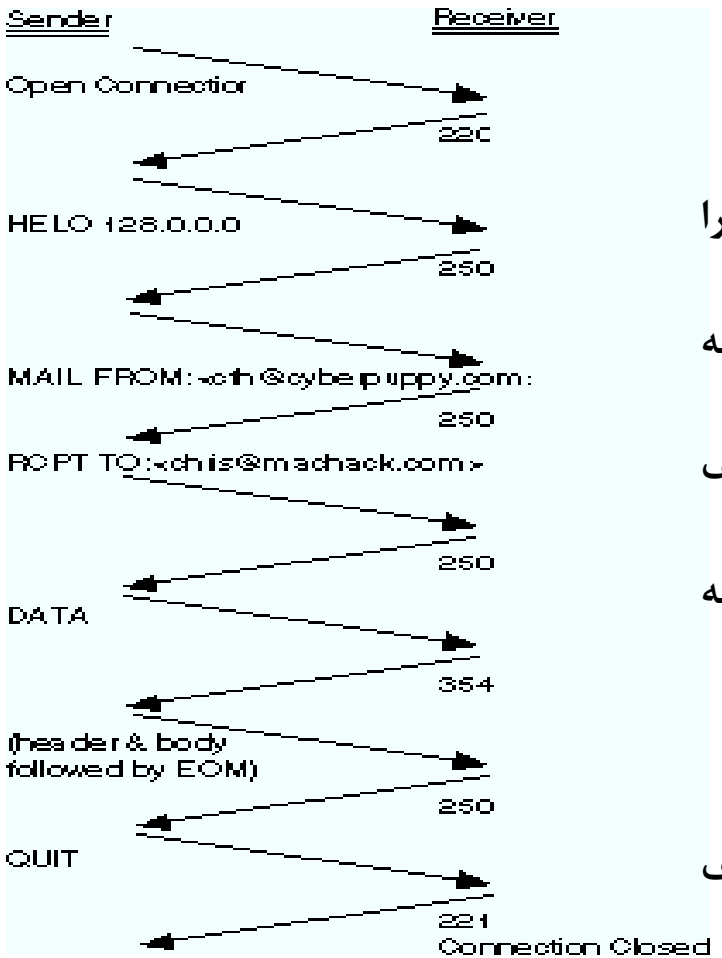



مدل SMTP

# SMTP – روند تراکنش نامه الکترونیکی

```
Server:  220 smtp.example.com ESMTP Postfix      <-- smtpd_client_restrictions
Client:  HELO mail.ora.com                       <-- smptd_helo_restrictions
Server:  250 smtp.example.com
Client:  MAIL FROM: <info@ora.com>               <-- smtpd_sender_restrictions
Server:250 Ok
Client:  RCPT TO: <kdent@example.com>            <-- smtpd_recipient_restrictions
Server:  250 Ok
Client:  DATA                                    <-- smtpd_data_restrictions
Server:  354 End data with <CR><LF>.<CR><LF>
Client:  To: Kyle Dent <kdent@example.com>       <-- header_checks
         From: <info@ora.com>
         Subject: SMTP Example

         This is a message body.                 <-- body_checks
         It continues until a dot
         is typed on a line by itself.
         .
Server:  250 Ok: queued as 6012B20DD6E
Client:  quit
Server:  221 Bye
```




**کد 220** : آماده سرویس

**کد 421** : عدم دسترسی سرویس

**فرمان mail** : فرستنده شروع تراکنش نامه را مقداردهی اولیه میکند.

**فرمان RCPT TO** : دادن آدرس های گیرنده به سرور

**کد 550** : اگر آدرس کاربری برای سرور تعریف نشده باشد.

**فرمان DATA** : فرستنده به گیرنده می گوید که محتویات پیام در ادامه خواهد آمد.

**کد 354** : جهت شروع نامه

**<CRLF>.<CRLF>** : برای پایان نامه

**فرمان QUIT** : درخواست خاتمه از طرف فرستنده(مشتری)

**کد 221** : بستن ارتباط از طرف گیرنده (سرور)



**توجه داشته باشید که سرآیند پیام ها بخشی از داده هایی است که انتقال داده می شود.**

# برنامه Send Mail

از فصل هفتم – قراردادهای پست الکترونیکی

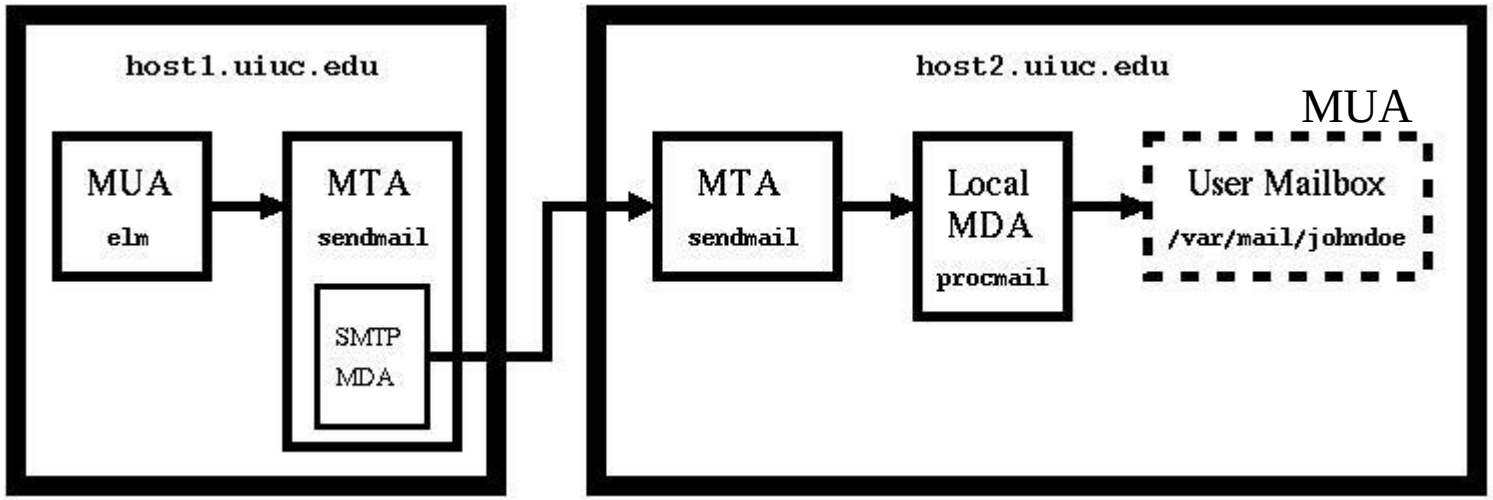
# برنامه Send Mail

- از یکی از قدیمی ترین برنامه های انتقال نامه در اینترنت مربوط به آرپانت نشأت گرفته است.
- یک ابزار مبتنی بر خط فرمان است.
- برای اکثر سیستمهای عامل شبیه Unix طراحی شده است.
- این ابزار، روشی برای ارسال نامه تعریف نمی کند بلکه مانند یک برنامه سرویس دهنده/مشتری از انواع قراردادهای نامه پشتیبانی می کند.
- از پورت ۲۵ برای پیام های ورودی استفاده می نماید.
- از یک سیستم صف برای مدیریت نامه های ورودی و خروجی استفاده می کند.
  - ✓ پیامهایی که مدت زمان زیادی در صف باشند سرانجام دور انداخته می شوند.
- امروزه در بیشتر پیاده سازی های sendmail از قرارداد SMTP استفاده می شود.

# برنامه Send Mail – طرز کار

- MUA (Mail User Agent) **عامل کاربر** : رابطی است که کاربر از طریق آن می تواند نامه ای ارسال یا دریافت کند.
- MTA (Mail Transfer Agent) **عامل انتقال نامه الکترونیکی** : مانند مسیریاب (براساس نوع سرآیند)
- MDA (Mail Delivery Agent) **عامل تحویل نامه الکترونیکی** : این عامل، پیامها را از عامل انتقال گرفته و به مکـان مناسب تحویل می دهد. (مانند لایه انتقال برای مدل نامه الکترونیکی)

**مدل نامه الکترونیکی**




در این مدل Sendmail نقش عامل انتقال را بازی می کند.

# MIME

از فصل هفتم – قراردادهای پست الکترونیکی

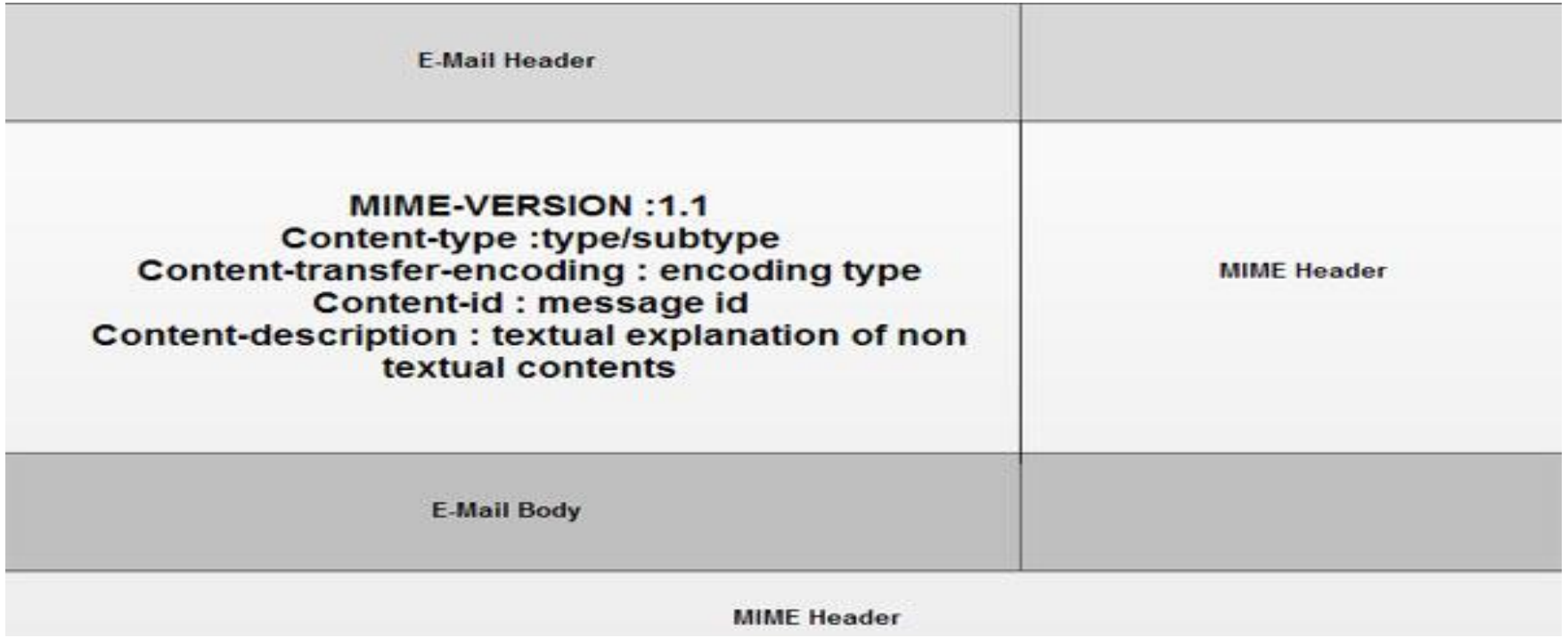
# (Multipurpose Internet Mail Extensions) MIME

- MIME (سیستم نامه الکترونیکی چندمنظوره)
  - ✓ استانداردی است شامل مکانیسم هایی برای حل مشکلات و محدودیت های SMTP :

- 1. SMTP cannot transfer executable files and binary objects.
- 2. SMTP cannot transmit text data of other language, *e.g.* French, Japanese, Chinese etc, as these are represented in 8-bit codes.
- 3. SMTP services may reject mails having size greater than a certain size.
- 4. SMTP cannot handle non-textual data such as pictures, images, and video/audio content.

# MIME – سرآیند

- قالب کلی سرآیند بصورت Keyword: Value است.



# MIME – سرآیند
## Content-type

| Type | Sub type | Description |
|---|---|---|
| Text | Plain | Unformatted text in US ASCII ISO 8859. |
| Image | jpeg | Image in JPEG Format. |
| | gif | Image in GIF format. |
| Video | mpeg | MPEG format. |
| Audio | Basic | Single- channel encoding of voice at 8kHz. |
| Message | rfc 882 | The boby is an encapsulated message that confirms to RFC 822. |
| | partial | Large mail is fragmented. |
| | External Body | contains pointer to an object that exists elsewhere and is accessible via FTP. TFTP etc. |
| Multipart | Mixed | The different parts are independent but are to be transmitted together. They should be presented to receiver in the appear in mail message. |
| | Parallel | same as mixed but order not defined. |
| | Alternate | The different parts are alternate versions of the same information |
| | Digest | similar to mixed, but the default type/subtype of each part is message/rfc 822. |
| Applciation | Postscript | Adobe postscript . |
| | Octet-stream | General binary data consisting of 8-bit bytes (Octets). |

**Context-type: <type/subtype; parameters>**

مشخص می کند محتوای بدنه چگونه تفسیر شود.

مقدار پیش فرض : **plaintext**

# MIME – سرآیند
# Content-transfer-encoding

روش های کدگذاری مختلف

| Type | Deseription |
|---|---|
| 7- bit | The body contains The 7- bit ASCII Characters With maximum length of 1000 characters |
| 8- bit | There can be non-ASCII 8- bit characters but the maximum length of the body is limited to 1000 characters. |
| Binary | Binary 8- bit characters without limitation of 1000 characters in the body. |
| Quoted- printable | This is useful when data consists of largely printable characters. Characters in the rang decimal equivalent 33 to 61 in ASCII are represented in ASCII. Others are represented as two- digit hex representation preceded by '=' sign, Non- text characters are replaced with six -digit hex sequence |
| Base 64 | 6-bit block of input data is encoded into 8-bit block of output. |

**Content-transfer-encoding : <type>**

طریقه کدگذاری شی درون متن را توصیف می کند.

# MIME – سرآیند

**Content-id : id = <content-id>**

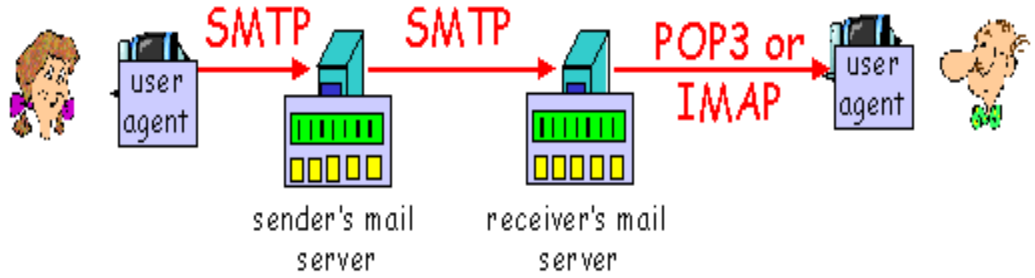
**مقدار مشخصه منحصر به فرد برای پیام**

**Content-description: <description>**

**برای داده های صدا یا تصویر با این پارامتر توضیح داده می شود.**

# قراردادهای دریافت نامه

## Mail access protocols




- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]
    - authorization (agent <-->server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored msgs on server
  - HTTP: Hotmail , Yahoo! Mail, etc.

- سرویس دهنده SMTP : به ایمیلهای ارسالی رسیدگی می کند.
- سرویس دهنده های POP3 و IMAP : به ایمیل های دریافتی رسیدگی می کنند.
- SMTP : پورت ۲۵
- POP3 : پورت ۱۱۰
- IMAP : پورت ۱۴۳

# قرارداد POP3

از فصل هفتم – قراردادهای پست الکترونیکی

# قراردادهای دریافت نامه – POP3 (Post Office Protocol)

- یک قرارداد ساده با قابلیت عملیاتی محدود است.
- دارای دو قسمت سرویس دهنده (محل ذخیره) و مشتری (گیرنده) می باشد.
- از دو عملیات حذف و دانلود در نامه های الکترونیکی استفاده می کند.
- سه حالت : احراز هویت، تراکنش و حالت بروز رسانی دارد.

✓ کاربر یک ارتباط TCP روی پورت ۱۱۰ برقرار می کند و نام و رمزعبور را برای سرور ارسال می نماید.

✓ بعد از مرحله احراز هویت وارد حالت تراکنش شده و می تواند نامه هایش را لیست کرده و دریافت نماید.

✓ در انتها با اجرای فرمان quit توسط کاربر، ارتباط وارد حالت بروزرسانی می شود.(سرور کلیه فرمانهای کاربر را انجام و ارتباط را قطع می کند.)

# قراردادهای دریافت نامه – POP3 – مراحل کار

## POP3 protocol

**authorization phase**
- client commands:
  - user: declare username
  - pass: password
- server responses
  - +OK
  - -ERR

**transaction phase**, client:
- list: list message numbers
- retr: retrieve message by number
- dele: delete
- quit

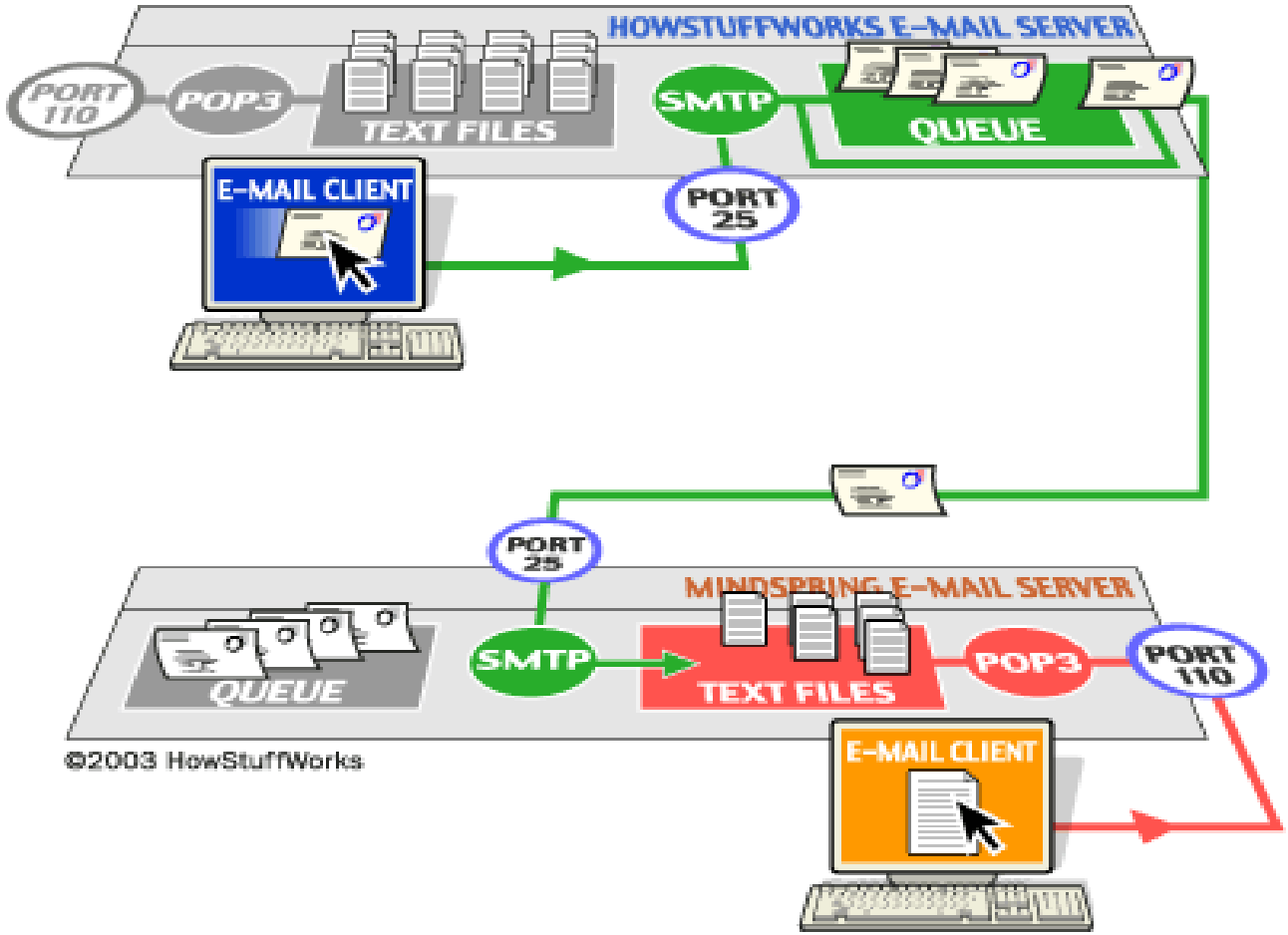
```
S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

2: Application Layer 40

- دستورها از طریق حالت احراز هویت و تراکنش انجام می گیرند، اما از طریق حالت بروز رسانی میسر نیست. (به جز دستور quit)
- پاسخ های سرور می تواند تا ۵۱۲ کاراکتر باشد.

# قراردادهای دریافت نامه – POP3 (Post Office Protocol)



# قراردادهای دریافت نامه – POP3 – دستورها

| حالت | دستور | شرح |
|---|---|---|
| حالت احراز هویت | USER | |
| | PASS | |
| | APOP | |
| | AUTH | |
| | QUIT | |
| حالت تراکنش | STAT | |
| | [msg#]LIST | |
| | RETR msg | |
| | DELE msg | |
| | NOOP | |
| | RSET | |
| | QUIT | |

# قرارداد IMAP4

از فصل هفتم – قراردادهای پست الکترونیکی

# قراردادهای دریافت نامه – IMAP4 (Internet Message Access Protocol)

- یک قرارداد پست الکترونیکی با دو تابع مشتری و سرویس دهنده است.
- سرویس دهنده های IMAP4 پیام ها را برای کاربران بیشتری نسبت به POP3 ذخیره می کند.
- IMAP4 به کاربر اجازه می دهد تا چندین صندوق پستی داشته باشد و هرکدام را در هر مکانی انتخاب کند.
- نامه ها همیشه توسط این قرارداد، روی سرور نگاه داشته شده و کپی نامه ها را برای کاربر می فرستد.
- IMAP4 برخلاف POP3 فقط قراردادی برای دریافت نامه نیست، بلکه می تواند نامه را نیز ارسال کند.
- IMAP4 به کاربران اجازه می دهد تغییرات را هم در موقع اتصال به سرور و هم موقع قطع آن اعمال نمایند.

# قراردادهای دریافت نامه – مقایسه بین IMAP4 و POP3

| Feature | POP3 | IMAP |
|---|---|---|
| Protocol defined in | RFC 1939 | RFC 2060 |
| TCP port used | 110 | 143 |
| E-mail stored on | user's PC | Server |
| e-mail is read | offline | on-line |
| Connection time required | little | much |
| Use of sever resources | minimal | extensive |
| Multiple mailboxes | No | Yes |
| Mailboxes backup by | user | ISP |
| Good for mobile users | No | Yes |
| User control over downloading | little | great |
| Partial message download facility | No | Yes |
| Simple to implement | Yes | No |
| Widespread support | Yes | Growing |

# قراردادهای دریافت نامه – سه مدل پایه ای IMAP4

- قرارداد IMAP4 در سه مدل پایه ای پیاده سازی شده است :
  - ✓ Offline : مشابه POP3 می باشد.
  - ✓ Online : برعکس مدل Offline
  - ✓ Disconnected : ترکیبی از دو مدل قبلی است.

# قراردادهای دریافت نامه – مقایسه سه مدل پایه ای IMAP4

| ویژگی ها | Offline | Online | Disconnected |
|---|---|---|---|
| استفاده از چندین مشتری بطور همزمان | نه | بله | بله |
| حداقل استفاده از زمان اتصال سرویس دهنده | بله | نه | بله |
| حداقل استفاده از زمان اتصال سرویس دهنده | بله | نه | نه |
| حداقل استفاده از منابع مشتری | نه | بله | نه |
| استفاده از چندین صندوق پستی دور | نه | بله | بله |
| راه اندازی سریع | نه | بله | نه |
| پردازش نامه وقتی که کاربر Online نیست | بله | نه | بله |

# قراردادهای دریافت نامه – حالت های IMAP4

- **حالت بدون احراز هویت** : در این حالت، مشتری هنوز احراز هویت نشده است.
- **حالت احراز هویت** : مشتری خود را به سرویس دهنده معرفی نموده و باید برای ادامه کار، یک صندوق پستی انتخاب نماید.
- **حالت انتخابی** : در این حالت، یک صندوق پستی با موفقیت انتخاب شده و می توان کارهای موردنیـاز را روی نامه های داخل صندوق انجام داد.
- **حالت خروج (Log out)** : در این حالت، ارتباط با درخواست مشتری و یا هر دلیل دیگری خاتمه می یابد.

# قراردادهای دریافت نامه – لیستی از حالت ها و دستورات IMAP4

- صفحه ۲۷۵