

بسمه تعالی



تمرین 3  
شبکه‌های کامپیوتری

محسن کربلائی امینی، 98242128  
اردیبهشت 1403

## فصل 3:

### سوال 1:

به صورت عکس یعنی اگر ارتباط از سمت B به سمت A باشد پورت مبدا y و پورت مقصد x می باشد. برای مثال این عکس یک ارتباط TCP را نمایش می دهد:

Source	Destination	Protocol	Length	Info
172.20.121.173	172.104.134.95	TCP	66	43570 → 24489 [ACK] Seq=
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]
172.104.134.95	172.20.121.173	TCP	1514	24489 → 43570 [ACK] Seq=
172.20.121.173	172.104.134.95	TCP	66	43570 → 24489 [ACK] Seq=
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]
172.20.121.173	172.104.134.95	TCP	66	43570 → 24489 [ACK] Seq=
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]
172.20.121.173	172.104.134.95	TCP	66	43570 → 24489 [ACK] Seq=
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]
172.20.121.173	172.104.134.95	TCP	66	43570 → 24489 [ACK] Seq=
172.104.134.95	172.20.121.173	TCP	1514	24489 → 43570 [ACK] Seq=
172.104.134.95	172.20.121.173	TCP	1514	24489 → 43570 [PSH, ACK]
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]
172.20.121.173	172.104.134.95	TCP	66	43570 → 24489 [ACK] Seq=
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]
172.20.121.173	172.104.134.95	TCP	66	43570 → 24489 [ACK] Seq=
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]
172.20.121.173	172.104.134.95	TCP	102	43570 → 24489 [PSH, ACK]
172.104.134.95	172.20.121.173	TCP	2962	24489 → 43570 [PSH, ACK]

### سوال 2:

بله. با وجود اینکه UDP امکان برقراری ارتباط مطمئن را ارائه نمی کند، اما این کار را می توان در لایه اپلیکیشن با ایجاد مکانیزم های ACK, sequence numbering, retransmission, flow control و ... عملی کرد.

### سوال 3:

بله هر دو به یک socket می‌رسند. در واقع پکت‌های ارسالی به یک پورت، فارق از هر چیزی در نهایت به اپلیکیشنی می‌رسند که روی آن پورت قرار دارد. اما اینکه چگونه این اپلیکیشن می‌تواند بین سگمنت‌های دریافتی از دو ماشین، تمایز قائل شود:

```
Wireshark · Packet 14 · wlan0
├─ Frame 14: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface wlan0, id 0
├─ Ethernet II, Src: b4:6d:c2:c2:d2:60 (b4:6d:c2:c2:d2:60), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
├─ Internet Protocol Version 4, Src: 172.20.100.195, Dst: 255.255.255.255
└─ User Datagram Protocol, Src Port: 48689, Dst Port: 48689
    Source Port: 48689
    Destination Port: 48689
    Length: 59
    Checksum: 0x3689 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
    └─ [Timestamps]
        UDP payload (51 bytes)
    └─ Data (51 bytes)
```

همان‌طور که در عکس بالا مشخص شده، هر سگمنت UDP شامل پورت مبدا و پورت مقصد و همچنین از پروتکل IP می‌دانیم که آدرس مبدا و مقصد هم در پکت موجود می‌باشد، بنابراین برای دو هاست با IP متفاوت تمایز به راحتی قابل تشخیص و انجام است.

### سوال 4:

(a) در TCP هر sequence number به یک byte اطلاق می‌شود. بنابراین سگمنت اول از اندازه 20 بایت بوده است.

(b) 90. گرچه که دیتای دریافتی توسط B بافر می‌شود و از بین نمی‌رود، اما از آنجایی که TCP مکانیزم ack تجمعی دارد، باید آخرین داده‌ای که منتظر دریافت آن هست را ack کند.

## سوال 5:

در UDP محاسبه checksum از طریق پیدا کردن one's complement در 16 بیت از UDP header و UDP payload انجام می‌شود. از آنجایی که حجم دیتایی که جمع می‌شود تا این 16 بیت را درست کند، بزرگ‌تر از 16 بیت می‌باشد، عملاً تضمین نمی‌شود که checksum حتماً خطاهای احتمالی را شناسایی کند. برای مثال در جمع دو عدد، اگر از یکی یک واحد کم شود و به دیگری یک واحد اضافه شود، در نهایت جمع و در نتیجه checksum متوجه اشکال ایجاد شده نمی‌شود.

## فصل 4:

### سوال 1:

Routing به معنی مسیریابی که در لایه 3 وظیفه پیدا کردن مسیر مناسب برای ارسال بسته‌ها را بر عهده دارد اما forwarding صرفاً به معنی دریافت و ارسال پکت‌ها از یک interface به interface ای دیگر در یک تجهیز شبکه مثل سویچ در لایه 2 می‌باشد.

### سوال 2:

پرشدن بافر، ازدحام و تاخیر در تصمیم‌گیری به دلیل طولانی بودن فرایند و قوانین آدرس‌دهی می‌تواند باعث از بین رفتن بسته‌ها در ورودی روترها شود.

برای رفع این مشکلات می‌توان از فرایندهای زیر کمک گرفت:

- افزایش اندازه بافر
- زمان‌بندی بسته‌ها (مثلاً بر اساس اولویت و ...)
- مکانیزم‌های مدیریت ازدحام
- تعادل بار و توزیع بار بین روترها و interface های مختلف.
- مکانیزم‌های Quality of Service یا QoS برای تعیین لینک‌های برای ارتباطات خاص (مثلاً می‌توان دانلودها را از یک لینک و live stream ها و تماس‌های برخط را از خط با کیفیت بالاتر عبور داد)
- Traffic Shaping and Policing مدیریت جریان ورودی و میزان عبوردهی

### سوال 3:

Head of line blocking یا HOL به معنی انتظار بسته‌های عقبی صف برای پردازش برای بسته‌ای در جلوی صف که به دلیل تاخیر یا عدم توانایی پردازش مانع بسته‌های دیگر شده است. این مورد در ورودی رخ می‌دهد.

### سوال 4:

	128	64	32	16	8	4	2	1
Octet 1: 208	1	1	0	1	0	0	0	0
Octet 2: 90	0	1	0	1	1	0	1	0
Octet 3: 142	1	0	0	1	0	0	0	0
Octet 4: 211	1	1	0	1	0	0	1	1

بنابراین:

11010000.01011010.10010000.11010011

### سوال 5:

در الگوریتم‌های link-state مانند OSPF ما کم‌هزینه‌ترین مسیر را بین مبدا و مقصد پیدا می‌کنیم و بر اساس آن دانش هر نود را از شبکه گسترش می‌دهیم. مثلاً در OSPF هر نود شبکه با دریافت اطلاعات همسایگانش با استفاده از الگوریتم Shortest-path first یا SPF، یک درخت پوشا که خودش ریشه این درخت می‌باشد تشکیل می‌دهد و اطلاعات این درخت را با هر همسایه‌ای که دارد به اشتراک می‌گذارد.

در الگوریتم‌های distance-vector مانند RIP، یک نود فقط می‌داند که برای ارسال بسته به یک شبکه باید آن را به کدام شبکه بدهد و هزینه اینکار چقدر خواهد بود. برای مثال در RIP بعد از شروع اتصال‌ها، هر نود دیگری را از شبکه‌هایی که به آن متصل است خبر دارد می‌کند و اصطلاحاً hop count یا هزینه رسیدن به آن شبکه را هم اعلام می‌کند. مثلاً برای نودهایی با چنین ارتباطاتی:



برای نود A چنین *routing table* نهایی‌ای را متصور خواهیم بود:

Destination	Next Hop	Hop Count
-----		
Network B	B	1
Network C	C	1
Network D	B	2
Network E	B	2
Network F	C	2
Network G	C	2
Network H	B	3
Network I	B	3
Network J	C	3

سوال 6:

(a)

Prefix	Link interface
11100000 00	0
11100000 01000000	1
1110000	2
11100001 1	3
otherwise	3

(b)

Address	match	Link interface
11001000 10010001 01010001 01010101	otherwise	3
11100001 01000000 11000011 00111100	1110000	2
11100001 10000000 00010001 01110111	11100001 1	3

سوال 7:

Step	visited	D(z), P(z)	D(y), P(y)	D(v), P(v)	D(w), P(w)	D(u), P(u)	D(t), P(t)
0	x	8,x	6,x	3,x	6,x	INF	INF
1	xz	8,x	6,x	3,x	6,x	INF	INF
2	xzy	8,x	6,x	3,x	6,x	INF	13,y
3	xzyv	8,x	6,x	3,x	6,x	6,v	7,v
4	xzyvw	8,x	6,x	3,x	6,x	6,v	7,v
5	xzyvwu	8,x	6,x	3,x	6,x	6,v	7,v
6	xzyvwut	8,x	6,x	3,x	6,x	6,v	7,v

(a)

Step	visited	D(z), P(z)	D(y), P(y)	D(x), P(x)	D(w), P(w)	D(u), P(u)	D(t), P(t)
0	v	INF	8,y	3,v	4,v	3,v	4,v
1	vy	20,y	8,y	3,v	4,v	3,v	4,v
2	vyz	20,y	8,y	3,v	4,v	3,v	4,v
3	vyzx	11,x	8,y	3,v	4,v	3,v	4,v
4	vyzxw	11,x	8,y	3,v	4,v	3,v	4,v
5	vyzxwu	11,x	8,y	3,v	4,v	3,v	4,v
6	vyzxwut	11,x	8,y	3,v	4,v	3,v	4,v

(b)

Step	visited	D(v), P(v)	D(y), P(y)	D(x), P(x)	D(w), P(w)	D(u), P(u)	D(t), P(t)
0	z	INF	12,z	8,z	INF	INF	INF
1	zy	20,y	12,z	8,z	INF	INF	19,y
2	zyx	11,x	12,z	8,z	14,x	INF	19,y
3	zyxv	11,x	12,z	8,z	14,x	14,u	15,v
4	zyxvt	11,x	12,z	8,z	14,x	14,u	15,v
5	zyxvtu	11,x	12,z	8,z	14,x	14,u	15,v
6	zyxvtuw	11,x	12,z	8,z	14,x	14,u	15,v

سوال 8:

Destination Network	Next hop	Cost
v	x	5
x	x	2
y	x	5
u	x,v	7