

شبکه های کامپیوتری برنامه های کاربردی شبکه - قسمت 2

سیامک سرمدی، وحید سلوک

1

فهرست مطالب

- پست الکترونیکی
- پروتکل های SMTP, POP3, IMAP
- سرویس دهنده نام دامنه (DNS)
- برنامه های کاربردی P2P
- برنامه نویسی سوکت

2

پست الکترونیکی

3

پست الکترونیکی

□ **پست الکترونیکی (Electronic Mail):** سرویس ارسال پیامی است که از ابتدای اینترنت مورد استفاده بوده و هنوز از پر استفاده ترین سرویس های اینترنت است.

■ مزایا:

- سریع
- ارزان
- توزیع آسان

■ امکانات جدید:

- ضمیمه ها (attachments)
- متن html
- تعبیه عکس در متن

4

پست الکترونیکی

دارای سه جزء اصلی

- نرم افزار کاربر (User Agent)**
 - اجازه خواندن، ارسال، پاسخ و ارجاع ایمیل را میدهد.
 - Apple Mail و Microsoft outlook
- سرویس دهنده:**
 - صف پیام (سرور smtp):** ایمیل را از کاربر دریافت کرده، در صف پیام قرار داده و بتدریج ارسال میکند.
 - صندوق پستی (سرور Pop3 یا IMAP):** پیام های دریافتی کاربر روی آن ذخیره میشوند و هنگام مراجعه نرم افزار کاربر، به کاربر ارائه میشوند.
- پروتکل انتقال (و صندوق پستی):**
 - SMTP اصلی ترین پروتکل پست الکترونیک، برای انتقال ایمیل از "کاربر به سرویس دهنده" و "مابین دو سرویس دهنده"
 - معماری سرویس دهنده-مشتري، و با پروتکل انتقال TCP
 - انتقال دهنده: مشتري (کلاينت)
 - دریافت کننده: سرویس دهنده (سرور)

Key:
 Outgoing message queue
 User mailbox

5

پروتکل ساده انتقال نامه (SMTP) – Simple Mail Transfer Protocol

برای انتقال ایمیل از برنامه کاربر به سرور فرستنده، و همچنین برای ارسال ایمیل از سرویس دهنده ارسال کننده به سرویس دهنده دریافت کننده استفاده می شود.

دارای سه فاز عملیاتی

1. **اتصال و دست تکانی:** بعد از برقراری اتصال TCP مشتري و سرویس دهنده خود را معرفی کرده و مشتري اطلاع میدهد که از چه کاربری و برای چه کاربری میخواهد ایمیل تحویل دهد و سرویس دهنده اطلاع میدهد که آیا ایمیل را تحویل خواهد گرفت یا نه (دست دهی). اگر سرویس دهنده، گیرنده ایمیل را شناسد از دریافت ایمیل خودداری میکند.

2. **انتقال پیام**

3. **قطع**

بسیار شبیه به HTTP

مبتنی بر مدل فرمان - پاسخ

فرمان: پیام های 7 بیتی (ASCII)

پاسخ: کد وضعیت و متن

6

پست الکترونیک – مراحل ایجاد توسط فرستنده، تا خواندن توسط گیرنده

1. ایمیل در برنامه کاربر (مثلا MS-Outlook یا یک برنامه تحت وب) ایجاد میشود.
2. ایمیل از برنامه کاربر با پروتکل SMTP به سرور ایمیلی که کاربر فرستنده روی آن حساب ایمیل دارد منتقل میشود.
3. ایمیل در سرور فرستنده، در صف ارسال قرار میگیرد و ایمیل های صف ارسال به تدریج ارسال میشوند.
4. ایمیل با پروتکل SMTP از سرور ایمیل فرستنده به سرور ایمیلی که گیرنده روی آن حساب دارد منتقل میشود و در صندوق پیام گیرنده ذخیره میشود.
5. اگر سرور فرستنده نتواند ایمیل را به سرور گیرنده منتقل کند، بصورت دوره ای، مثلا هر نیمساعت و حداکثر تا چند روز سعی مجدد میکند. اگر تا چند روز تحویل پیام به سرور گیرنده ممکن نشود، سرور فرستنده با برگشت زدن ایمیل (همراه با پیامی مبنی بر علت برگشت) این مسئله را به کاربر فرستنده اطلاع داده، و پیام را از صف ارسال حذف میکند.
6. گیرنده با کمک برنامه کاربر، با پروتکل های دسترسی به صندوق پیام (مانند POP3 یا IMAP) به سرور وصل شده و ایمیل را دریافت میکند.

7

پست الکترونیک – مراحل دست دهی و انتقال ایمیل بین دو سرور

- متن پیامهای رد و بدل شده بین مشتری (client) و سرویس دهنده (server) در یک جلسه انتقال نمونه در تصویر زیر نمایش داده شده است. هر پیام (دستور یا پاسخ) با کارکترهای CR و LF به پایان میرسد.
- نکته: هدر subject اختیاری است.

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

8

پست الکترونیک – مثال 2 – فرمت بدنه پیام

```
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.com
S: 250 smtp.example.com, I am glad to meet you
C: MAIL FROM:<bob@example.com>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.com>
C: To: Alice Example <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 Jan 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines
in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
{The server closes the connection}
```

- در مثال بعدی تفاوت های جزئی در توضیحات پیامهای پاسخ (و نه در کد وضعیت) مشاهده میشود.
- در این مثال پیام ارسال شده از هدر و بدنه تشکیل شده
- هدر با یک خط خالی از بدنه مجزا شده
- فرمت هدر و متن پیامهای SMTP در یک RFC جداگانه به شماره 5322 آمده است.
- به تفاوت بین دستورات RCPT TO و RCPT FROM و فیلدهای هدر در بدنه پیام توجه کنید. این هدرها جزو بدنه پیام بوده و با دستورات SMTP تفاوت دارند (اگرچه دو مورد فوق همپوشانی دارند)

9

مقایسه: HTTP vs. SMTP

□ HTTP:

- اتصال پایا و غیر پایا
- نحوه ارتباط: Pull
- درخواست:
- پیام های فرمان با هر قالبی
- داده (در پاسخ)
- هر شیء در یک پیام پاسخ جداگانه

□ SMTP:

- فقط اتصال پایا
- نحوه ارتباط: Push
- دستورات (مشابه درخواست ها):
- پیام های درخواست با قالب 7 بیت ASCII کد میشوند.
- داده (در دستور)
- کل محتوی در یک پیام

10

پروتکل دسترسی نامه

- تا اوایل دهه 1990 روش دسترسی به ایمیل به این شکل بود که روی کامپیوتر کاربر (معمولا یک سرور محلی در محل کار کاربر)، یک نرم افزار سرویس دهنده SMTP نصب بود و ایمیل ها بطور مستقیم به کامپیوتر کاربر ارسال و به شکل فایل های محلی در آن ذخیره میشد (که با برنامه ای قابل مشاهده بودند).
- با گسترش استفاده از کامپیوترهای شخصی و با توجه به اینکه این کامپیوترها خیلی وقت ها خاموش بوده و از آدرس های متغییر IP استفاده میکردند، روش بالا دیگر کارایی نداشت.
- بنابراین پروتکل هایی ساخته شد تا برنامه کاربر بتواند به سروری که ایمیل ها در آن ذخیره میشوند متصل شود و ایمیل ها را دریافت کرده و یا ببیند.

11

پروتکل دسترسی نامه

- برای دریافت یا دسترسی به ایمیل ها توسط کاربر دریافت کننده از کامپیوتر سرویس دهنده (که کاربر روی آن حساب دارد) بکار میروند.

POP3 (Post Office Protocol v3)

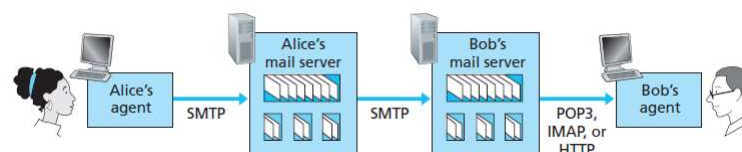
- اتصال روی پورت 110
- دریافت مجوز (لاگین)، دسترسی و انتقال، به روز رسانی

IMAP (Internet Mail Access Protocol)

- امکان خواندن نامه بدون انتقال به دستگاه مشتری (ایمیل ها روی سرویس دهنده ذخیره شده و باقی می مانند)
- دارای ویژگی های بیشتر، تغییر پیام های ذخیره شده

HTTP

- کلیه سرویس دهندگان مبتنی بر وب



12

پروتکل POP3

- یک پروتکل بسیار ساده دسترسی به ایمیل است که در RFC1939 تشریح شده است.
- به علت سادگی پروتکل، توانایی های آن نیز محدود است.
- از پروتکل لایه انتقال TCP روی پورت 110 استفاده میکند.
- **فازها:** در 3 فاز ایمیل ها را دریافت میکند و به برنامه کاربر (User Agent) منتقل می نماید
- **دریافت مجوز (Login یا Authentication):** در این مرحله برنامه کاربر با ارسال نام کاربری و کلمه رمز، مجوز دسترسی به صندوق پیام را بدست می آورد.
- **دسترسی و انتقال (Transaction):** در این مرحله برنامه کاربر ایمیل ها را دریافت کرده، آنها را بر روی سرور برای حذف علامت گذاری میکند، و آمار ایمیل ها را بدست می آورد.
- **به روز رسانی (Update):** در این مرحله پیامهایی که برای حذف علامت گذاری شده اند، حذف می گردند.
- **پاسخ و کد پاسخ:** هر دستور برنامه کاربری پاسخی از سمت سرور دریافت میکند. 2 نوع پاسخ ممکن وجود دارند:
- **پاسخ +OK:** گاهی با ارسال داده از سرور به کلاینت همراه است و نشاندهنده موفقیت آمیز بودن اجرای دستور درخواستی است.
- **پاسخ ERR-:** نشاندهنده این است که اجرای دستور با شکست مواجه شده.
- در طی یک ارتباط، سرور اطلاعات وضعیتی را در مورد کاربر (مجاز ها، ایمیل های علامت گذاری شده برای حذف و ...) نگه میدارد.

13

پروتکل POP3 – فازها

- **فاز دریافت مجوز:** در این فاز برنامه کاربر، با ارسال نام کاربری و کلمه رمز مجوز دسترسی به صندوق پستی را میگیرد. با توجه به حفظ وضعیت در طول ارتباط TCP، این مجوز در اجرای دستورات بعدی (تا زمان قطع ارتباط) موثر خواهد بود.
- **فاز دسترسی و دریافت:** برنامه کاربری میتواند در دو مود "بگیر و حذف کن" یا مود "بگیر و نگهدار" کار کند.
- **مود بگیر و حذف کن:** بعد از دریافت هر ایمیل با دستور retr، دستور dele برای حذف آن ایمیل صادر میشود.
- **مود بگیر و نگهدار:** فقط دستور retr اجرا میشود و در نتیجه ایمیل بعد از دریافت توسط برنامه کاربر، حذف نمیشود. این مود در مواقعی که کاربر از چند نقطه به صندوق پستی خود متصل میشود مفید است.
- **فاز به روز رسانی:** در این فاز که پس از ارسال دستور quit توسط برنامه کاربر آغاز میشود، ایمیل هایی که برای حذف علامت گذاری شده اند، حذف میگردند.

```
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on
```

- نمونه مرحله دریافت مجوز:

14

پروتکل POP3 – نمونه یک نشست

```
S: <wait for connection on TCP port 110>
C: <open connection>
S: +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C: APOP mrose c4c9334bac560ecc979e58001b3e22fb
S: +OK mrose's maildrop has 2 messages (320 octets)
C: STAT
S: +OK 2 320
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S: <the POP3 server sends message 1>
S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
S: <the POP3 server sends message 2>
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
C: <close connection>
S: <wait for next connection>
```

- در این نشست، دریافت مجوز با دستور APOP انجام شده که معادل دو دستور user و pass است.
- دستور stat تعداد ایمیل ها و حجم کل آنها را بدست می آورد.
- دستور list سایز هرکدام از ایمیل های موجود را بدست می آورد.
- دستور retr n ایمیل شماره n را بازیابی و دریافت می کند.
- دستور dele n ایمیل n را برای حذف علامت گذاری میکند.
- دستور quit نشست را از طرف کلاینت به اتمام می رساند.

15

پروتکل IMAP و HTTP

- **IMAP:** پروتکلی برای دسترسی به صندوق پستی است که در آن معمولاً پیامها در سرور ذخیره شده و در برنامه کاربری ذخیره نمیگردند. امکانات خیلی بیشتری در مقایسه با POP3 دارد. در RFC3501 تشریح شده.
- **فولدرها:** ایمیل های تازه به فولدر inbox نسبت داده میشوند. کاربر میتواند فولدر های جدیدی درست کرده و ایمیل ها را به آنها نسبت بدهد. کاربر میتواند از کامپیوترهای مختلف از IMAP استفاده کند و در همه جا ساختار فولدر ها و ایمیل های درون آنها بطور یکسان دیده خواهد شد.
- **جستجو:** کاربر میتواند بر روی ایمیل های ذخیره شده روی سرور ، از راه دور، جستجو انجام دهد.
- **امکان دریافت اجزاء:** کاربر میتواند اجزاء مشخصی از ایمیل ها را دریافت کند. مثلاً تنها یکی از attachment ها را. این قابلیت برای مواقعی که برنامه کاربر از پهنای باند و حجم داده مجاز کمی بهره مند است، مفید باشد.
- **HTTP:** یکی دیگر از روش های دسترسی به صندوق پستی با کمک یک رابط وب (استفاده از پروتکل HTTP) است. در این روش یک وب سرور و اسکریپت هایی که رو آن اجرا میشوند، در کنار سرور ایمیل نصب میشوند. این سرور وب اجازه دسترسی به صندوق پستی و همچنین ارسال ایمیل را میدهد.
- **روش دسترسی به پیامها:** این سرورها معمولاً یا با دسترسی فایلی به ایمیل های ذخیره شده روی سرور دسترسی پیدا میکنند و یا با کمک پروتکل IMAP این دسترسی را ایجاد مینمایند.

16

سرویس نام دامنه (DNS)

17

مشخصه ها و نامها

- **مشخصه های انسانها:** انسانها با استفاده از مشخصه های مختلف قابل تشخیص از هم می باشند. برای مثال
 - نام و نام خانوادگی شناسنامه ای
 - شماره ملی
 - شماره شناسنامه
 - شماره دانشجویی
 - ...
- **مشخصه های عددی و حرفی (نمادین):** در امور مختلف، مشخصه متفاوتی از لیست بالا برای استفاده مناسب تر می باشند.
 - در محیط های کامپیوتری مشخصه های عددی کاربرد بیشتری دارند. برای مثال برای سیستم های ثبت احوال، بانکی و سلامت شماره ملی بیشتر استفاده میشود.
 - انسانها اسامی را بهتر به یاد می سپارند. بنابراین نام های حرفی (نام و نام خانوادگی) بیشتر از روشهای دیگر توسط آنها استفاده میشوند.

18

مشخصه و نام میزبان های اینترنتی و نام دامنه

- **مشخصه میزبان های اینترنتی:** میزبانها نیز همانند انسانها با مشخصه های مختلفی قابل شناسائی هستند. از جمله مشخصه های مورد استفاده برای میزبان ها بر روی اینترنت میتوان به موارد زیر اشاره کرد:
 - آدرس های لایه 4 عددی IP
 - 195.134.12.3
 - آدرس های حرفی نام دامنه
 - www.test.com
- **آدرس های مناسب برای مسیریابی:** با توجه به طول متفاوت آدرس های دامنه، پردازش زمانبرنامهای حرفی (نمادین)، و عدم وابستگی آدرس به محدوده جغرافیایی (به استثناء نام های کشوری مثل example.ir) روترها و تجهیزات لایه 4 شبکه معمولاً از آدرس های عددی IP (که طول ثابت و ساختار سلسله مراتبی دارند) استفاده میکنند.
- آدرس های نمادین (حرفی) در کاربرد های لایه 5 استفاده می شوند. این آدرس ها باید برای آدرس دهی و مسیریابی بسته ها در لایه 4، به آدرس های IP تبدیل شوند.
- **آدرس مناسب برای انسانها:** انسانها آدرس های حرفی دامنه را ترجیح می دهند زیرا به یاد سپردن آنها آسانتر است.
- **سرویس DNS:** تبدیل آدرس های حرفی (نمادین) به آدرس های IP توسط سرویس و کاربرد DNS (Domain Name Service) انجام میگردد.

19

سرویس نام دامنه (DNS)

- **نام دامنه:** آدرس های نمادین و حرفی که به جای آدرس های عددی اینترنتی (آدرس های IP) استفاده می شوند
- **انواع آدرس های نمادین:**
 - آدرس با توصیف کامل (Fully Qualified Domain Name، FQDN): آدرس کامل، قابل شناسائی در شبکه اینترنت
 - آدرس با توصیف کوتاه شده (Partially Qualified Domain Name، PQDN): آدرسی معتبر برای داخل شبکه، بدون قسمت های مربوط به شناسائی شبکه بیرون
- **سرویس نام دامنه:** با پروتکل UDP و بر روی شماره پورت 53 سرویس ارائه میکند.
- از جمله برنامه های سرور نام دامنه میتوان به برنامه BIND اشاره کرد که روی سیستم های یونیکس و لینوکس اجرا میشود.

FQDN

lib.uut.ac.ir
en.wikipedia.org
www.sandbox.com

PQDN

lib
en
www

20

سرویس نام دامنه (DNS) - مراحل تبدیل نام دامنه به آدرس IP

- **موارد استفاده:** نام دامنه معمولا در کاربردهای لایه 5 مورد استفاده قرار میگیرند.
 - برای مثال در هنگام استفاده از مرورگر وب (پروتکل HTTP)، آدرس استفاده شده برای اشیاء از یک آدرس نمادین دامنه که یک میزبان را مشخص میکند، و مسیر شیء تشکیل می گردد.
- `https://www.uut.ac.ir/index.php`
- برای اتصال به میزبان مورد نظر ابتدا باید نام میزبان (`www.uut.ac.ir`) به آدرس IP ترجمه شود.
- **مراحل تبدیل نام دامنه به آدرس IP به شرح زیر است:**
 1. کامپیوتر مشتری (که مرورگر روی آن نصب شده)، برنامه مشتری DNS را اجرا میکند.
 2. برنامه مرورگر (مثلا کروم)، آدرس میزبان را از URL درآورده و برای ترجمه به آدرس IP، به برنامه فوق میفرستد.
 3. برنامه مشتری DNS نام دامنه را برای ترجمه به یک سرور DNS که روی کامپیوتر مشتری مشخص شده میفرستد.
 4. سرور نام دامنه را پس از اتصال به سرورهای دیگر DNS بر روی اینترنت و یا از Cache خود، به آدرس IP ترجمه کرده و آدرس IP را به برنامه مشتری DNS میفرستد.
 5. برنامه مشتری آدرس IP را به کاربرد سوال کننده (مرورگر)، تحویل میدهد.
 6. مرورگر یک اتصال TCP (به پورت 80 کامپیوتر سرویس دهنده HTTP) را به آدرس IP مشخص شده برقرار میکند.
- ترجمه آدرس تاخیر قابل توجهی را به کاربردها تحمیل میکند که با استفاده از DNS Cache قابل کاهش است.*

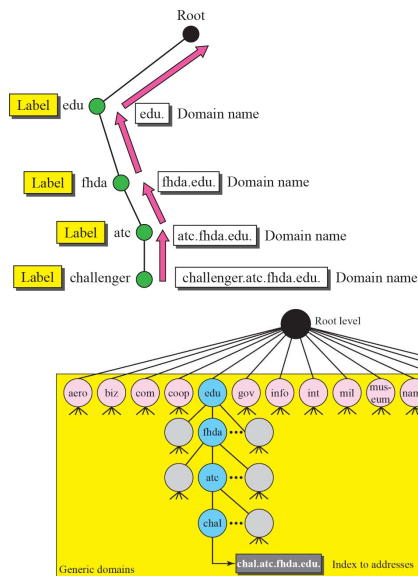
21

سرویس نام دامنه (DNS) - وظایف دیگر و پروتکل DNS

- **اسم مستعار میزبان (CNAME):** امکان تعیین نامهای مستعار (دوم، سوم و ...) برای میزبانها را فراهم میکند. با کمک سرویس DNS امکان استخراج نام اصلی میزبان و IP آن وجود دارد.
- **امکان مشخص کردن سرویس دهنده ایمیل دامنه:** از سرور DNS میتوان برای مشخص کردن این که چه میزبانی و با چه آدرس IP مسئول ارسال و دریافت ایمیل های یک دامنه است میتوان استفاده کرد (با کمک رکورد MX)
- **امکان پخش بار:** برای یک اسم مشخص مثلا `www.google.com` میتوان بیش از یک آدرس IP را مشخص کرد. به این ترتیب، سرویس دهنده DNS هربار یکی از آدرسهای IP را بعنوان ترجمه نام دامنه در نظر میگیرد. این فرآیند را میتوان برای پخش کردن درخواست های زیاد HTTP بین چند سرور وب مورد استفاده قرار داد.
- **پروتکل DNS:** در RFC های 1034 و 1035 توصیف شده است و در چندین RFC بعدی به روز رسانی گردیده است. این پروتکل، تا حدی پیچیده است و بنابراین بصورت مختصر به آن پرداخته میشود.

22

فضای نام ها اینترنتی



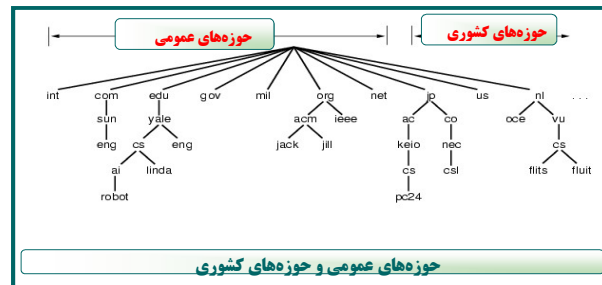
□ **فضای نام ها:** فضایی از نام های نمادین و همچنین آدرس های IP است که به نام ها نگاشت شده اند.

□ از درختی با حداکثر 128 سطح تشکیل می شود.

■ بالاترین سطح، ریشه یا سطح صفرمی باشد و بدون برچسب است.

■ باقی سطوح دارای برچسب می باشند (مثل com، edu یا fhda و ...)

■ هر گره دارای نامی است که از چسباندن برچسب های آن گره و نود های بالای آن تشکیل می شود (نام دامنه)



23

سرویس نام دامنه (DNS) – نحوه ترجمه آدرس نمادین به آدرس اینترنتی

```

hosts - Notepad
File Edit Format View Help
# Copyright (c) 1993-2006 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97      rhino.acme.com    # source server
#       38.25.63.10     x.acme.com        # x client host

127.0.0.1       localhost dev.example.com
#127.0.0.1      example.com beta.example.com

192.168.2.1     router
192.168.2.25    printer1
192.168.2.26    printer1
192.168.2.27    fax

209.85.149.103  example.com assets1.example.com
#209.85.135.104 example.com assets2.example.com

65.55.175.254   beta.example.com assets1.beta.exmple.com
#10.22.112.49   beta.example.com assets2.beta.exmple.com
  
```

□ از دیدگاه برنامه کاربردی DNS یک جعبه سیاه مترجم نام است. ولی عمل ترجمه نام فرآیندی پیچیده دارد و با دو روش کلی قابل اجرا است:

□ **روش متمرکز**

■ تعریف دستی در فایل متنی (hosts)

■ قابل استفاده در سیستم مشتری یا شبکه های خیلی کوچک

■ مسیر فایل در ویندوز:

x:\Windows\System32\drivers\etc\hosts

■ برای تغییر فایل، برنامه ویرایشگر باید با دسترسی administrator اجرا شود (Run as administrator...)

□ **روش سلسله مراتبی**

24

سرویس نام دامنه (DNS) – نحوه ترجمه آدرس نمادین به آدرس اینترنتی

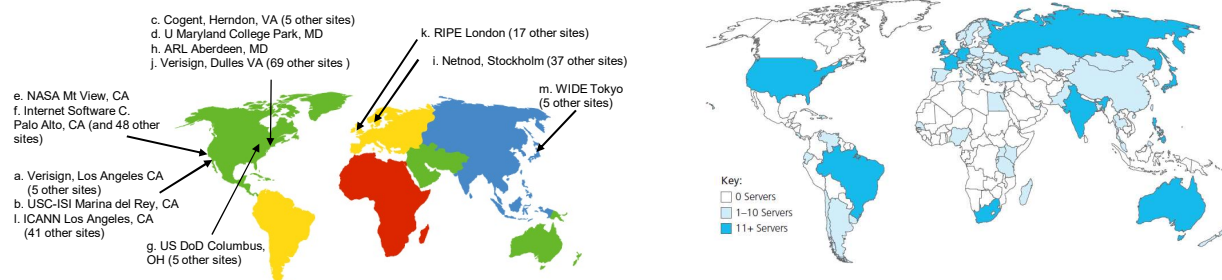
- **تحلیل گر نام متمرکز:** ساده ترین روش ممکن، ایجاد و نگهداری یک و یا چند سرور است که لیستی از تمام میزبانها و آدرس IP آنها را نگهداری میکنند. این روش ایرادات زیر را دارد:
 - ایجاد تک نقطه خرابی (Single point of failure)
 - حجم زیاد درخواستها بر روی سرور مربوطه
 - ثبت وبه روزرسانی نام و آدرس IP تمام میزبانهای اینترنتی بر روی یک یا چند سرور DNS فضا، نیاز به زمان و منابع زیادی دارد.
 - حتی در صورت ایجاد تعداد بیشتری از سرورها، مورد سوم همچنان وجود دارد.
- **تحلیل گر نام سلسله مراتبی:** در این روش نگهداری و ترجمه نام ها بر روی تعداد زیادی سرور در سراسر دنیا، و بصورت سلسله مراتبی انجام میگردد.
 - به 4 نوع سرور DNS نیاز خواهد بود.
 - سرور های ریشه (root DNS servers)
 - سرور های نام دامنه سطح بالا (Top level domain, TLD servers)
 - سرور های معتبر (Authoritative DNS servers)
 - سرور های محلی (Local DNS servers)

25

سرویس نام دامنه (DNS) – انواع سرورهای نام دامنه

- **سرورهای ریشه:** حدود 400 سرور در مناطق مختلف دنیا قرار دارند که توسط 13 موسسه اداره میشوند (نام گذاری بر اساس حروف A-M). این سرورها آدرس سرورهای DNS سطح بالا (TLD) را نگهداری کرده و به پرسشهای مربوط به آنها پاسخ میدهند.
- **سطح بالا:** مسئول نگهداری دامنه های سطح بالا از جمله دامنه های کشوری (nic.ir مسئول نگهداری .ir). این سرورها آدرس IP میزبان های DNS دامنه های محلی را نگهداری میکنند.
- **معتبر:** متعلق به سازمانها و شبکه های کوچک هستند. معمولاً آدرس میزبانهای یک دامنه را نگه میدارند. ممکن است دوباره ساختار سلسله مراتبی داشته باشند.

13 root name “servers” worldwide



26

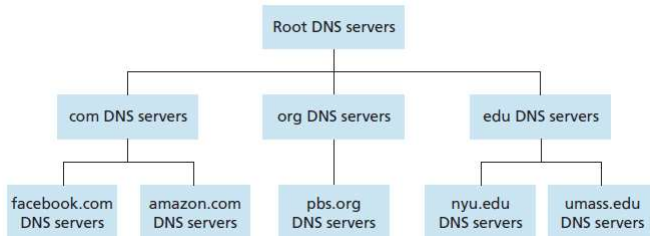
سرویس نام دامنه (DNS) – انواع سرورهای نام دامنه

- **محلی:** نوع چهارم سرور های DNS است که در ساختار درختی رورهای DNS قرار نمیگیرد.
- هر ISP یا سازمان یک یا چند سرور محلی را ایجاد میکند. هنگامی که یک کامپیوتر به شبکه وصل میشود، آدرس این سرورها (با کمک DHCP یا بطور دستی) روی آن کامپیوتر تنظیم میشود.
- هرگاه بر روی یک کتامپیوتر نیاز به ترجمه نام دامنه به IP باشد، برنامه مشتری DNS این درخواست را به سرور محلی منتقل میکند و سرور محلی با ارتباط با سرورهای سلسله مراتبی DNS بر روی اینترنت، آنرا ترجمه میکند.
- معمولاً نامهای ترجمه شده تا مدتی Cache می شوند.

27

سرویس نام دامنه (DNS) – نحوه ترجمه آدرس نمادین به آدرس اینترنتی

- برای اینکه متوجه شویم این 4 نوع سرور چگونه با هم ارتباط برقرار میکنند به مثال زیر توجه کنید.
- فرض کنید یک برنامه مشتری DNS میخواهد آدرس IP نام دامنه **www.amazon.com** را بدست بیاورد.
- کلاینت DNS ، به یک سرور DNS محلی (Local) وصل شده و درخواست خود را به برای ترجمه به آن ارائه میکند.
- سرور محلی به یکی از سرور های ریشه وصل میشود و آدرس سرور های نام دامنه سطح بالای **.com** را بدست می آورد
- سرور محلی به یکی از سرور های نام دامنه سطح بالای **.com** وصل شده و آدرس سرور DNS معتبر دامنه **amazon.com** را سوال میکند و آدرس IP آنها را بدست می آوزد (معمولاً حداقل 2 آدرس)
- سرور محلی به یکی از سرورهای DNS دامنه **amazon.com** وصل شده و آدرس میزبان **www** را سوال کرده و بدست می آورد.

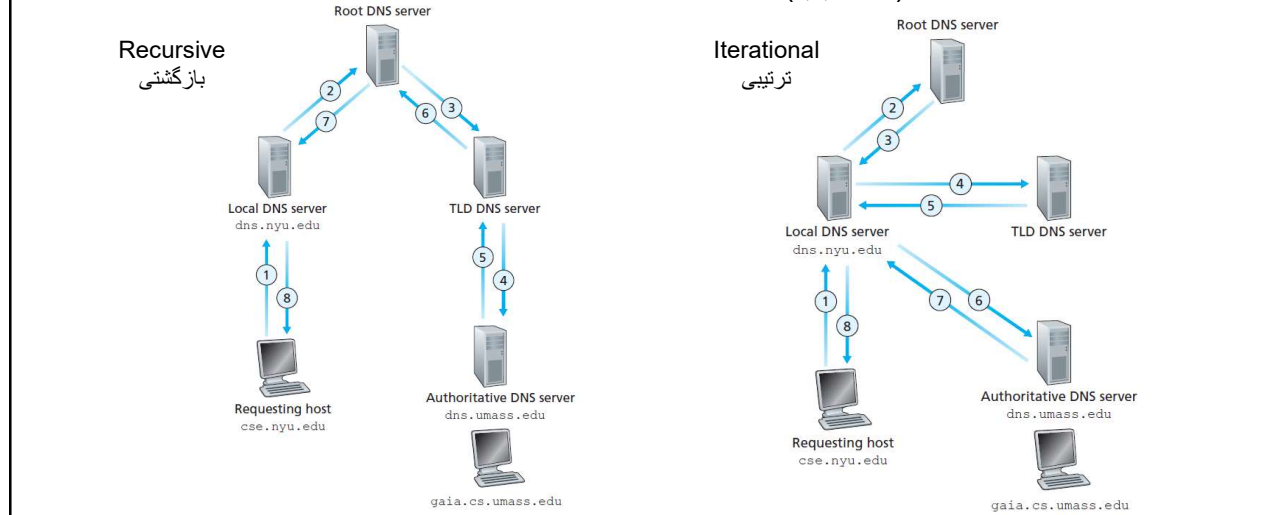


- سرور محلی آدرس را توسط برنامه مشتری DNS به برنامه سوال کننده (مثلاً Chrome) تحویل می گردد.

28

سرویس نام دامنه (DNS) – نحوه ترجمه آدرس نمادین به آدرس اینترنتی

- علاوه بر روش ترتیبی بحث شده (سمت راست، ارتباط با DNS سرورهای لایه های مختلف) امکان ترجمه با روش بازگشتی نیز وجود دارد (سمت چپ) ولی در عمل از روش ترتیبی استفاده میشود.



29

پایگاه داده DNS

Common DNS Record Types	
Record	Description
A	Address record (IPv4)
AAAA	Address record (IPv6)
CNAME	Canonical Name record
MX	Mail Exchanger record
NS	Nameserver record
PTR	Pointer record
SOA	Start of Authority record
SRV	Service Location record
TXT	Text record

- **رکورد:** آیتم ها یا رکوردهای موجود در سرور DNS معمولاً در فایل هایی ذخیره میشوند. معمولاً رکوردهای مربوط به هر نام دامنه سطح بالا (zone) در یک فایل مجزا ذخیره میشوند.

- **فیلد:** هر رکورد دارای چند فیلد است که عبارتند از نام فیلد (Name)، زمان اعتبار (TTL)، نوع فیلد (Type) و مقدار (Value).

انواع فیلد:

- A: نام میزبان
- AAAA: نام میزبان IPv6
- CNAME: نام مستعار
- MX: میزبان ایمیل دامنه
- NS: سرور نام دامنه مربوط به این حوزه یا دامنه معتبر
- TXT: فیلد متنی (برای کاربردهای مختلف)
- SRV: محل یا میزبان ارائه کننده یک سرویس خاص (مثلاً پیام رسان)
- SOA: مشخصات محل، مسئول مدیریت DNS این دامنه و تنظیمات

30

پایگاه داده DNS

مثال زیر، محتویات فایل ذخیره اطلاعات دامنه example.com در سرور DNS از نوع Bind را نشان میدهد.

```
$ORIGIN example.com.
@
      3600 SOA      ns1.hosting.net. (
                        admin.hosting.net      ; address of responsible party
                        2016072701              ; serial number
                        3600                    ; refresh period
                        600                    ; retry period
                        604800                 ; expire time
                        1800                   ; minimum ttl
      86400 NS      ns1.hosting.net.
      86400 NS      ns2.hosting.net.
      14400 MX      10 mail.example.com.
      14400 A       204.13.248.106
      14400 TXT      "v=spf1 includespf.dynect.net ~all"
mail  14400 A       204.13.248.106
vpn   14400 A       216.146.45.240
erp   14400 A       216.146.46.10
office 14400 A      216.146.46.11
www   14400 CNAME   example.com.
ftp   14400 CNAME   example.com.
```

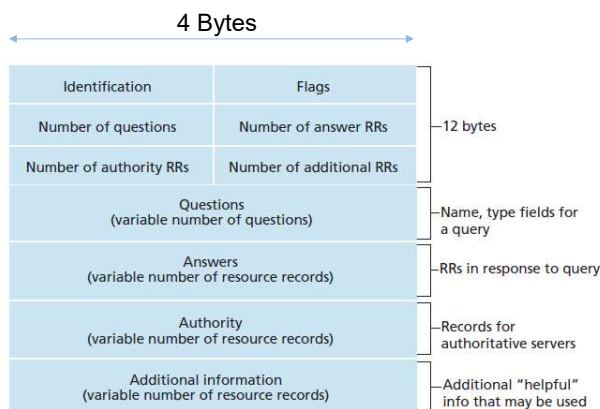
```
@ IN SOA {primary-name-server} {hostmaster-email} (
      {serial-number}
      {time-to-refresh}
      {time-to-retry}
      {time-to-expire}
      {minimum-TTL} )
```

فرمت رکورد SOA

31

ساختار پیامهای DNS

DNS از دو نوع پیام سوال (Query) و پاسخ (reply) استفاده میکند که فرمت هر دو در تصویر پایین نمایش داده شده است.



Identification: مشخصه سوال (در پاسخ هم می آید)

Flag: از چند فیلد 1 بیتی تشکیل شده از جمله:

■ مشخص کننده نوع: سوال است یا جواب

■ مشخص کننده نوع جواب دهنده: معتبر یا غیره

■ مشخص کننده نوع اجرا: ترتیبی یا بازگشتی

■ فیلدهایی برای مشخص کردن تعداد شوال یا جوابها

■ قسمت پرسش ها (نام مورد سوال، نوع مورد سوال)

■ قسمت پاسخ ها (فیلدهای رکوردهای مورد نظر)

■ رکوردهای سرور معتبر: سایر سرورهای معتبر

■ اطلاعات اضافه: رکوردهای مفید

32

برنامه نویسی سوکت

33

برنامه نویسی شبکه

- در این قسمت از فصل برنامه نویسی کاربردهای لایه کاربرد با استفاده از پروتکل های لایه انتقال TCP و UDP بررسی خواهد شد.
- برای برنامه نویسی از زبان پایتون استفاده خواهد شد زیرا:
 - مفاهیم برنامه نویسی سوکت را به راحتی نشان میدهد
 - نیاز به تعداد خطوط کمتری از کد برای ایجاد یک کاربرد است.
 - دستورات حتی برای یک برنامه نویس تازه کار به راحتی قابل فهم هستند.
- در صورتیکه به پیاده سازی کاربردها با زبان برنامه نویسی جاوا علاقمند هستید میتوانید برای دیدن نمونه کدهای جاوا، به وبسایت کتاب مراجعه کنید.

34

برنامه نویسی سوکت

□ **سوکت:** واسطه (دروازه) بین نماینده لایه کاربرد (برنامه یا پردازش) و لایه انتقال است.

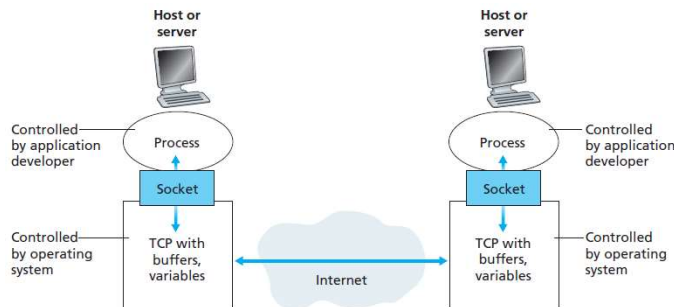
□ زیرساخت لازم برای اجرای برنامه های کاربردی

□ برقراری ارتباط بین سرویس دهنده-مشتري توسط سوکت

□ انواع سوکت (لایه انتقال)

□ **UDP:** بدون برقراری اتصال، غیر قابل اطمینان

□ **TCP:** قابل اطمینان – اتصال با پایه جریان بایت



35

کاربرد نمونه UDP - کاربرد اکو (با uppercase)

□ **پروتکل لایه انتقال:** در این قسمت برنامه های مشتری و سرویس دهنده ای با پروتکل لایه انتقال UDP ایجاد میگرد.

□ **پروتکل ارتباطی لایه کاربرد:** پروتکل بسیار ساده ارتباطی بین مشتری و سرور به شرح زیر است:

1. مشتری: دریافت یک سطر کاراکتر از طریق صفحه کلید
2. مشتری: ارسال کاراکترها به سرویس دهنده
3. سرویس دهنده: دریافت کاراکترها و تبدیل به حروف بزرگ
4. سرویس دهنده: ارسال کاراکترهای جدید به مشتری
5. سمت مشتری: دریافت کاراکترهای جدید و نمایش در صفحه نمایش

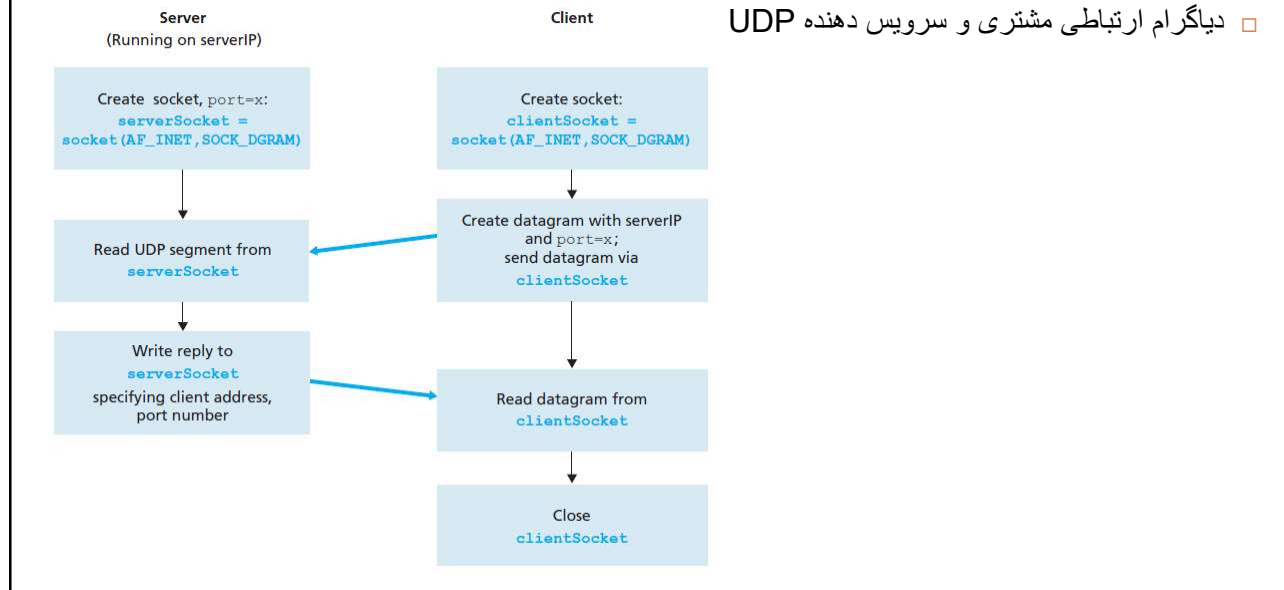
□ **خصوصیات پروتکل UDP:**

- به اتصال میزبان ها (دست تکانی) پیش از ارسال داده نیازی نیست.
- آدرس اینترنتی میزبان IP توسط فرستنده در هر بسته (Packet) لایه 3 قرار داده میشود. همچنین آدرس پردازش (پورت) گیرنده در دیتاگرام یا بسته UDP قرار گرفته در داخل بسته IP اشاره شده، قرار داده می شود.
- استخراج آدرس IP میزبان و پردازش ارسال کننده از داخل بسته های فوق در میزبان گیرنده ممکن است.

36

کاربرد UDP - ارتباط سوکت با UDP

□ دیاگرام ارتباطی مشتری و سرویس دهنده UDP



37

سوکت UDP - سمت مشتری

□ در کد زیر، ساختار کد مشتری (client) پروتکل مورد نظر نمایش داده شده است.

□ برای هر بسته، آدرس و پورت مقصد درج شده و بسته ارسال میشود

```

from socket import *
serverName = 'localhost'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
message = input("Input lowercase sentence: ")
clientSocket.sendto(message.encode(),(serverName, serverPort))
receivedMessage, serverAddress = clientSocket.recvfrom(2048)
print(receivedMessage.decode())
clientSocket.close()
  
```

اضافه کردن همه توابع و ... ماجول سوکت
متغییری برای آدرس یا نام سرور
متغییری برای شماره پورت سرویس دهنده
ایجاد سوکت IPv4 و UDP (Datagram)
دریافت پیام از کاربر
ارسال دیتاگرام به IP خاص و پورت سرور (برای هر بسته دیتاگرام)
دریافت پاسخ با سائز بافر 2048 (آدرس سرور و پورت هم دریافت میشود)
چاپ پیام بعد از تبدیل آرایه بایت به رشته
بستن سوکت

38

سوکت UDP - سمت سرویس دهنده

□ در کد زیر، ساختار کد سرویس دهنده (server) نمایش داده شده است.

from socket import *	اضافه کردن همه توابع و ... ماحول سوکت
serverPort = 12000	متغیری برای شماره پورت سرویس دهنده
serverSocket = socket(AF_INET, SOCK_DGRAM)	ایجاد سوکت IPv4 و UDP (Datagram)
serverSocket.bind(("", serverPort))	چسباندن سوکت به پورت مشخص سرور
print("The server is ready to receive...")	چاپ پیام
while True:	حلقه بی پایان
message, clientAddress = serverSocket.recvfrom(2048)	دریافت پیام با سائز بافر 2048 (آدرس کلاینت و پورت هم دریافت میشود)
modifiedMessage = message.decode().upper()	تبدیل پیام آرایه بایت به رشته و تبدیل به حروف بزرگ
serverSocket.sendto(modifiedMessage.encode(), clientAddress)	ارسال پیام پاسخ دیتاگرام (IP و پورت برای هر بسته دیتاگرام ذکر میشود)

39

برنامه نویسی سوکت با TCP

□ **سرویس دهنده:**

- یک سوکت پذیرنده در حالت Listen قرار میگیرد.
- برای تبادل اطلاعات با هر مشتری جدید یک سوکت جدید ایجاد شده و مشتری به آن مرتبط میشود (سوکت پذیرنده به پذیرش اولیه ارتباط ها ادامه میدهد).
- امکان برقراری ارتباط با بیش از یک مشتری (با سوکت های مجزا) وجود دارد.

□ **مشتری:**

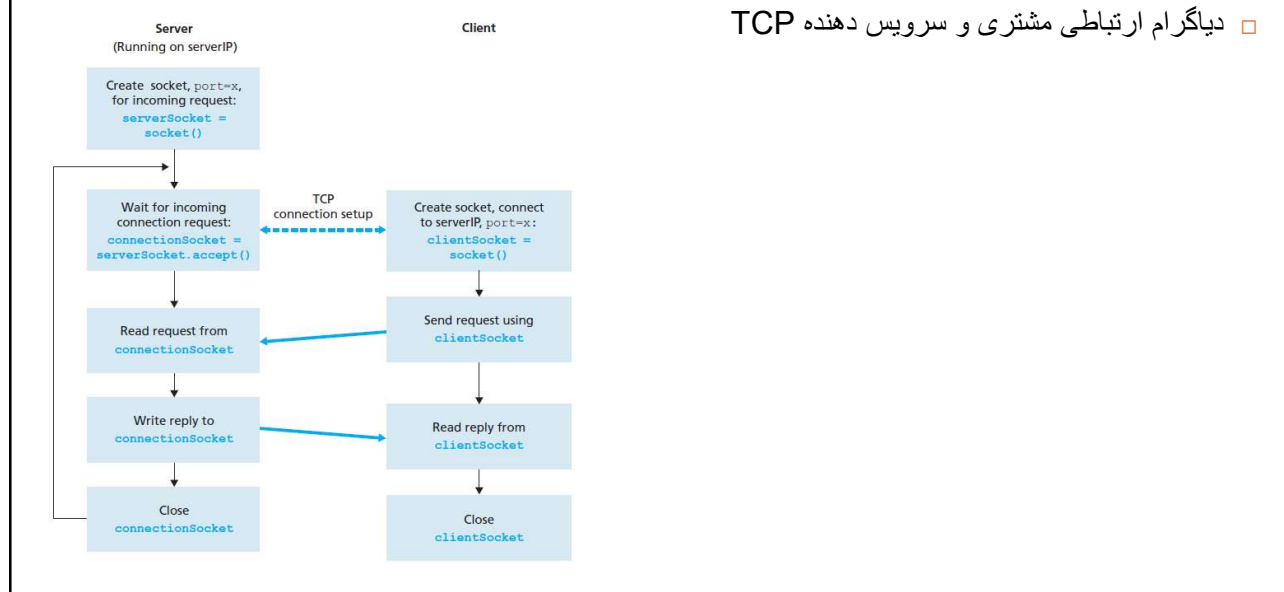
- در ابتدا نیاز ایجاد یک سوکت TCP و آغاز یک اتصال (connection) به آدرس و پورت سرور می باشد.
- قبل از تبادل داده، در سمت مشتری یک سوکت مجزا برای ارتباط با این مشتری ایجاد میشود. البته درخواست ابتدایی روی سوکت اصلی سرور (پورت استاندارد) ارسال میگردد.
- در انتهای تماس، کلاینت پیام پایان اتصال را به سرور ارسال میکند (در پیاده سازی ارائه شده)

□ **نکته:** سوکت TCP امکان ارسال داده بین سرویس دهنده و مشتری (رو طرفه) بصورت جریان بایت مرتب برقرار می کند. بایت ها با ترتیب ارسال شده تحویل گیرنده می شوند.

40

ارتباط سوکت با TCP

□ دیاگرام ارتباطی مشتری و سرویس دهنده TCP



41

سوکت TCP – سمت مشتری

- در کد زیر، ساختار کد مشتری (client) پروتکل مورد نظر نمایش داده شده است.
- در ابتدا یک ارتباط (connection) ایجاد میشود و دیگر برای تک تک بسته ها آدرس و پورت مقصد اضافه نمیشود زیرا همه پیام ها روی همان ارتباط یا جریان (Stream) برقرار شده TCP ارسال میگردند.

```

from socket import *
serverName = 'localhost'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input('Input lowercase sentence: ')
clientSocket.send(sentence.encode())
receivedSentence = clientSocket.recv(1024)
print('From Server: ', receivedSentence.decode())
clientSocket.close()
  
```

اضافه کردن همه توابع و ... ماجول سوکت
 متغییری برای آدرس یا نام سرور
 متغییری برای شماره پورت سرویس دهنده
 ایجاد سوکت IPv4 و TCP (Stream)
 برقراری اتصال به سرور (با آدرس و پورت مشخص)
 دریافت پیام از کاربر
 ارسال پیام به طرف مقابل اتصال (بدون ارائه دوباره مشخصات گیرنده)
 دریافت پاسخ با سائز بافر 2048 (بدون دریافت دوباره اطلاعات فرستنده)
 چاپ پیام بعد از تبدیل آرایه بایت به رشته
 بستن سوکت (به سرور پیام پایان اتصال میفرستد)

42

سوکت TCP – سمت سرویس دهنده

□ در کد زیر، ساختار کد سرویس دهنده (server) نمایش داده شده است.

from socket import *	اضافه کردن همه توابع و ... مایجول سوکت
serverPort = 12000	متغیری برای شماره پورت سرویس دهنده
serverSocket = socket(AF_INET, SOCK_STREAM)	ایجاد سوکت IPv4 و TCP (Stream)
serverSocket.bind(('', serverPort))	چسباندن سوکت به پورت مشخص سرور
serverSocket.listen(1)	قرار دادن سوکت اصلی در حال پذیرش
print('The server is ready to receive...')	چاپ پیام
while True:	حلقه بی پایان
connectionSocket, addr = serverSocket.accept()	دریافت اتصال و ایجاد سوکت جدید (برای پاسخ به این مشتری جدید)
sentence = connectionSocket.recv(1024).decode()	دریافت پیام با سایز بافر 2048 (بدون دریافت دوباره اطلاعات فرستنده)
capitalizedSentence = sentence.upper()	تبدیل پیام آرایه بایت به رشته و تبدیل به حروف بزرگ
connectionSocket.send(capitalizedSentence.encode())	ارسال پیام پاسخ دیتاگرام (بدون ارائه دوباره مشخصات گیرنده)
connectionSocket.close()	بستن سوکت مختص به پاسخگویی به این مشتری (سوکت اصلی همچنان در وضعیت پاسخگویی)