

4 TCP,UDPПРОТОКОЛ

МТЭС Програм хангамж

О.Энэрэл 22B1NUM0506

4.1 Ажлын зорилго

Энэхүү лабораторийн ажлаар TCP/IP -ын тээвэрлэлтийн давхарга (*Transport layer*) -ын Transmission Control Protocol (TCP), User Datagram Protocol (UDP) протоколуудынажиллах зарчмыг судлах болно.

4.2 Туршилт

Wireshark программын тусламжтайгаар UDP болон TCP протокол, түүнтэй холбоотой мессежүүд хэрхэн дамжиж байгааг судлах болно. Wireshark программаа нээж, сүлжээний интерфейсээр дамжиж өнгөрөх урсгалуудыг барьж авна. Шүүлтүүр хэсэг TCP, UDP протоколуудыг бичсэнээр уг протоколоор дамжсан өгөгдлийн урсгалыг шүүж харах боломжтой юм.

4.3 Даалгавар

4.3.1 User Datagram Protocol

1. Өөрийн цуглуулсан өгөгдлийн урсгалаас дурын нэг UDP пакетыг сонгоно. Энэ пакетын толгой мэдээлэл (header) хэсэгт ямар талбарууд байгааг тайлбарлана уу.

No.	Time	Source	Destination	Protocol	Length	Info
14	0.716571	fe80::1	ff02::c	SSDP	188	M-SEARCH * HTTP/1.1
15	1.025749	192.168.1.1	239.255.255.250	SSDP	174	M-SEARCH * HTTP/1.1
42	1.208379	192.168.1.1	239.255.255.250	SSDP	174	M-SEARCH * HTTP/1.1
165	1.471905	fe80::1	ff05::c	SSDP	189	M-SEARCH * HTTP/1.1
170	1.727595	fe80::1	ff05::c	SSDP	189	M-SEARCH * HTTP/1.1
177	2.034467	fe80::1	ff02::c	SSDP	189	M-SEARCH * HTTP/1.1
180	2.248366	fe80::1	ff02::c	SSDP	189	M-SEARCH * HTTP/1.1
187	2.552606	192.168.1.1	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
190	2.773625	192.168.1.1	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
198	3.181081	192.168.1.19	255.255.255.255	UDP	246	59727 → 6667 Len=204

Frame 198: 246 bytes on wire (1968 bits), 246 bytes captured (1968 bits) on interface \Device\NPF_{96D4538E-98D3-4805-B282-000905E2F5C7}, id 0
 Ethernet II, Src: TuyaSmart_0a:80:90 (c4:82:e1:0a:80:90), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Internet Protocol Version 4, Src: 192.168.1.19, Dst: 255.255.255.255
 User Datagram Protocol, Src Port: 59727, Dst Port: 6667
 Source Port: 59727
 Destination Port: 6667
 Length: 212
 Checksum: 0xdca2 [unverified]
 [Checksum Status: Unverified]
 [Stream index: 3]
 [Stream Packet Number: 1]
 [Timestamps]
 UDP payload (204 bytes)
 Data (204 bytes)

Figure 1 UDP header

UDP пакетын толгой мэдээлэлийн талбарууд :

source port: Илгээж буй хостын портын дугаар.

destination port: Хүлээн авч буй хостын портын дугаар

length: UDP толгой болон өгөгдлийн урт.

checksum : Пакетийн агуулгыг шалгах зориулалттай алдааны шалгалтын утга.

2. Wireshark дээр уг пакетын packet content хэсгийг ажиглаж, UDP пакетын толгойн

мэдээллийн урт хэдэн байт болохыг тэмдэглэнэ.

UDP толгой нь 4 талбараас бүрддэг бөгөөд талбар бүр нь 2 байт тул UDP пакетын толгойн мэдээллийн урт нь 8 байт.

3. UDP пакетын толгойн хэсэгт Length талбар юуны уртыг илэрхийлж байгаа вэ? Өөрийн цуглуулсан UDP пакет дээрээ баталж, тайлбарлан тайланд оруулна.

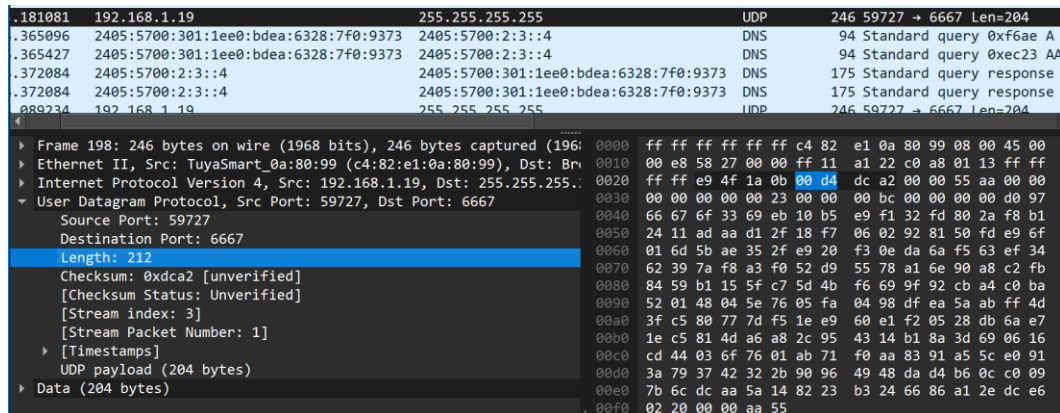


Figure 2 Length

Толгойн мэдээллийн урт Length талбар нь UDP пакетын нийт уртыг илэрхийлдэг. Энэ нь толгойн мэдээллийн урт ба өгөгдлийн хэсгийг (payload) багтаана.

Length талбарын утга 00 d4 байна, 10-т руу хөрвүүлбэл 212.

UDP толгойн мэдээлэл: 8 байт

UDP өгөгдлийн хэсэг (payload): 204 байт

4. UDP payload хэсэгт хамгийн ихдээ хэд байтын өгөгдлийг дамжуулах вэ? UDP пакетын нийт уртыг тодорхойлдог Length талбар нь 16-битийн урттай тул хамгийн ихдээ $2^{16} = 65535$ байт утгыг агуулж чадна. Гэхдээ нийт урт нь UDP толгой болон өгөгдлийн хэсгийг багтах тул $65535 - 8 = 65527$ байт өгөгдлийг дамжуулна.
5. Илгээгчийн портын дугаар хамгийн ихдээ хэд байх боломжтой вэ? UDP портын дугаар нь 16-битийн урттай байдаг. Энэ нь илгээгч болон хүлээн авагч талын портын дугааруудыг 0-65535 хооронд үүсгэх боломжтой гэсэн үг. Тиймээс илгээгчийн портын дугаарын хамгийн их утга нь 65535 байж болно.
6. Сүлжээний давхаргын толгой мэдээлэлд агуулагдах UDP протоколын дугаар (type) хэд байна вэ? Сүлжээний давхаргын толгой мэдээллийг ажиглаж хариулна уу.

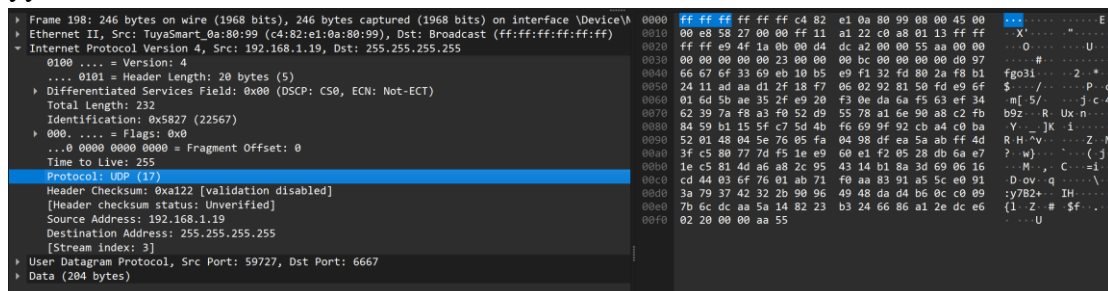


Figure 3 UPD type

UDP протоколын дугаар 17 байна.

7. Өөрийн PC-ээс илгээгдэж байгаа хэд хэдэн UDP пакетууд дээр дараахажиглалтыг

хийгээрэй. Илгээгч эхний UDP пакетыг илгээх бөгөөд хоёр дахь UDP пакет нь эхний UDP пакетын хариу гэвэл (Энд, эхний пакетын хариуд хоёр дахь пакет илгээгдсэн бол эхний пакет илгээгч нь хоёр дахь пакет нь хүлээн авагч байх ёстой), уг хоёр пакетын порт дугааруудын хоорондын хамаарал ямар байх вэ, тайлбарлаарай.

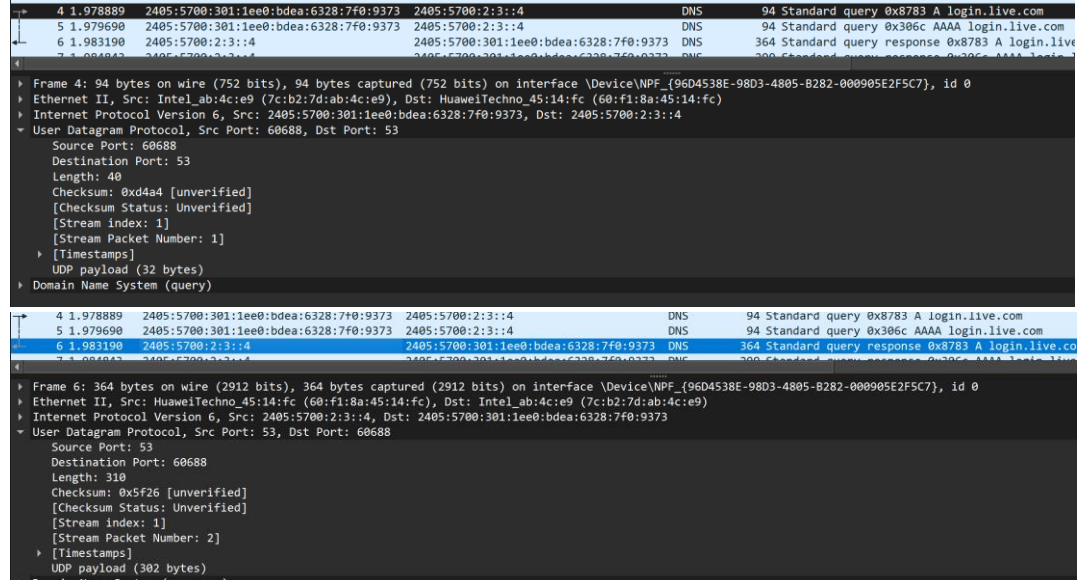


Figure 4 ip addr and port number

Эхний пакет дээр илгээгчийн Source Port хаяг хариу пакетын Destination Port болж, хүлээн авагчийн Destination Port хаяг хариу пакетын Source Port болж солигдсон.

4.3.2 Transmission Control Protocol

Түшилт 1

TCP холболтын 3-way handshake процесс болон HTTP протоколын POST мессежийн талаар судална. Wireshark программыг эхлүүлж, сүлжээний интерфэйсээр дамжиж өнгөрөх урсгалуудыг барьж авна. Веб хөтчөөс дараах хаяг руу хандаж <http://netconf.num.edu.mn/wiresharkfiles/alice.txt> файлыг татаж авна. <http://netconf.num.edu.mn/wiresharkfiles/TCP-wireshark-file1.html> хуудас руу хандаж Браузерийг ашиглан дээр татсан файлыг сонгож, “Upload alice.txt file” дарж gaia.cs.umass.edu сервер рүү илгээнэ. Wireshark программыг зогсоож, дараах асуултад хариулна уу.

1. Илгээгч, хүлээн авагч төхөөрөмжүүдийн IP хаяг, TCP портын дугаар ямар байна вэ? Аль пакетаас уг мэдээллийг авч байгааг тайланд тусгаарай.

```

705 17.311018 192.168.1.94 64.119.31.104 HTTP 573 GET /wiresharkfiles/TCP-wireshark-file1.html HTTP/1.1
706 17.316994 64.119.31.104 192.168.1.94 HTTP 1254 HTTP/1.1 200 OK (text/html)

Frame 705: 573 bytes on wire (4584 bits), 573 bytes captured (4584 bits) on interface \Device\NPF...
Ethernet II, Src: Intel_ab:4c:e9 (7c:b2:7d:ab:4c:e9), Dst: HuaweiTechno_45:14:fc (60:f1:8a:45:14:fc)
Internet Protocol Version 4, Src: 192.168.1.94, Dst: 64.119.31.104
Transmission Control Protocol, Src Port: 57054, Dst Port: 80, Seq: 982, Ack: 55259, Len: 519
  Source Port: 57054
  Destination Port: 80
  [Stream index: 91]
  [Stream Packet Number: 33]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 519]
  Sequence Number: 982 (relative sequence number)
  Sequence Number (raw): 1454643914
  [Next Sequence Number: 1501 (relative sequence number)]
  Acknowledgment Number: 55259 (relative ack number)
  Acknowledgment number (raw): 3716166198
  0101 .... = Header Length: 20 bytes (5)
  Flags: 0x018 (PSH, ACK)
  [Calculated window size: 130816]
  [Window size scaling factor: 256]

```

Figure 5 ip address, port number

Илгээгчийн IP хаяг: 192.168.1.94, портын дугаар 57054

Хүлээн авагчийн IP хаяг: 64.119.31.104, портын дугаар 80

HTTP GET пакетаас уг мэдээллийг авсан.

- Илгээгч компьютер болон серверийн хооронд TCP холболт үүсгэж эхлэхэд ашиглагдаж байгаа TCP SYN сегментийн дарааллын дугаар (sequence number) ямар байна вэ? Уг портын дугаарыг дараа дараагийн дамжуулалд хэрхэн ашиглаж байгааг ажиглан тайланд тусгана уу.

```

627 11.408828 192.168.1.94 64.119.31.104 TCP 66 57057 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
Frame 627: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{96D4538E-9803-4805-B282-000905E2F5C7}, id 0000
Ethernet II, Src: Intel_ab:4c:e9 (7c:b2:7d:ab:4c:e9), Dst: HuaweiTechno_45:14:fc (60:f1:8a:45:14:fc)
Internet Protocol Version 4, Src: 192.168.1.94, Dst: 64.119.31.104
Transmission Control Protocol, Src Port: 57057, Dst Port: 80, Seq: 0, Len: 0
  Source Port: 57057
  Destination Port: 80
  [Stream index: 86]
  [Stream Packet Number: 1]
  [Conversation completeness: Incomplete (37)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 1265384465
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1000 .... = Header Length: 32 bytes (8)
  Flags: 0x002 (SYN)
  Window: 64240
  [Calculated window size: 64240]
  Checksum: 0x220c [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted

```

Figure 6 TCP SYN

TCP SYN сегментийн дарааллын дугаар нь 0.

- Серверээс клиент рүү илгээгдсэн SYN ACK сегментийн дарааллын дугаар ямар байна? SYN ACK сегментийн acknowledgement талбар ямар утгатай байна вэ? Сегментийн аль утга дээр үндэслэн SYN ACK сегмент таньж байгаа вэ?

```

638 12.484124 64.119.31.104 192.168.1.94 TCP 66 80 → 57054 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1412 SACK_PERM=1 WS=1
Frame 638: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{96D4538E-9803-4805-B282-000905E2F5C7}, id 0000
Ethernet II, Src: HuaweiTechno_45:14:fc (60:f1:8a:45:14:fc), Dst: Intel_ab:4c:e9 (7c:b2:7d:ab:4c:e9)
Internet Protocol Version 4, Src: 64.119.31.104, Dst: 192.168.1.94
Transmission Control Protocol, Src Port: 80, Dst Port: 57054, Seq: 0, Ack: 1, Len: 0
  Source Port: 80
  Destination Port: 57054
  [Stream index: 91]
  [Stream Packet Number: 2]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 3716110930
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 1454642933
  1000 .... = Header Length: 32 bytes (8)
  Flags: 0x012 (SYN, ACK)
  Window: 64240
  [Calculated window size: 64240]
  Checksum: 0xcda7 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
  [Timestamps]
  [SEQ/ACK analysis]

```

Figure 7 SYN ACK

SYN ACK сегментийн дарааллын дугаар нь 0, acknowledgement талбар нь 1 утгатай байж Flags талбараас SYN ACK сегмент таньж байна.

- HTTP POST мессэжийг агуулж байгаа TCP сегментийн дарааллын дугаар ямар байна вэ?

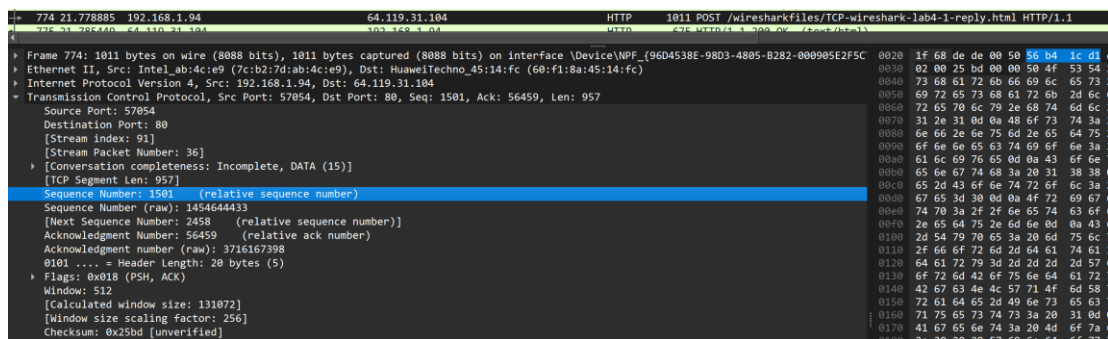


Figure 8 http post

TCP сегментийн дарааллын дугаар нь 1501.

5. TCP холболтын эхний сегментээр HTTP POST агуулж байгаа TCP сегментийг гэж үзвэл эхний 6 сегментүүдийн дарааллын дугаар ямар байна вэ? Дээрх 6 сегмент тус бүрийн илгээсэн болон acknowledgement хүлээн авсан хоорондын хугацаа болох RTT (Round Trip Time) нь сегмент тус бүр ямар байна вэ? АСК хүлээж авсны дараах EstimatedRTT утга ямар байх вэ?

108	4.434545	192.168.1.94	64.119.31.104	TCP	66	60110	→ 80	[SYN, ACK] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM WS=1
111	4.440678	64.119.31.104	192.168.1.94	TCP	66	60110	→ 80	[ACK] Seq=0 Win=64240 Len=0 MSS=1412 SACK_PERM WS=1
112	4.440758	192.168.1.94	64.119.31.104	TCP	54	60110	→ 80	[ACK] Seq=1 Ack=1 Win=131072 Len=0
115	4.463317	192.168.1.94	64.119.31.104	TCP	66	60111	→ 80	[SYN, ACK] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM WS=1
116	4.469414	64.119.31.104	192.168.1.94	TCP	66	80	→ 60111	[SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1412 SACK_PERM WS=1
117	4.469504	192.168.1.94	64.119.31.104	TCP	54	60111	→ 80	[ACK] Seq=1 Ack=1 Win=131072 Len=0
122	7.728234	2405:5700:301:1ee0:bdea:6328:7f0:9373	2620:1ec:21::14	TCP	74	60109	→ 443	[FIN, ACK] Seq=2142 Ack=575 Win=511 Len=0
123	7.818702	2620:1ec:21::14	2405:5700:301:1ee0:bdea:6328:7f0:9373	TCP	74	443	→ 60109	[ACK] Seq=575 Ack=2143 Win=16387 Len=0
124	7.818702	2620:1ec:21::14	2405:5700:301:1ee0:bdea:6328:7f0:9373	TCP	74	443	→ 60109	[FIN, ACK] Seq=575 Ack=2143 Win=16387 Len=0
125	7.818702	2405:5700:301:1ee0:bdea:6328:7f0:9373	2620:1ec:21::14	TCP	74	60109	→ 443	[ACK] Seq=2143 Ack=576 Win=511 Len=0
135	12.261387	192.168.1.94	64.119.31.104	HTTP	1011	POST	/wiresarkfiles/TCP-wireshark-lab4-1-reply.html	HTTP/1.1
136	12.273872	64.119.31.104	192.168.1.94	HTTP	676	HTTP/1.1	200 OK	(text/html)
137	12.276952	64.119.31.104	192.168.1.94	TCP	54	80	→ 60110	[ACK] Seq=1 Ack=958 Win=64128 Len=0
138	12.276994	192.168.1.94	64.119.31.104	TCP	54	60110	→ 80	[ACK] Seq=958 Ack=623 Win=130560 Len=0
140	14.166351	192.168.1.94	64.119.31.104	TCP	54	60111	→ 80	[FIN, ACK] Seq=1 Ack=1 Win=131072 Len=0
141	14.166522	192.168.1.94	64.119.31.104	TCP	54	60110	→ 80	[FIN, ACK] Seq=958 Ack=623 Win=130560 Len=0
142	14.177106	64.119.31.104	192.168.1.94	TCP	54	80	→ 60111	[FIN, ACK] Seq=1 Ack=2 Win=64256 Len=0
143	14.177106	64.119.31.104	192.168.1.94	TCP	54	80	→ 60110	[FIN, ACK] Seq=623 Ack=959 Win=64128 Len=0
144	14.177248	192.168.1.94	64.119.31.104	TCP	54	60111	→ 80	[ACK] Seq=2 Ack=2 Win=131072 Len=0
145	14.177248	192.168.1.94	64.119.31.104	TCP	54	60110	→ 80	[ACK] Seq=959 Ack=624 Win=130560 Len=0

Figure 9 RTT

Packet	Segment	Sequence Number	Илгээсэн цаг(s)	Хүлээн авсан(s)	RTT(ms)	EstimatedRTT(ms)
108	SYN	0	4.434545	4.440678	133	104.13
111	SYN-ACK	0	4.440678	4.440758	80	101.64
112	ACK	1	4.440758	12.261387	7820	1066.43
135	HTTP POST	1	12.261387	12.276994	15.607	935.10
138	ACK	958	12.276994	14.166522	188.952	1053.39
141	FIN, ACK	958	14.166522			

Туришлт 2

TCP холболтын бөгнөрөлөөс зайлсхийх арга болох TCP Reno -ийн талаар судална. Доорх зургийг ажиглан асуултад хариулаарай.

1. *TCP Slow start* эхэлж байгаа хугацааны интервалыг тодорхойл.

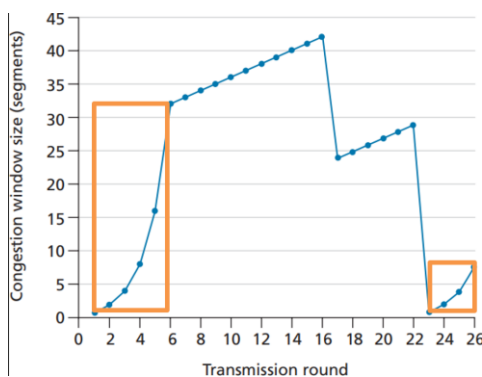


Figure 10 Slow start

Slow start нь [1,6], [23, 26] хугацааны интервалд үүсэж байна.

2. TCP congestion avoidance эхэлж байгаа хугацааны интервалыг тодорхойл.

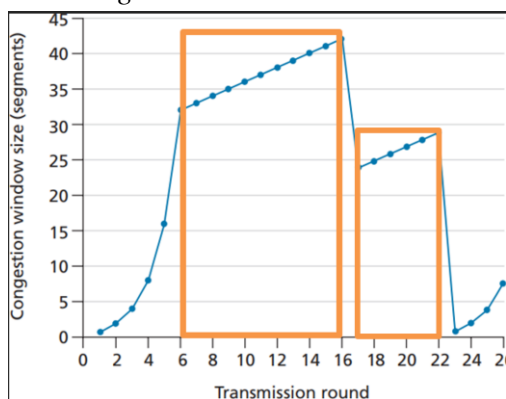


Figure 11 congestion avoidance

Congestion avoidance [6,16], [17, 22] хүртэл үргэлжилж байна.

3. 16-р дамжууллын дараа, сегментийг *triple duplicate ACK* эсвэл *timeout* -ийн аль нь болсон эсэхийг тодорхойл.

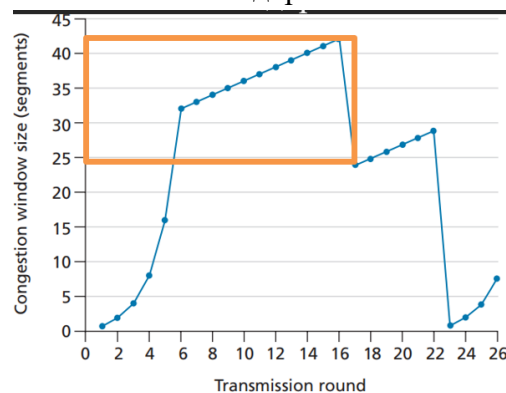


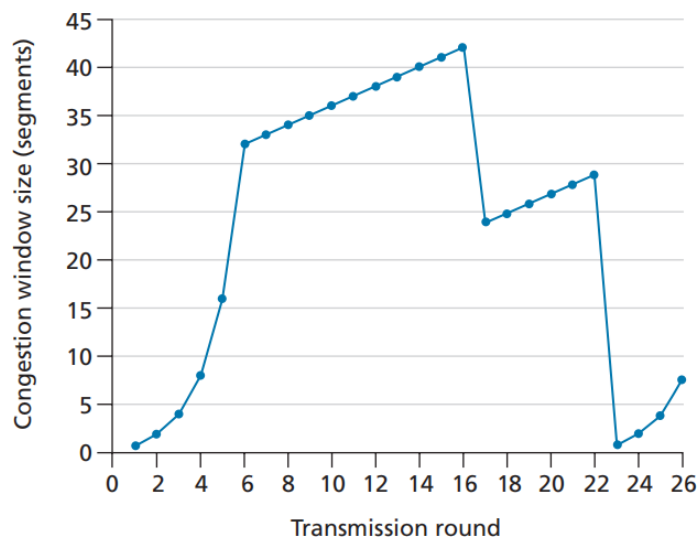
Figure 12 triple duplicate

16-р дамжууллын дараа графикт огцом уналт гарч, congestion window нь ойролцоогоор 42 сегментээс 24 орчим сегмент хүртэл буураад тэндээс дахин аажмаар нэмэгдэж байгаа нь *triple duplicate ACK* болж, TCP нь fast recovery механизмаар ажиллаж эхэлснийг илэрхийлж байна. Timeout болсон бол уналт илүү их байх байсан.

4. 22-р дамжууллын дараа, сегментийн алдагдлыг *triple duplicate ACK* илрүүлсэн эсвэл *timeout* болсон эсэхийг тодорхойл.

22-р дамжууллын дараа 0 сегмент хүртэл буурсан тус timeout болсон.

5. Эхний дамжууллын үед *ssthresh* утга хэд байх вэ?
Эхний дамжууллын үед *ssthresh* утга 32 байна.
6. 18 дахь дамжууллын үед *ssthresh* утга хэд байх вэ?
16 дахь дамжууллын үеэр алдагдал илэрсэн үед congestion window-ийн хэмжээ 42 байсан тул 18 дахь дамжууллын үед *ssthresh* утга $42/2 = 21$ байна.
7. 24 дахь дамжууллын *ssthresh* утга хэд болсон байна вэ?
22 дахь transmission үеэр алдагдал илэрсэн үед congestion window-ийн хэмжээ 26 байсан иймд 24 дахь дамжууллын *ssthresh* утга $26/2 = 13$ байна.
8. 70 дахь сегментийг дамжуулж байна гэж үзвэл, хэд дэх дамжууллын үед илгээгдэх вэ?
70 дахь сегментийг 30-40 дамжуулалтын хооронд илгээгдэх магадлалтай.
9. 26 дахь дамжууллын дараа *triple duplicate ACK* хүлээн авч пакетын алдагдлыг илрүүлсэн бол, *congestion window size* утга болон *ssthresh*-ийн утга хэд болох вэ?
Triple duplicate ACK хүлээн авах үед TCP нь congestion window-ийг бууруулдаг. Хэрэв 26 дахь дамжууллын үед congestion window-ийн утга 8 сегмент байна, шинэ congestion window ойроцоогоор 4 сегмент болж *ssthresh* нь 4 болно.
10. Дээрх тохиолдолд TCP *tahoe* аргыг ашиглаж байна гэж үзвэл, 16 дахь дамжууллын үед *triple duplicate ACK* хүлээн авсан гээ. Тэгвэл 19 дахь дамжууллын үед *ssthresh* болон *congestion window size* хэд байх вэ?
16 дахь дамжууллын үед congestion window нь 42 сегмент байсан, Tahoe нь *triple duplicate ACK* хүлээн авснаар *ssthresh*-ийн утга нь *cwnd*-ийн хагас болж шинэчилнэ, иймд 21 сегмент болно. Үүний дараа, 17, 18, болон 19 дэх дамжууллаудад *slow start* эхэлж, congestion window дахин өсөж эхэлнэ.
17 дахь дамжуулалтын дараа *cwnd* = 2 сегмент
18 дахь дамжуулалтын дараа *cwnd* = 4 сегмент
19 дахь дамжуулалтын дараа *cwnd* = 8 сегмент



Зураг 4.2 TCP Reno алгоритм

Дүгнэлт

Тээвэрлэлтийн давхаргын протоколууд нь хостуудын хооронд логик холболт үүсгэх, өгөгдлийг сегментээр дамжуулахад чухал үүрэгтэй. TCP (Transmission

Control Protocol) нь холболтод суурилсан протокол бөгөөд өгөгдлийг найдвартай, дараалалтайгаар дамжуулахад зориулагдсан. TCP-ийн three-way handshaking механизмаар логик холболтыг үүсгэж, сегментүүдийн дараалал, алдаагийн хяналтыг хангадаг. UDP (User Datagram Protocol) нь холболтод суурилагүй протокол бөгөөд энгийн дамжуулалт хийхэд ашиглагддаг. UDP нь найдвартай байдал, дарааллыг хангадаггүй учраас хурдтай дамжуулалт шаарддаг стрийминг үйлчилгээ, дуу, видео зэрэгт хэрэглэгддэг. UDP протокол нь сегмент бүрийг хүлээн авагчид хүргэх эсэхийг нягтладаггүй бөгөөд энэ нь алдагдсан пакетуудыг анхаардаггүй.