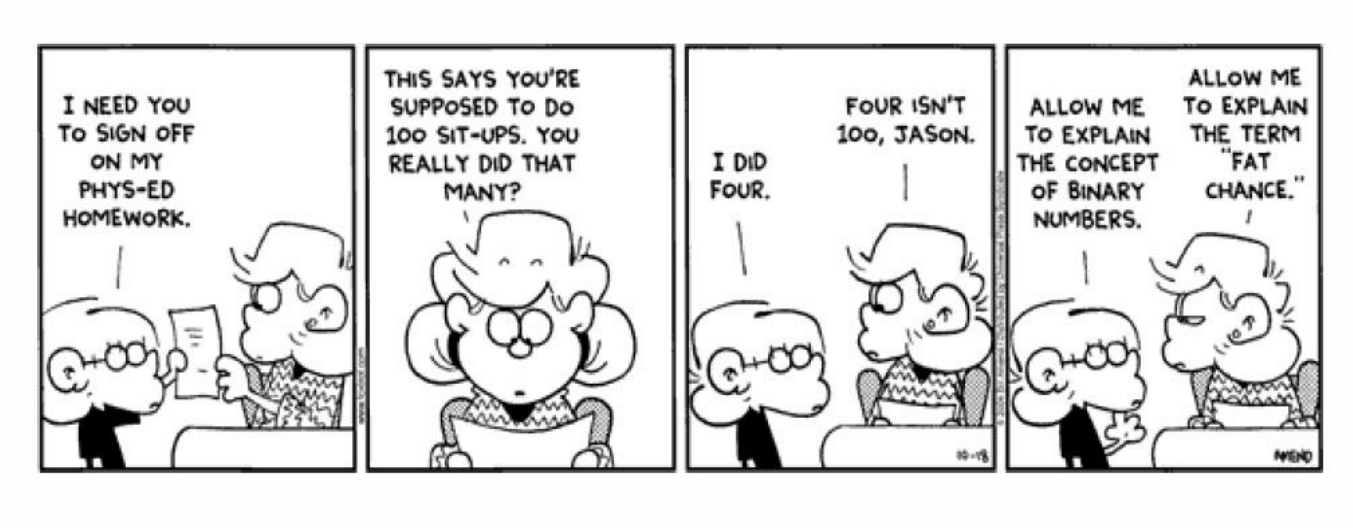
تبدیل اعداد دهدهی (Decimal) به دودویی (Binary):

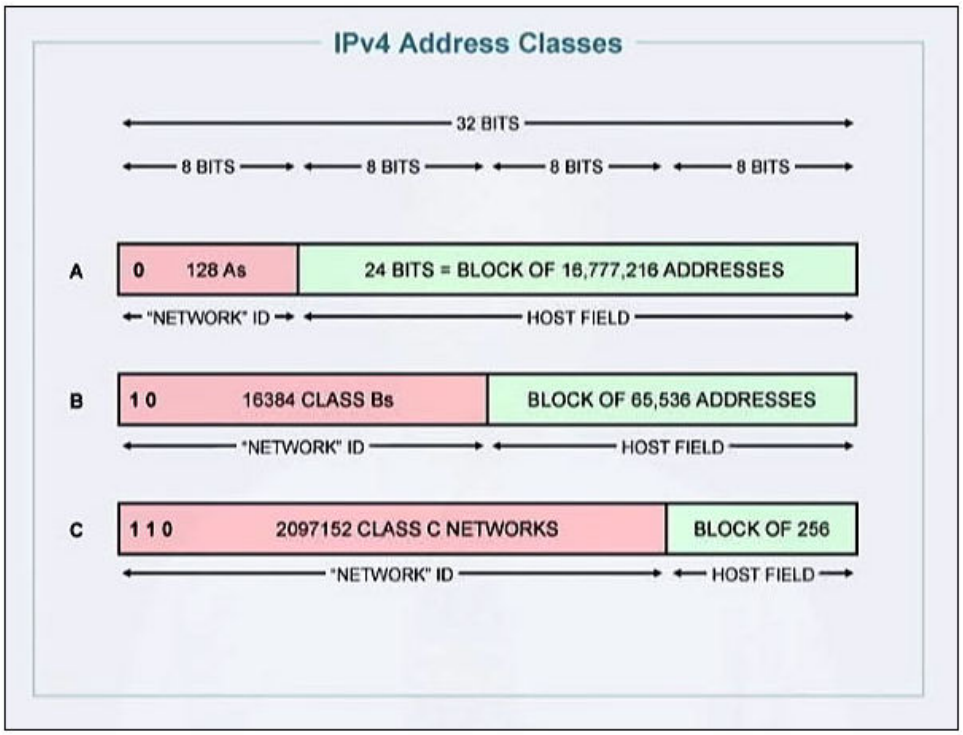
برای تبدیل اعداد دهدهی به دودویی، خیلی سریع ستون‌ها و ارزش‌های آن‌ها را بچینید و از سمت چپ شروع به تقسیم کردن عدد دهدهی به ارزش ستون‌ها کنید. (مثلاً اول ۷۵ را به ۱۲۸ تقسیم کنید) اگر عدد دهدهی بزرگ‌تر از آن ارزش بود، در آن ستون، 1 بگذارید در غیراینصورت 0 . سپس 1 یا 0 را در ارزش ضرب کنید و از عدد دهدهی کم کنید و باقیمانده را با ستون بعد مقایسه کنید... این روال را ادامه دهید تا به ستون آخر برسید.

	128	64	32	16	8	4	2	1
75	0	1	0	0	1	0	1	1
201	1	1	0	0	1	0	0	1
255	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0

تفریح: اگر توانایی‌تان در تبدیل مبناها قوی باشد، باید این جوک را درک کنید و به آن بخندید ☺ :



کلاس‌های IP :



در IP v_4 پنج کلاس مختلف قابل تصور است: کلاس‌های A، B، C، D، E.

نکته: کلاس D برای Multi casting و کلاس E برای Broad casting می‌باشد و در شبکه کاربرد خاصی ندارد.

در کامپیوتر با کلاس‌های A، B، C سر و کار داریم و کلاس‌های D و E روی هیچ کامپیوتری قابل set کردن نیست.

نکته: هر کلاس با اولین Octet از سمت چپ شناخته می‌شود.

√	.	.	.

کلاس A :

مشخصه این کلاس این است که: سمت چپ‌ترین بیت این کلاس با 0 شروع می‌شود.

کمترین IP در این کلاس (یعنی کمترین مقدارOctet چپ) 00000000 و بیشترین IP در این کلاس 01111111 است.

A	128	64	32	16	8	4	2	1	
کمترین	0	0	0	0	0	0	0	0	0
بیشترین	0	1	1	1	1	1	1	1	127

پس کلاس A در Octet سمت چپ مقداری بین 0-127 دارد.

کلاس B :

دو بیت سمت چپ این کلاس با 10 شروع می‌شود.

کمترینIP در این کلاس (کمترین مقدارOctet چپ) 10000000 و بیشترین مقدار 10111111 است.

B	128	64	32	16	8	4	2	1	
کمترین	1	0	0	0	0	0	0	0	128
بیشترین	1	0	1	1	1	1	1	1	191

پس کلاس B در Octet سمت چپ مقداری بین 191-128 دارد.

کلاس C :

سه بیت سمت چپ این کلاس با 110 شروع می‌شود.

کمترین IP در این کلاس (یعنی کمترین مقدارOctet چپ) 11000000 و بیشترین IP در این کلاس 11011111 است.

C	128	64	32	16	8	4	2	1	
کمترین	1	1	0	0	0	0	0	0	192
بیشترین	1	1	0	1	1	1	1	1	223

پس کلاس C در Octet سمت چپ مقداری بین 223 -192 دارد.

Default Subnet Mask (سابنت مَسکِ پیشفرض) :

نکته: هر کلاس یک Default Subnet Mask دارد که در زیر مشخص است:

	Default Subnet Mask
Class A	255.0.0.0
Class B	255.255.0.0
Class C	255.255.255.0

Subnet Mask به سوئیچ و روتر کمک می‌کند تا شبکه‌ای که یک کامپیوتر به آن متعلق است را شناسایی کند.

در ادامه به کاربردهای Subnet Mask آشنا خواهید شد.

Private IPs (IPهای خصوصی) :

در هر کلاس یک رِنج IP با عنوان Private IP در نظر گرفته شده است. این IPها در اینترنت تعریف نشده و غیر قابل استفاده هستند. اگر این آی.پی‌ها نباشند، در یک شبکه محلی که در یک اتاق دارید، باید تمام کامپیوترهایی که قصد اتصال به اینترنت را دارند، هر کدام جداگانه اشتراک اینترنت بگیرند. (یعنی اینطور بخواهد بود که یکی به اینترنت وصل شود و بقیه نیز از اینترنت آن استفاده کنند)

در اصطلاح گفته می‌شود IPهای خصوصی در اینترنت non-Routable هستند. (غیرقابل مسیردهی)

	Private IP Range		
Class A	10.0.0.0	-	10.255.255.255
Class B	172.16.0.0	-	172.31.255.255
Class C	192.168.0.0	-	192.168.255.255

Loop Back IP Address:

رنج IP از 127.0.0.0 تا 127.255.255.255 نیز رنج Private IP به حساب می‌آیند. و آی.پی 127.0.0.1 با نام Loop Back Address شناخته می‌شود. 127.0.0.1 یعنی همین کامپیوتر جاری. هر چه به این آی.پی بفرستید در حقیقت به کامپیوتر خودتان فرستاده‌اید.

نکته: Loop Back Addressها non-Routable و non-usable هستند. (نه قابل مسیردهی و نه قابل استفاده)

Host Address و Network Address :

اگر IP یکی از کامپیوترها در یک شبکه را با Default Subnet Mask آن AND کنیم Network Address یا (IP شبکه) به دست می‌آید.

مثال:

192 . 168 . 1 . 1

AND
11000000 . 10101000 . 00000001 . 00000001
11111111 . 11111111 . 11111111 . 00000000

پاسخ AND کردن : 11000000 . 10101000 . 00000001 . 00000000

و این یعنی : 192 . 168 . 1 . 0

به بخشی که بعد از AND کردن، تغییر نکرده است، Network Address و به بخشی که با AND کردن تغییر می‌کند، Host Address گفته می‌شود.

Host Address در حقیقت شناسه این کامپیوتر در شبکه 192.168.1 است.

پس داریم:

Class C	192	.	168	.	1	.	1
	11000000	.	10101000	.	00000001	.	00000001
Subnet Mask	255	.	255	.	255	.	0
	11111111	.	11111111	.	11111111	.	00000000
And result	192	.	168	.	1	.	0
	11000000	.	10101000	.	00000001	.	00000000
	Network Address						Host Address

نکته: دقت کنید که در هر شبکه Network Address باید برای تمام کامپیوترهای آن شبکه یکسان باشد در غیر اینصورت آن کامپیوتر در شبکه مورد نظر شما به حساب نمی‌آید و شناخته نمی‌شود.

نکته: اگر Network Address مربوط به دو کامپیوتر متفاوت باشد، چون در دو شبکه با رنج آی.پی‌های متفاوت قرار می‌گیرند، بین آن‌ها باید یک **Router** قرار گیرد تا بتوانند به یکدیگر دسترسی داشته باشند.

در هر کلاس چند هاست و چند شبکه مختلف می‌توان متصور شد؟

با توجه به اینکه در کلاس A، یک بیت سمت چپ، ثابت است و هفت بیت متغیر در اکتت سمت چپ داریم، بنابراین در این کلاس 2^7=128 شبکه مختلف می‌توان داشت و ۲۴ بیت نیز مربوط به Host Address می‌شود، پس:

در کلاس A می‌توان 2^{24}-2 (یعنی ۱۶۷۷۷۲۱۴) هاست مختلف داشت.

با توجه به اینکه در کلاس B، دو بیت سمت چپ، ثابت است و ۱۴ بیت متغیر در دو اکتت سمت چپ داریم، بنابراین در این کلاس 2^{14}=16384 شبکه مختلف می‌توان داشت و ۱۶ بیت نیز مربوط به Host Address می‌شود، پس:

در کلاس B می‌توان 2^{16}-2 (یعنی ۶۵۵۳۴) هاست مختلف داشت.

با توجه به اینکه در کلاس C، سه بیت سمت چپ، ثابت است و ۲۱ بیت متغیر در سه اکتت سمت چپ داریم، بنابراین در این کلاس 2^{21}=2097152 شبکه مختلف می‌توان داشت و ۸ بیت نیز مربوط به Host Address می‌شود، پس:

در کلاس C می‌توان 2^8-2 (یعنی ۲۵۴) هاست مختلف داشت.

نکته: در هر رنج IP ، کمترین وبیشترین مقدارها نمی‌توانند به عنوان Host Address استفاده شوند یعنی بیت‌های بخش Host نمی‌توانند همه 0 یا همه 1 باشند.

به طور مثال در رنج آی.پی *.192.168 به آی.پی 192.168.1.0 آی.پی شبکه یا همان Network Address گفته می‌شود و قابل استفاده نیست و به آی.پی 192.168.1.255 آی.پی انتشار یا BroadCast IP گفته می‌شود و باز هم قابل استفاده نیست. (اگر اطلاعاتی را به 192.168.1.255 بفرستید، به تمام کامپیوترهای شبکه فرستاده می‌شود)

Class	Leading Bits	Size of *Network Number* Bit field	Size of *Rest* Bit field	Number of Networks	Addresses per Network	Start address	End address
Class A	0	8	24	128 (2^7)	16,777,216 (2^{24})	0.0.0.0	127.255.255.255
Class B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	128.0.0.0	191.255.255.255
Class C	110	24	8	2,097,152 (2^{21})	256 (2^8)	192.0.0.0	223.255.255.255
Class D (multicast)	1110	not defined	not defined	not defined	not defined	224.0.0.0	239.255.255.255
Class E (reserved)	1111	not defined	not defined	not defined	not defined	240.0.0.0	255.255.255.255

چند مثال:

مثال ۱: Network IP مربوط به آی.پی زیر را به دست آورید:

192.149.24.191

پاسخ: با توجه به اینکه در کلاس C قرار دارد، ساب‌نت مسک پیشفرض (Default Subnet Mask) برابر است با: 255.255.255.0

به جای AND کردن می‌توانیم هر جا که 255 بود خود آن عدد و هر جا که 0 بود 0 می‌نویسیم. پس آی.پی شبکه برابر است با: 192.149.24.0

مثال ۲: Network IP مربوط به آی.پی زیر را به دست آورید:

10.10.10.10
255.0.0.0
10.0.0.0

روش دیگر: با استفاده از AND کردن بیت‌ها:

```
AND { 11111111 . 00000000 . 00000000 . 00000000
      00001010 . 00001010 . 00001010 . 00001010
      00001010 . 00000000 . 00000000 . 00000000
```

این یعنی 10.0.0.0 (Network Address).

بحث Subnetting:

همیشه رنج آی.پی‌های زیادی در اختیار نداریم که بخواهید با تغییر رنج، یک شبکه جدید ایجاد کنیم!

فرض کنید برای کشور ایران، یک رنج آی.پی به صورت 117.*.*.* از طرف مؤسسه مرکز تخصیص آی.پی به کشورها در نظر گرفته شده است.

طبیعتاً باید برای ۳۲ استان کشور شبکه ایجاد کرد به طوری که دقیقاً مشخص باشد یک آی.پی از کدام استان به اینترنت متصل شده است.

می‌دانید که آی.پی بالا در کلاس A قرار دارد یعنی می‌توان $2^{24}-2$ کامپیوتر مختلف را همزمان به اینترنت وصل کرد.

اما مشکل این است که ما چگونه باید این رنج را به شبکه‌های مختلف تقسیم کنیم؟

بله،اگر دستمان باز بود، کلاس B را انتخاب می‌کردیم و می‌گفتیم مثلاً 128.1.*.* تهران باشد. 128.2.*.* مثلاً اصفهان باشد و ...

اما الان ما فقط یک آی.پی شبکه در اختیار داریم.

اینجاست که بحث Subnetting مطرح می‌شود.

Subnetting یعنی تبدیل یک شبکه (Net) به چندین زیرشبکه (Subnet).

در این عملیات، ما با تغییر Subnet Mask تعدادی از بیت‌های مربوط به Host را قرض گرفته و به بیت‌های Network اختصاص می‌دهیم.

اگر فقط یک بیت از بخش Host را قرض بگیریم، دو شبکه مختلف می‌توانیم ایجاد کنیم.

در مثال بالا داریم:

IP: 117.0.0.0 = 01110101.00000000.00000000.00000000
Default Subnet Mask: 255.0.0.0 = 11111111. 00000000.00000000.00000000

یک بیت را برای Network Address قرض می‌گیریم. پس Subnet Mask دیگر همان ساب‌نت مسک پیشفرض (Default) نخواهد بود. بلکه سفارشی است. یعنی:

Custom Subnet Mask: 11111111.10000000.00000000.00000000 = 255.128.0.0

در این صورت یکی از شبکه‌ها رنج آی.پی‌هایشان می‌شود:

117.0.0.0 تا 117.127.255.255

و دیگری می‌شود:

117.128.0.0 تا 117.255.255.255

یعنی دو شبکه مختلف با رنج آی.پی‌های مختلف از همان آی.پی اول ایجاد کردیم! به این کار Subnetting گفته می‌شود.

حالا محاسبه کنید که برای اینکه آن رنج را به ۳۲ شبکه مختلف تقسیم کنیم، باید چند بیت را قرض بگیریم؟

واضح است که باید دید با چند بیت می‌توان ۳۲ را نمایش داد؟

با ۵ بیت.

پس باید ۵ بیت از بخش Host Address را برای شبکه قرض گرفت:

Custom Subnet Mask: 11111111.11111000.00000000.00000000 = 255.248.0.0

در شبکه بالا، تعداد زیرشبکه‌ها می‌شود: « ۲ به توان تعداد بیت‌های قرض گرفته شده »

و تعداد هاست‌ها در هر زیر شبکه می‌شود: « ۲ به توان تعداد بیت‌های باقیمانده برای بخش Host منهای ۲ » (یعنی منهای پایین‌ترین و بالاترین مقدار هاست که قبلاً گفته شد که قابل استفاده نیست)

پس:

تعداد زیرشبکه‌ها = 2^5 = ۳۲ (یعنی همان چیزی که انتظار داشتیم)

تعداد آدرس‌های هاست قابل استفاده در هر زیرشبکه: 2^{19}-2 = 524288-2 = ۵۲٤۲۸۶

فرمول‌های کلی:

تعداد زیرشبکه‌ها: 2^s

s = تعداد بیت‌های قرض گرفته شده.

تعداد هاست‌های قابل استفاده: 2^h-2

h = تعداد بیت‌های بخش Host

مثال: با توجه به ساب‌نت مسک 255.255.255.192 تعیین کنید که این شبکه چند زیرشبکه و در هر زیرشبکه چند هاست می‌تواند داشته باشد؟

	128	64	32	16	8	4	2	1
192	1	1	0	0	0	0	0	0

s = تعداد بیت های قرض گرفته شده = 2

تعداد زیر شبکه = $2^s = 2^2 = 4$

	128	64	32	16	8	4	2	1
192	1	1	0	0	0	0	0	0

h= تعداد بیت های در نظر گرفته شده برای هاست = 6

تعداد Host = $2^h = 2^6 - 2 = 62$

مثال دیگر:

Number of needed subnets **14**
Number of needed usable hosts **14**
Network Address **192.10.10.0**

Address class C

Default subnet mask 255 . 255 . 255 . 0

Custom subnet mask 255 . 255 . 255 . 240

Total number of subnets 16

Total number of host addresses 16

Number of usable addresses 14

Number of bits borrowed 4

مثال دیگر:

Number of needed subnets **1000**
Number of needed usable hosts **60**
Network Address **165.100.0.0**

Address class B

Default subnet mask 255 . 255 . 0 . 0

Custom subnet mask 255 . 255 . 255 . 192

Total number of subnets 1,024

Total number of host addresses 64

Number of usable addresses 62

Number of bits borrowed 10

تمرین بیشتر:

سعی کنید تمامی تمرینات کتاب (workbook) IP addressing and subnetting را حل کنید.

IPv6:

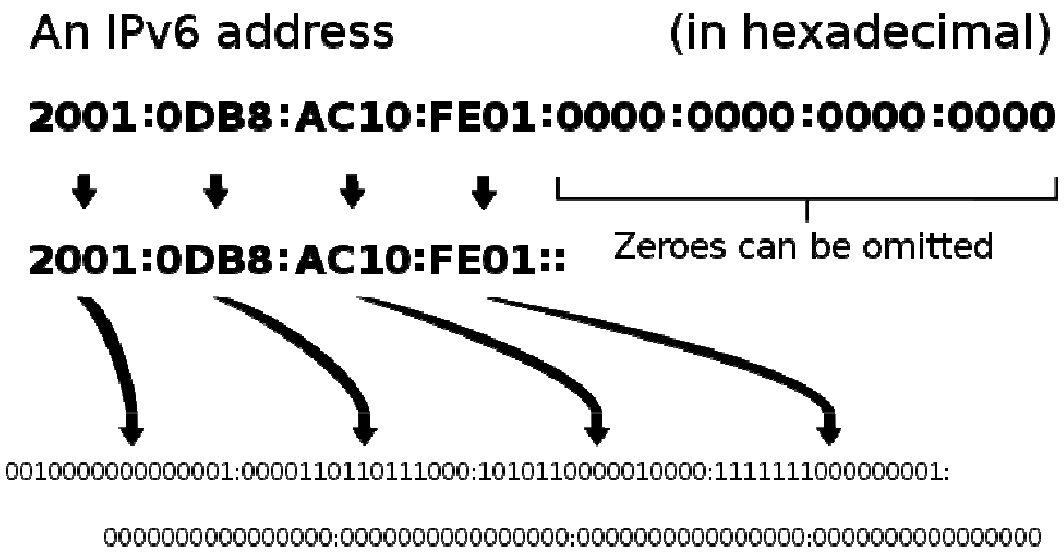
با توجه به رزرو بودن بسیاری از رنج‌های IP در ورژن ٤ (مثل آی.پی‌های Private) و با توجه به اینکه دائماً شاهد افـزایش پایانـه‌هـای متصل به اینترنت هستیم، پیش‌بینی می‌شود به زودی با کمبود IP معتبر مواجه شویم.

به همین دلیل در سال ۱۹۹۸ مؤسسه IETF ورژن ۶ از IP را معرفی کرد.

این نسخه، ۱۲۸ بیتی است در حالی که IP نسخه ٤، ۳۲ بیتی بود. در نتیجه در IPv6 تعداد 2^{128} یعنی تقریباً ۳.٤ ضـرب در 10^{38} آی.پـی مختلف می‌توان متصور شد! و این انعطاف‌پذیری و قدرت تصمیم‌گیری بهتری نسبت به ٤ میلیارد آی.پی در IPv4 به انسان می‌دهد.

برای نمایش IPv6 از مبنای ۱۶ (Hexadecimal) استفاده می‌شود.

پس ۸ بخش خواهد داشت که با : از هم جدا می‌شوند. در هر بخش ٤ عدد می‌بینید بین 0 تا F:



بخش‌هایی که 0 هستند، می‌توانند نمایش داده نشوند.

هر عدد در مبنای ۱۶ را می‌توان با ٤ بیت نشان داد ($2^4=16$) پس ۳۲ عدد داریم که هر کدام ٤ بیتـی هسـتند، یعنـی IPv6، ۱۲۸ بیتـی است.

مشهورترین پروتکل‌های شبکه:

پروتکل: قوانینی هستند که ارتباط بین شبکه‌ها را مدیریت می‌کنند.

Protocol: is a rule that governs how networks communicate.

TCP/IP Protocol: Transmission Control Protocol / Internet Protocol

TCP/IP فقط یک پروتکل نیست بلکه شامل چندین SubProtocol (زیرپروتکل) است مانند ARP ، UDP، TCP، IP و... اما اکثر مدیران شبکه به این گروه از پروتکل‌ها TCP/IP و یا حتی IP می‌گویند.

ریشه‌های TCP/IP را باید در سازمان دفاع آمریکا (Defense Of The U.S.) جست که این پروتکل را برای Advanced Research Projects Agency NETwork (شبکه آژانس پروژه‌های تحقیقاتی پیشرفته) (که مخفف آن ARPANET است) توسعه داد.

ARPANET نمونه اولیه اینترنت امروزی است.

این پروتکل به خاطر Open source بودن (استفاده از کدها به صورت آزاد و رایگان) خیلی سریع (1960-70) گسترش یافت .

برای اطمینان از انتقال صحیح یک Packet از یک نقطه به نقطه دیگر، از پروتکل TCP استفاده می‌شود. این پروتکل در لایه Transport از مدل OSI عمل می‌کند.

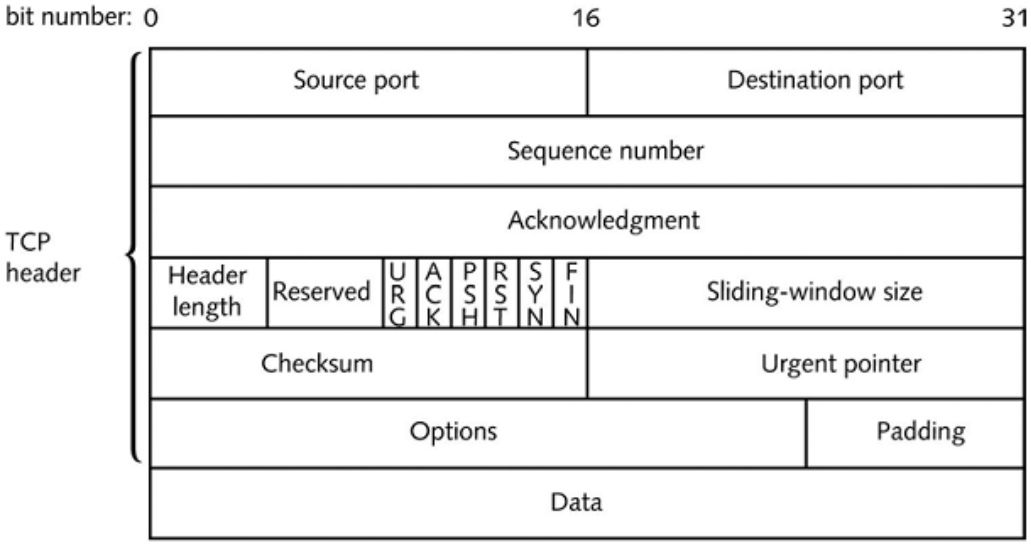
وظیفه این پروتکل در یک جمله:

Provides reliable data delivery services

یعنی سرویس‌های مربوط به تحویل موفقیت آمیز داده‌ها را فراهم می‌کند.

اجزای TCP:

در حقیقت TCP با استفاده از اجزایی به اصل داده اضافه می‌کند؛ مثل checksum و acknowledgment و شماره port مبدأ و مقصد، کار اطمینان از صحت ارسال داده‌ها را انجام می‌دهد.



Port چیست؟

همانطور که می‌دانید برنامه‌ها و سرویس‌های مختلفی مثل مرورگر، ایمیل، برنامه چت و نرم افزارهای مختلف دیگر، از طریق یک اتصال شبکه کار می‌کنند. سؤال این است که وقتی یک Packet وارد کامپیوتر شما می‌شود کامپیوتر چگونه می‌فهمد که این Packet برای برنامه ایمیل است یا مرورگر یا دیگر برنامه‌ها؟

پاسخ: هر برنامه یا سرویس در دنیا یک شماره مختص و ثبت شده دارد که به آن شماره port گفته می‌شود. این شماره توسط مؤسسه IANA (مخفف Internet Assigned Number Authority به معنی مسؤول عدد نسبت داده شده اینترنتی) تعیین می‌شود.

لیستی از پورت‌ها در آدرس مقابل قابل مشاهده است: http://www.iana.org/assignments/port-numbers

به طور مثال شماره پورت برنامه Telnet عدد ۲۳ است، شماره پورت Ftp عدد ۲۰، شماره پورت Http عدد ۸۰ و شماره پورت Smtp عدد ۲۵ است.

برخی از مهم‌ترین سرویس یا برنامه‌ها و شماره پورت آن‌ها:

Table 4-3 Commonly used TCP/IP port numbers

Port Number	Process Name	Protocol Used	Description
7	ECHO	TCP and UDP	Echo
20	FTP-DATA	TCP	File Transfer - Data
21	FTP	TCP	File Transfer–Control
22	SSH	TCP	Secure Shell
23	TELNET	TCP	Telnet
25	SMTP	TCP	Simple Mail Transfer Protocol
53	DNS	TCP and UDP	Domain Name System
69	TFTP	UDP	Trivial File Transfer Protocol
80	HTTP	TCP and UDP	World Wide Web HTTP
110	POP3	TCP	Post Office Protocol 3
119	NNTP	TCP	Network News Transport Protocol
143	IMAP	TCP	Internet Message Access Protocol
443	HTTPS	TCP	Secure implementation of HTTP

وقتی یک Packet به پروتکل TCP تحویل داده می‌شود در همان ابتدا با زدن برچسب شماره Port مشخص می‌کند که این Packet از چه برنامه‌ای آمده و به چه برنامه‌ای خواهد رفت.

Acknowledgment :

پروتکل TCP بر روی هر Packet یک عدد به عنوان Acknowledgment (ACK) (به معنی تأییدیه) قرار می‌دهد و اگر Packet با موفقیت به مقصد رسید این عدد به عنوان تأییدیه ارسال صحیح، به فرستنده ارسال می‌شود.

Checksum :

مانند روش CRC روشی برای چک کردن رخ دادن یا ندادن خطا ارائه می‌کند. در حقیقت از Header و داده‌ها محافظت می‌کند.
در مواردی که نیاز به اطمینان از رسیدن موفقیت آمیزداده‌ها است از checksum استفاده می‌شود.

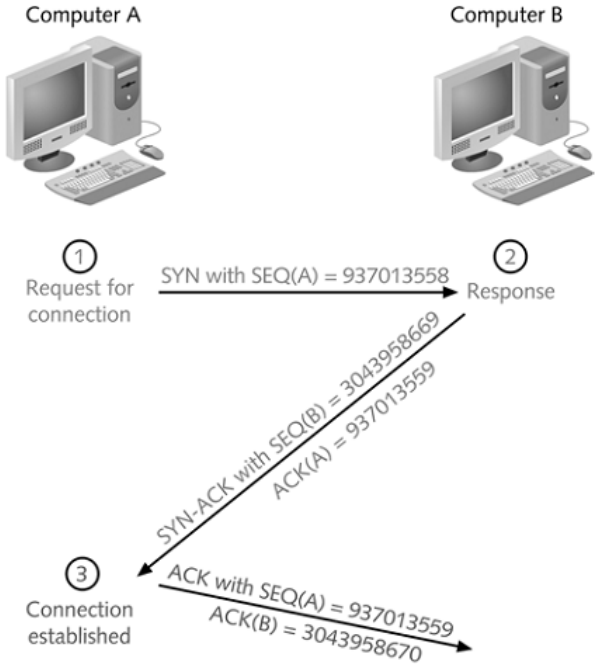


FIGURE 4-3 *Establishing a TCP connection*

تصویری از اجزای یک قطعه داده که با پروتکل TCP ارسال شده است:

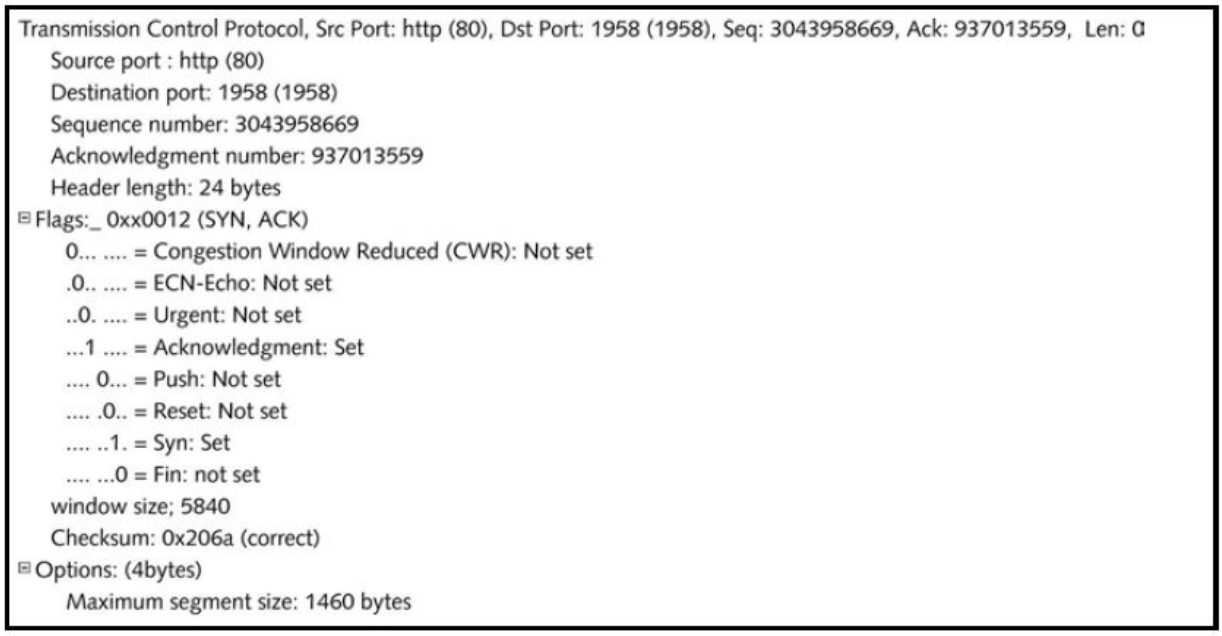


FIGURE 4-2 *TCP segment data*

UDP (User Datagram Protocol):

در پروتکل TCP رسیدن صحیح Packetها مهم است اگر یک Packet به درستی نرسد عمل ارسال دوباره و دوباره اتفاق می‌افتد. در حقیقت در این روش به محض ارسال یک Packet، یک ساعت فرضی شروع به کار می‌کند، اگر در زمان مشخصی ACK رسید همه چیز درست است اما اگر در آن زمان بسته نرسید دوباره و دوباره ارسال می‌شود. این کار آنقدر اتفاق می‌افتد تا این که بسته برسد و یا اینکه نهایتاً پیغام Destination unreachable مشاهده شود. به عبارتی پروتکل TCPرسیدن بسته را ضمانت می‌کند:

TCP Guarantees Delivery

این در مورد ارسال فایل و یا ایمیل خوب است اما تصور کنید بخواهیم صدا یا ویدئو را به صورت زنده و Real-time از طریق TCP ارسال کنیم (مثل بحث جدید Voice Over IP = VOIP که در آن شما از طریق اینترنت با افراد مختلف گفتگوی صوتی یا ویدئویی انجام می‌دهید) آیا در این نوع ارسال‌ها، زمان کافی (Time) برای این هست که هر Packet که ارسال شد منتظر جواب بمانیم و اگر اشتباه شد دوباره ارسال کنیم؟

واضح است که خیر، نمی‌توان کاربر را آنطرف خط معطل نگاه داشت تا یک پکت به مقصد برسد!

گفته می‌شود: There is no time to retransmit data

در این حالت به جای TCP از UDP استفاده می‌شود.

پس TCP برای ارسال داده و UDP برای ارسال صدا و ویدئو است. (دقت کنید که ویدئو و صدا به صورت زنده و نه یک کلیپ ویدئویی که قبلاً ضبط شده و حالا می‌خواهید بفرستیم)

با توجه به توضیحات بالا مدل UDP بسیار ساده‌تر است:

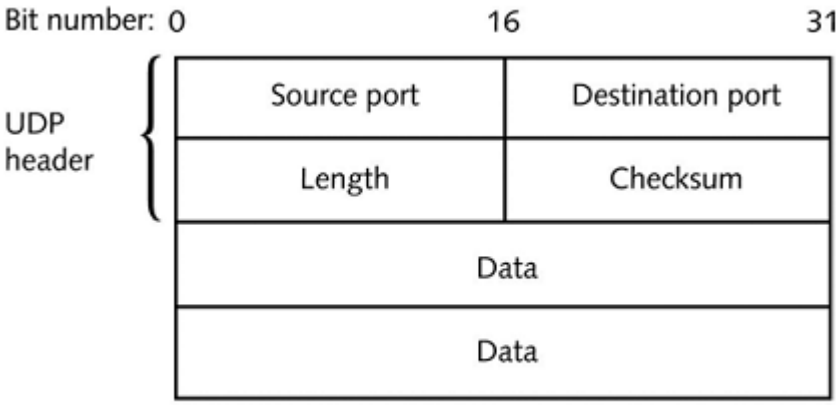


FIGURE 4-4 *A UDP Segment*

نکته ۱: checksum در پروتکل UDP اختیاری است اما برای امنیت بیشتر در ارسال، بهتر است که باشد.

نکته ۲: در اصطلاح گفته می‌شود TCP، وابسته به اتصال (Connection-Oriented) و UDP، غیروابسته به اتصال (Connection-Less) می‌باشد.

وقتی گفته می‌شود TCP یک مدل اتصال محور است یعنی یک کانکشن باید بین مبدأ و مقصد برقرار (Establish) باشد تا پروتکل شروع به انتقال داده کند.

تمرین: برای فهم بهتر TCP و UDP، متن زیر را به فارسی ترجمه کنید:

TCP is a connection-oriented subprotocol, which means that a connection must be established between communicating nodes before this protocol will transmit data. TCP further ensures reliable data delivery through sequencing and checksums. Without such measures, data would be transmitted indiscriminately, without checking whether the destination node was offline, for example, or whether the data became corrupt during transmission. Finally, TCP provides flow control to ensure that a node is not flooded with data.