Homework 5

Congestion Control and TCP Variants

Homework Overview

- Getting familiar with TCP congestion control, including the related concepts and algorithms.
- Thinking about the relationships and differences between TCP variants.
- Doing some hands-on work with Wireshark and learning how to use Wireshark to analyze network traffic.

Question 1: TCP Congestion Control Window

Assuming TCP Reno is the protocol resulting in the behavior shown hereafter in the Figure below. Answer the following questions. In all cases, try to provide a short explanation for each of your answer in 2-3 sentences. Remember that Threshold is the limit after which TCP switches from slow start to congestion avoidance.

For simplicity, we assume that whenever packets are sent or received, the whole congestion window is sent or received. We then call a transmission round the time period between the emission of a congestion window worth of packets and the reception of the corresponding acks.

Question 1: TCP Congestion Control Window

Assuming TCP Reno is the protocol resulting in the behavior shown hereafter in the Figure below. Answer the following questions. In all cases, try to provide a short explanation for each of your answer in 2-3 sentences. Remember that Threshold is the limit after which TCP switches from slow start to congestion avoidance.

For simplicity, we assume that whenever packets are sent or received, the whole congestion window is sent or received. We then call a transmission round the time period between the emission of a congestion window worth of packets and the reception of the corresponding acks.

Question 1: TCP Congestion Control Window

Assuming TCP Reno is the protocol resulting in the behavior shown hereafter in the Figure below. Answer the following questions. In all cases, try to provide a short explanation for each of your answer in 2-3 sentences. Remember that Threshold is the limit after which TCP switches from slow start to congestion avoidance.

For simplicity, we assume that whenever packets are sent or received, the whole congestion window is sent or received. We then call a transmission round the time period between the emission of a congestion window worth of packets and the reception of the corresponding acks.

Question 1: TCP Congestion Control Window

Assuming TCP Reno is the protocol resulting in the behavior shown hereafter in the Figure below. Answer the following questions. In all cases, try to provide a short explanation for each of your answer in 2-3 sentences. Remember that Threshold is the limit after which TCP switches from slow start to congestion avoidance.

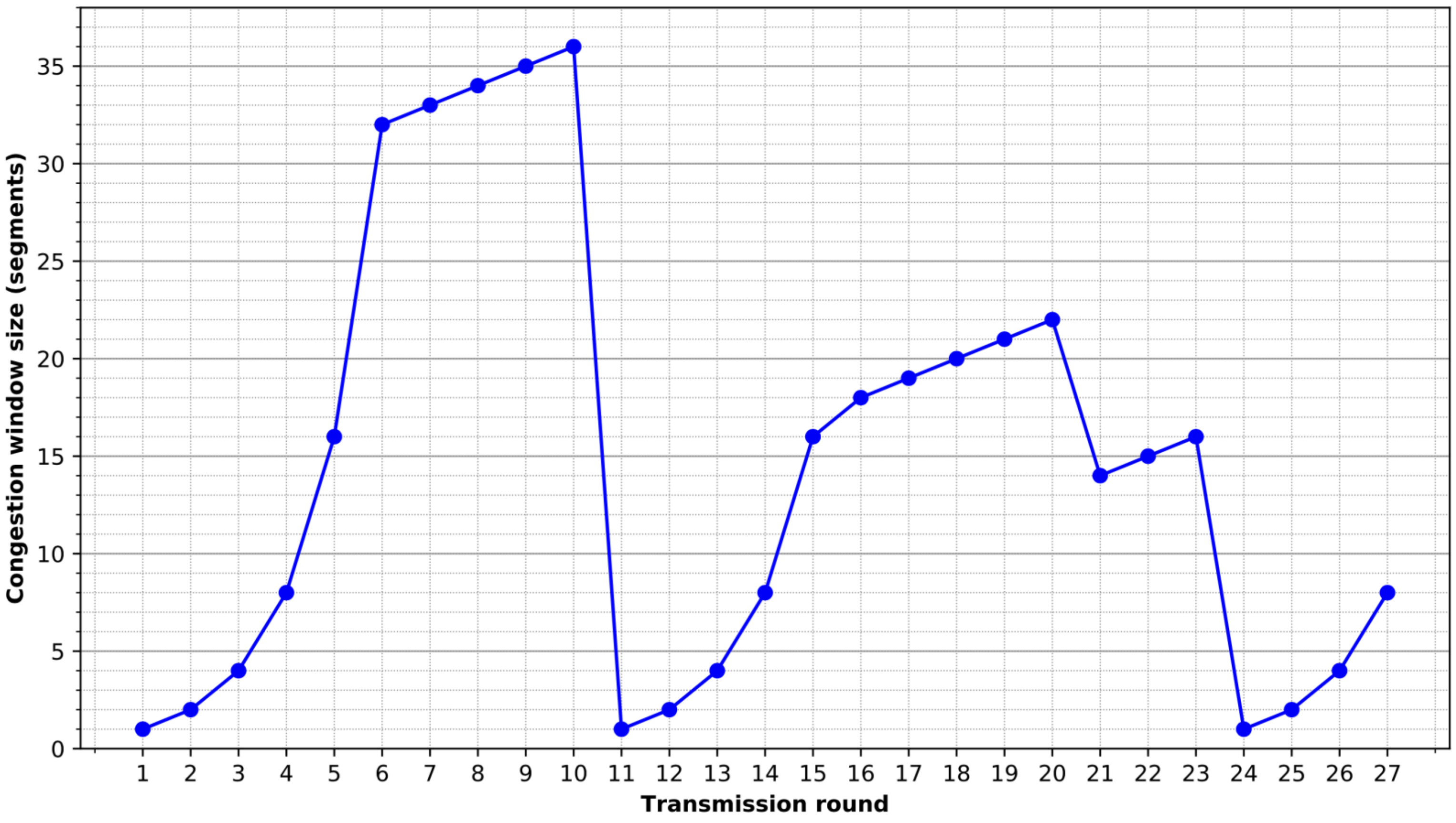
For simplicity, we assume that whenever packets are sent or received, the whole congestion window is sent or received. We then call a transmission round the time period between the emission of a congestion window worth of packets and the reception of the corresponding acks.

Question 1: TCP Congestion Control Window

Assuming TCP Reno is the protocol resulting in the behavior shown hereafter in the Figure below. Answer the following questions. In all cases, try to provide a short explanation for each of your answer in 2-3 sentences. Remember that Threshold is the limit after which TCP switches from slow start to congestion avoidance.

For simplicity, we assume that whenever packets are sent or received, the whole congestion window is sent or received. We then call a transmission round the time period between the emission of a congestion window worth of packets and the reception of the corresponding acks.

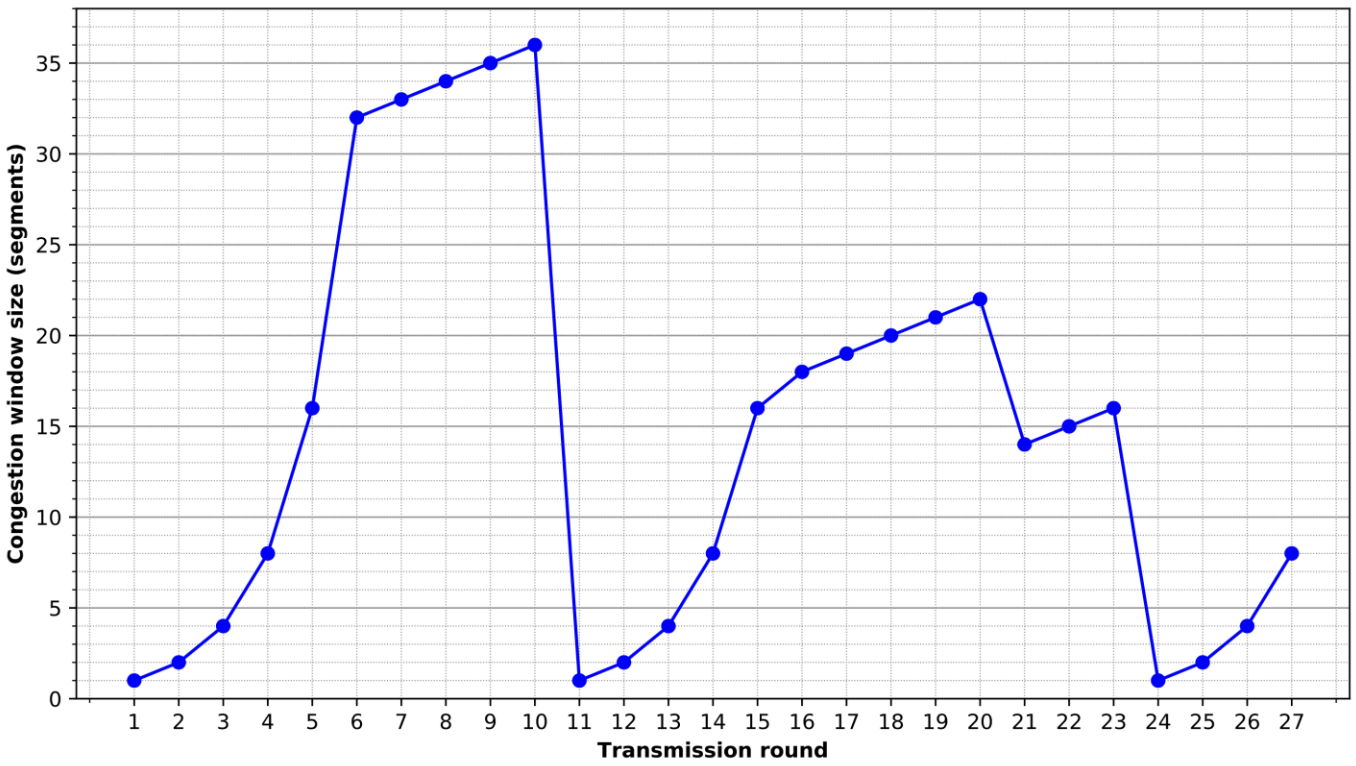
Question 1: TCP Congestion Control Window



As an example, in the 3rd transmission round, 4 packets are sent and 4 ACKs are received.

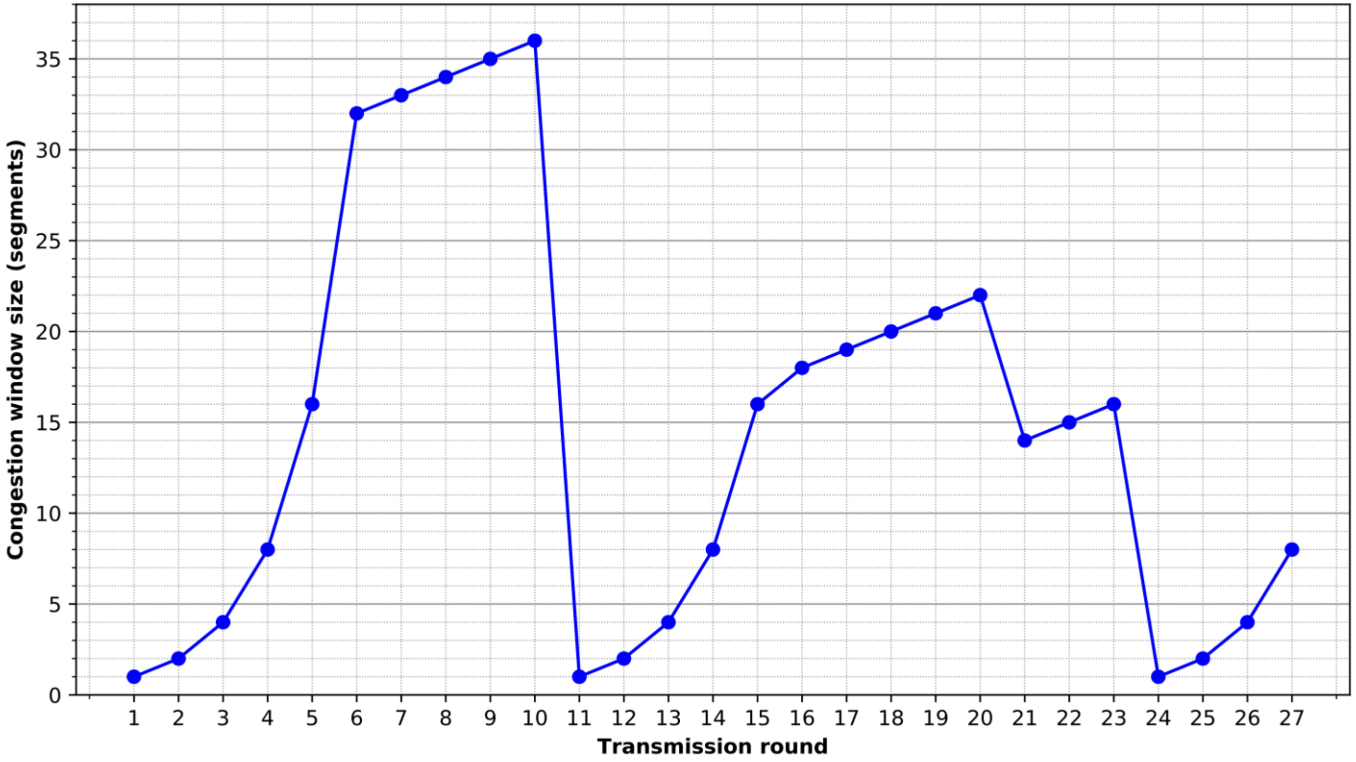
Question 1 (a)

Identify the time intervals when TCP slow start is operating.



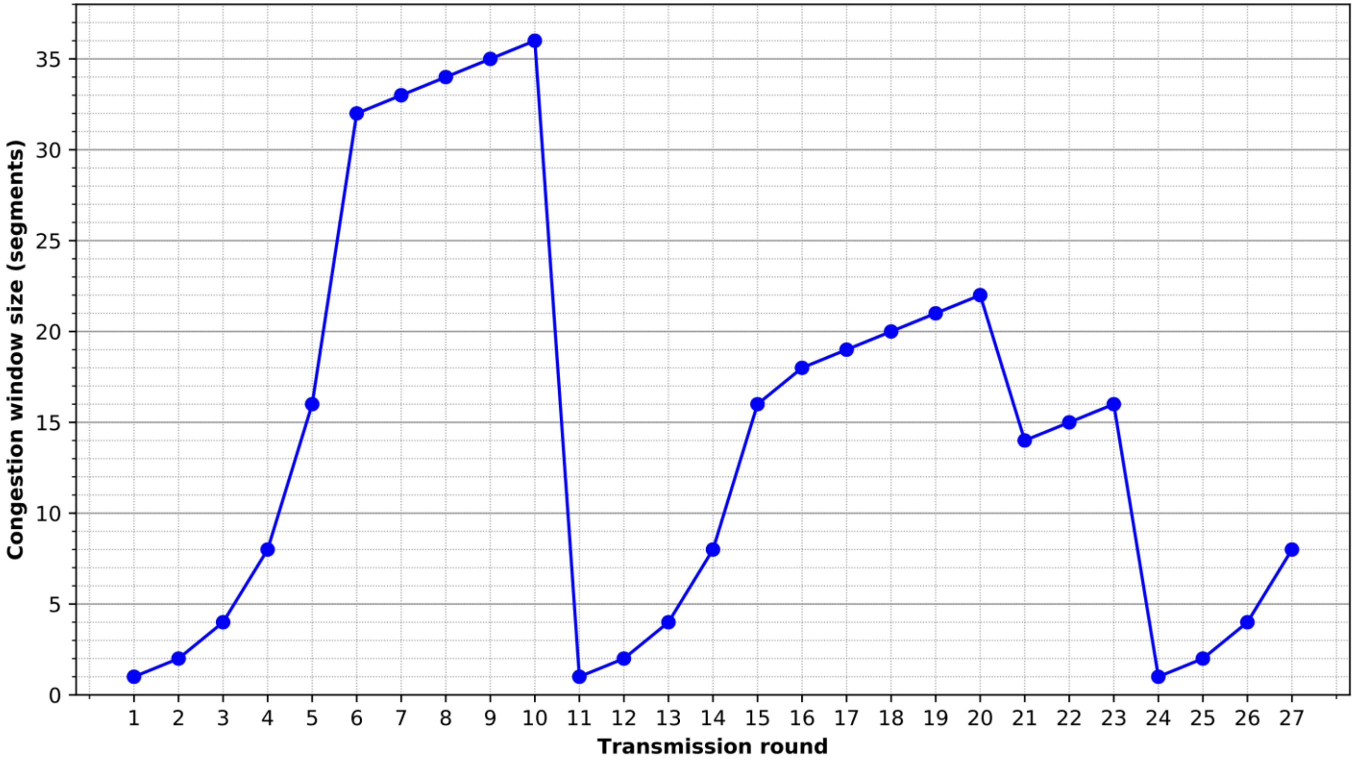
Question 1 (a)

Identify the time intervals when TCP slow start is operating.



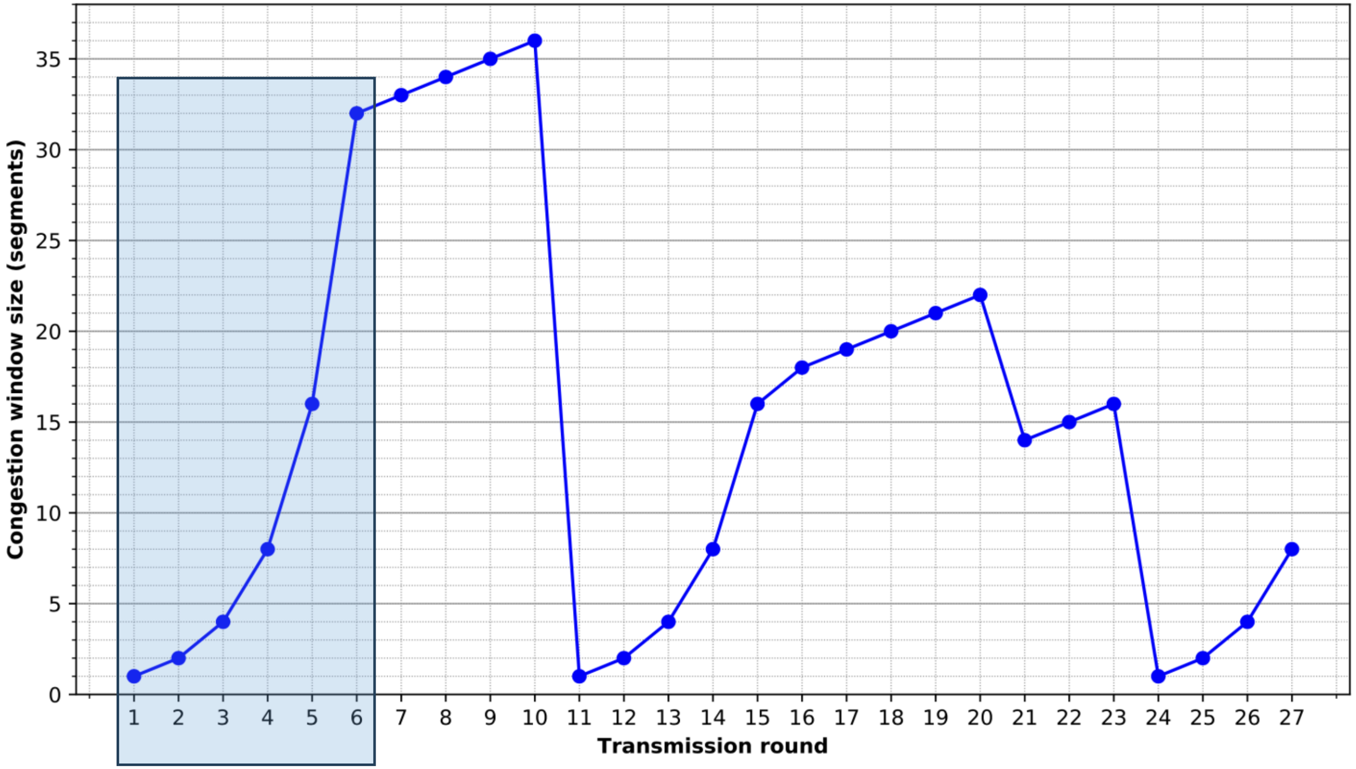
Question 1 (a)

Identify the time intervals when TCP slow start is operating.



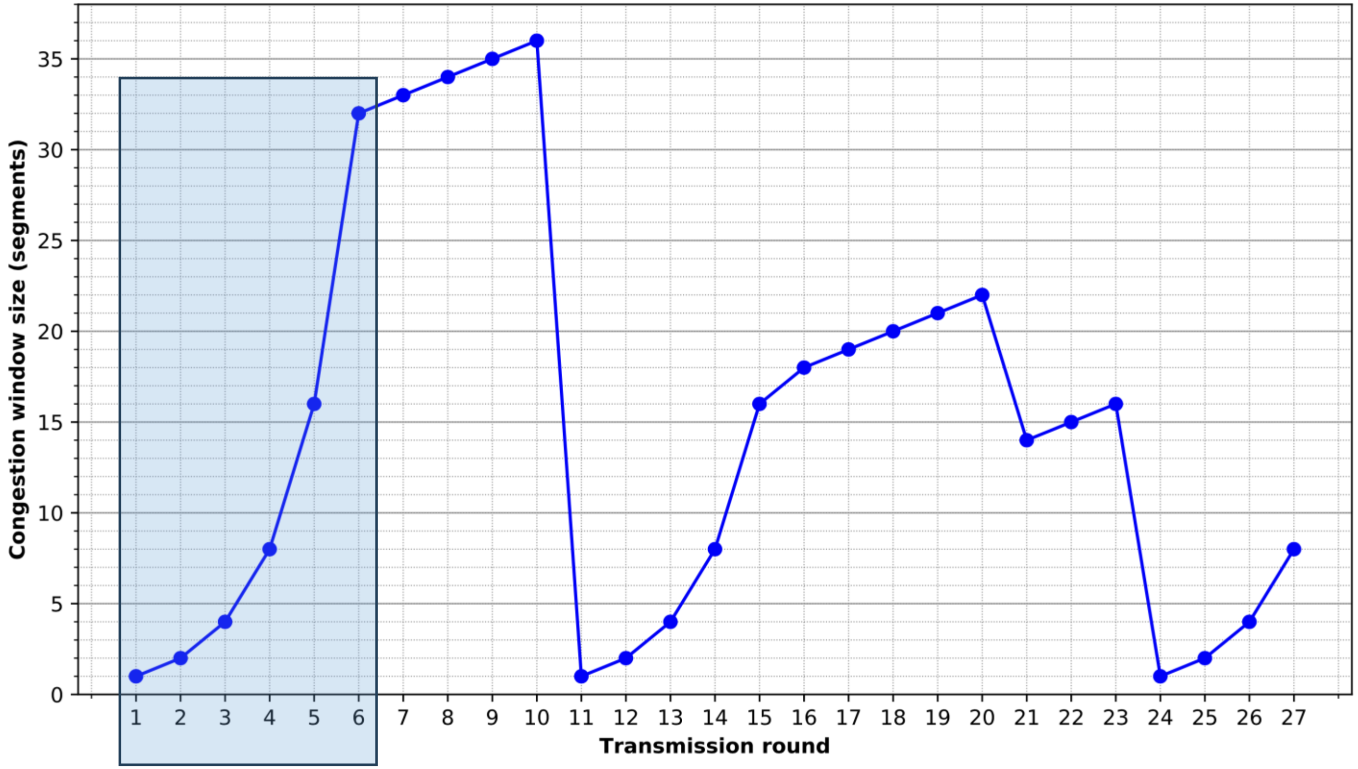
Question 1 (a)

Identify the time intervals when TCP slow start is operating.



Question 1 (a)

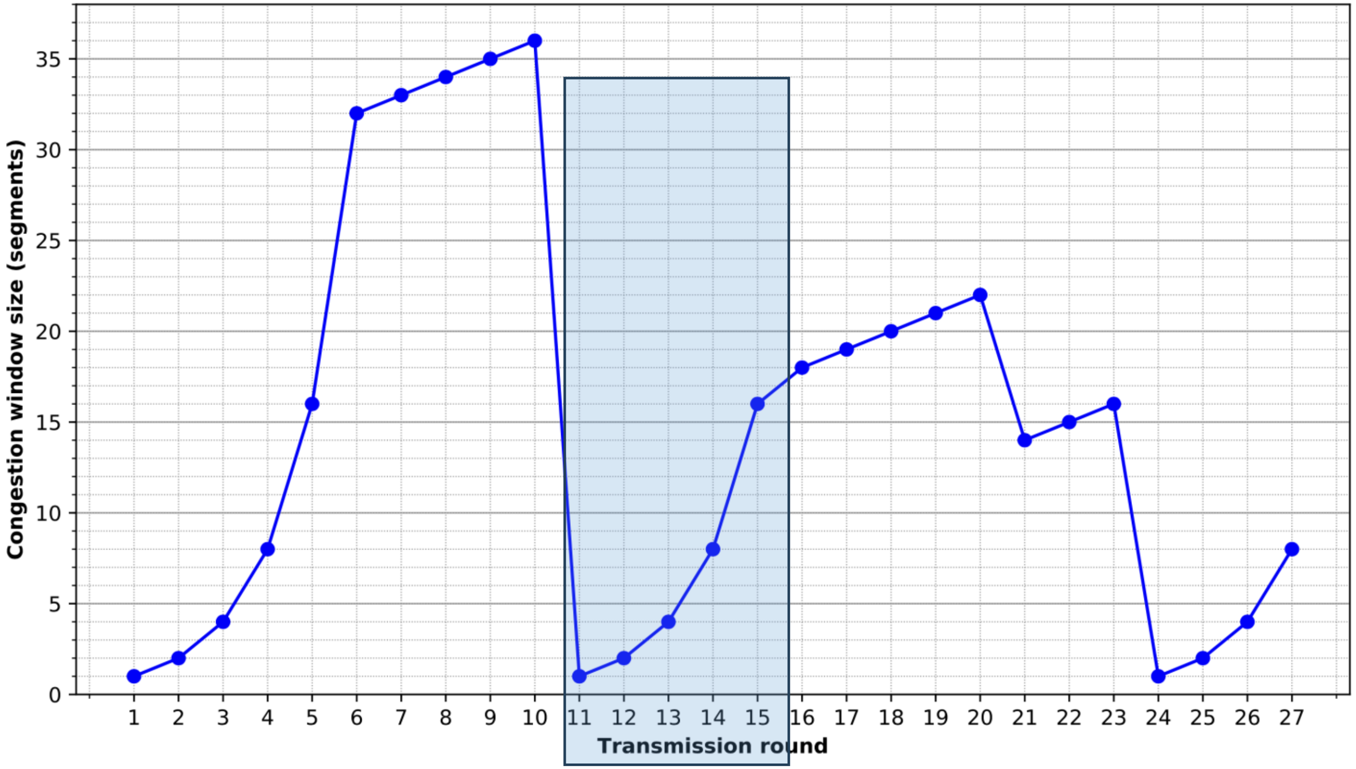
Identify the time intervals when TCP slow start is operating.



[1,6]

Question 1 (a)

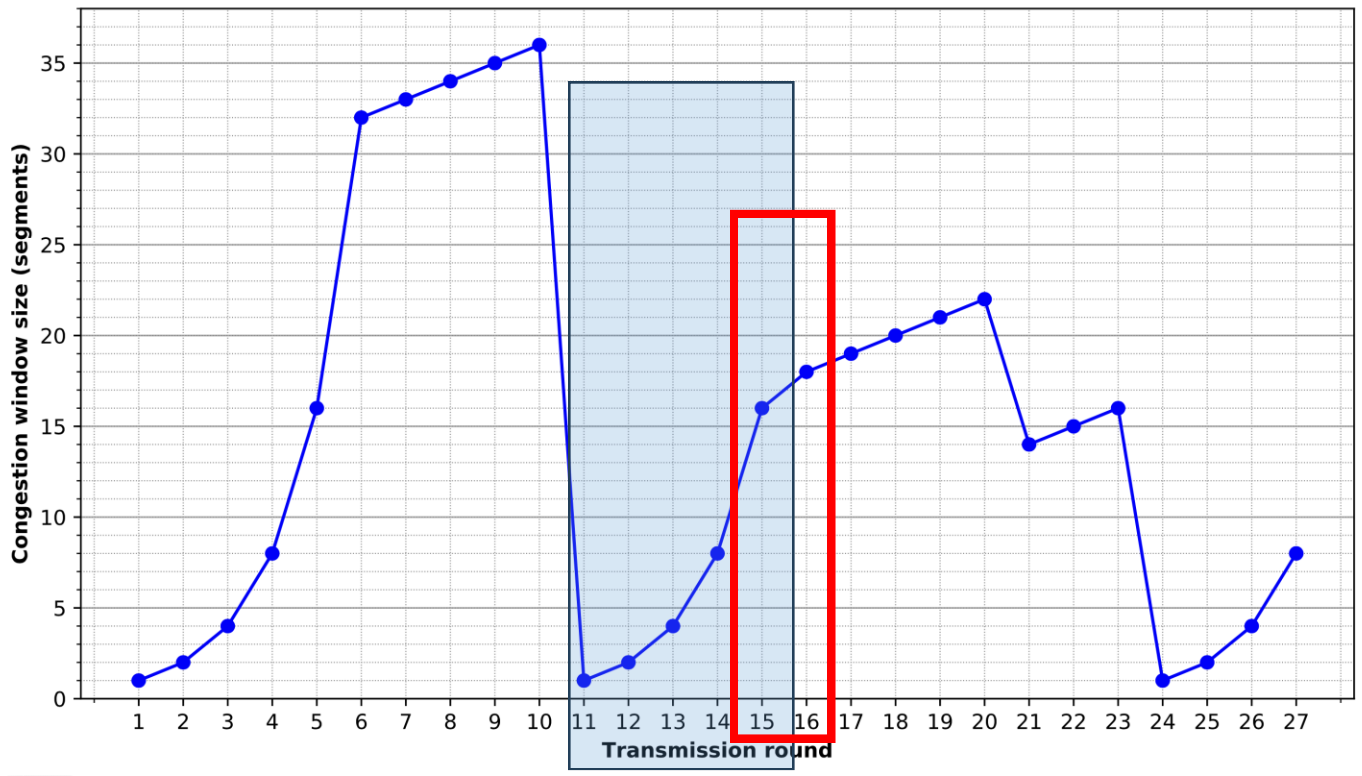
Identify the time intervals when TCP slow start is operating.



[1,6]

Question 1 (a)

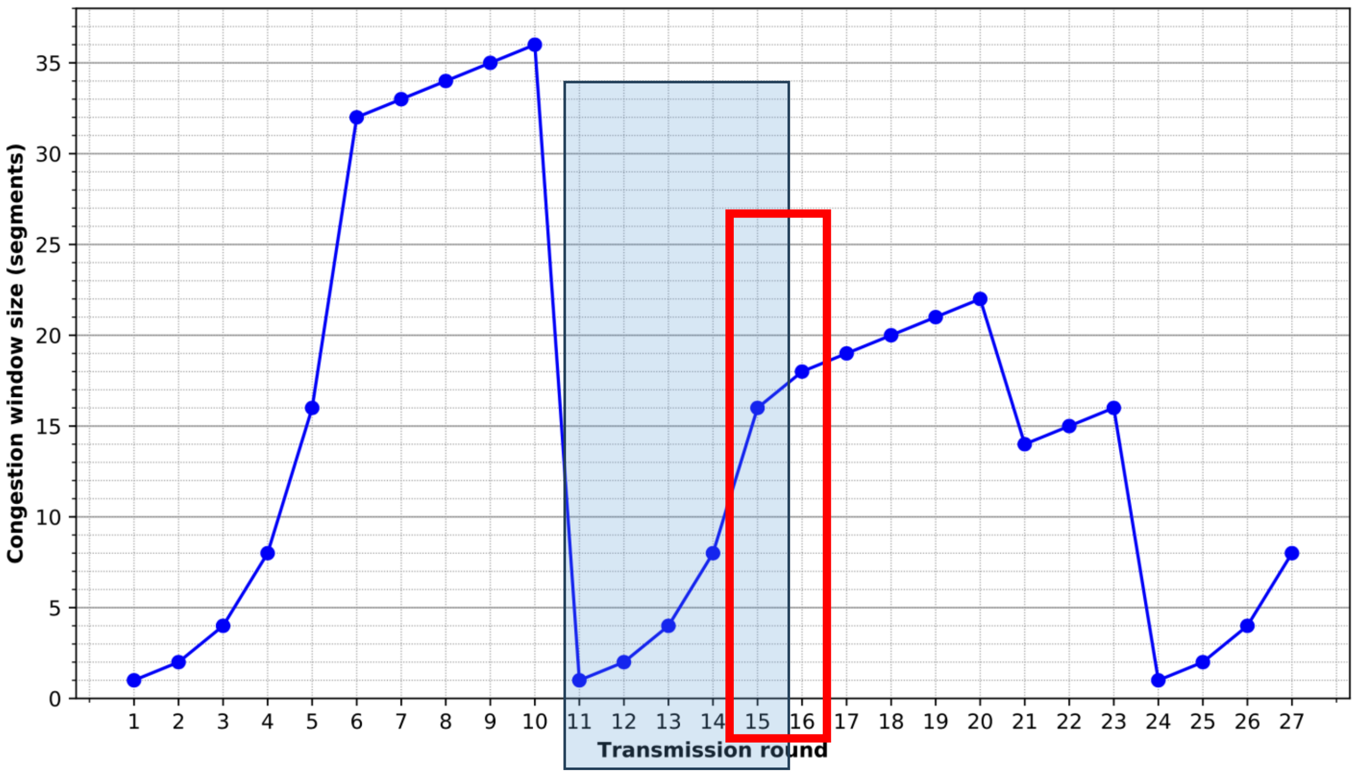
Identify the time intervals when TCP slow start is operating.



[1,6]

Question 1 (a)

Identify the time intervals when TCP slow start is operating.

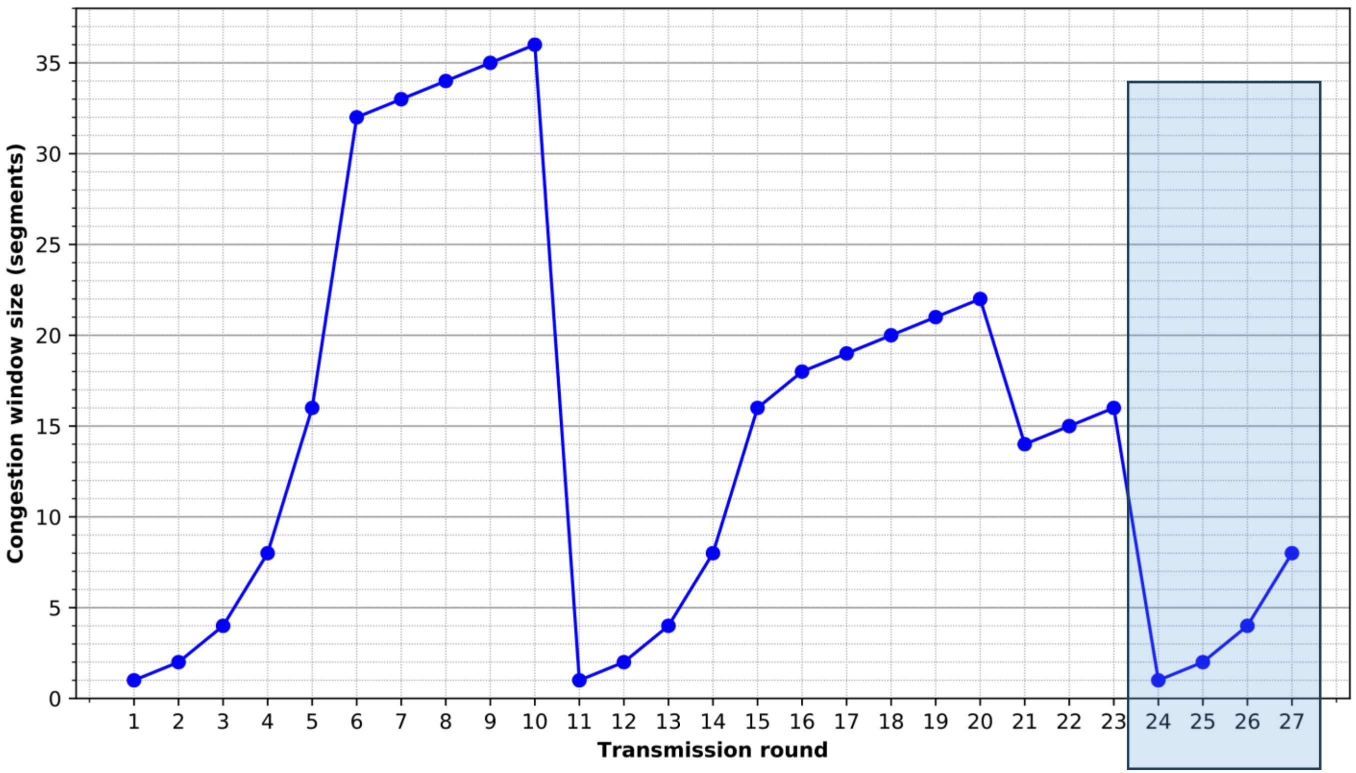


[1,6]

[11,15/16]

Question 1 (a)

Identify the time intervals when TCP slow start is operating.

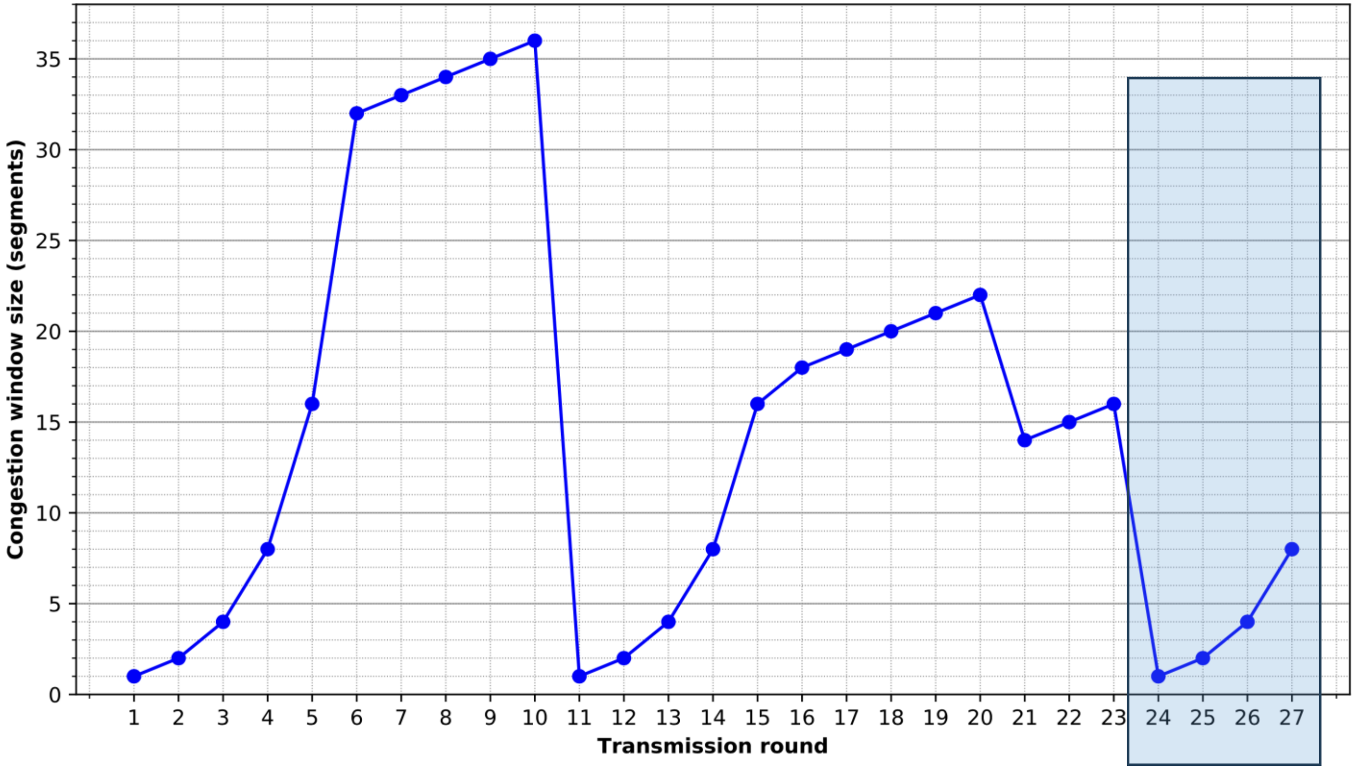


[1,6]

[11,15/16]

Question 1 (a)

Identify the time intervals when TCP slow start is operating.



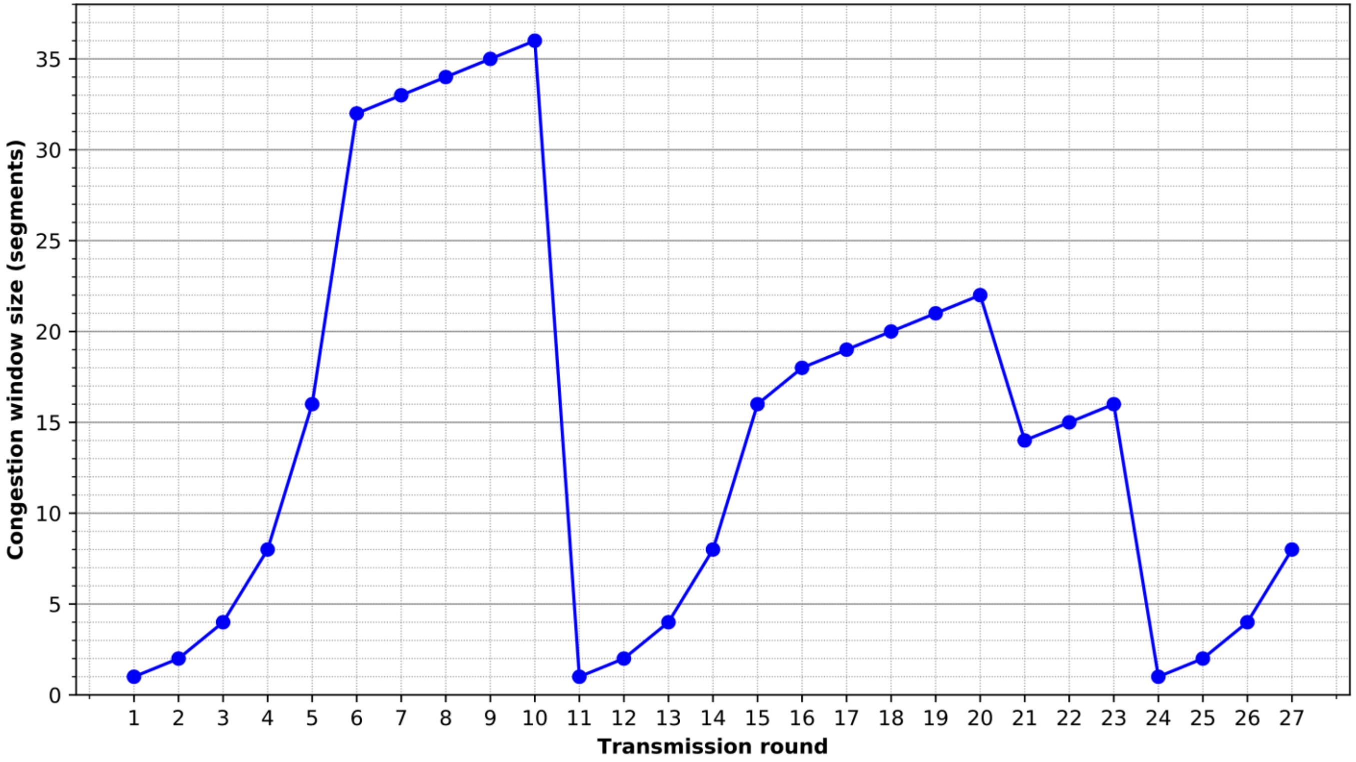
[1,6]

[11,15/16]

[24,27]

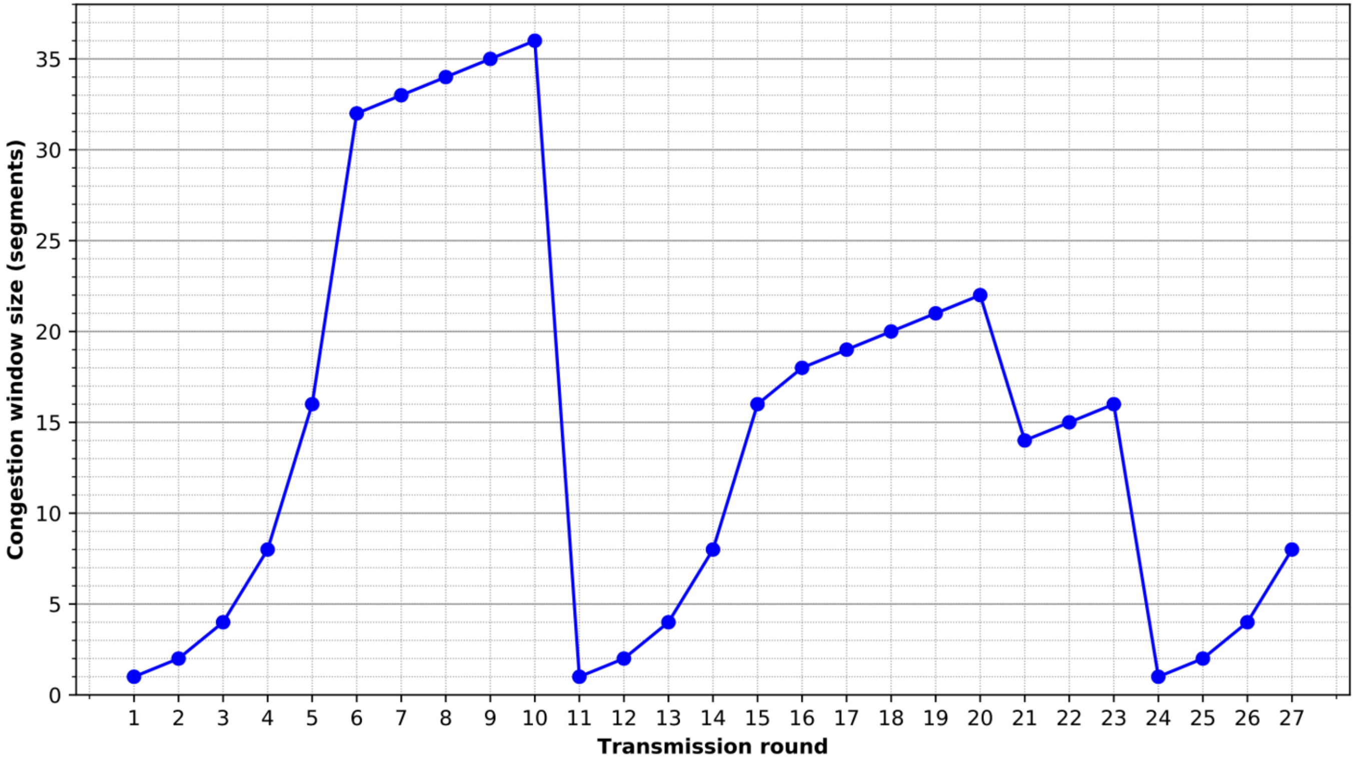
Question 1 (b)

Identify the time intervals when TCP congestion avoidance is used.



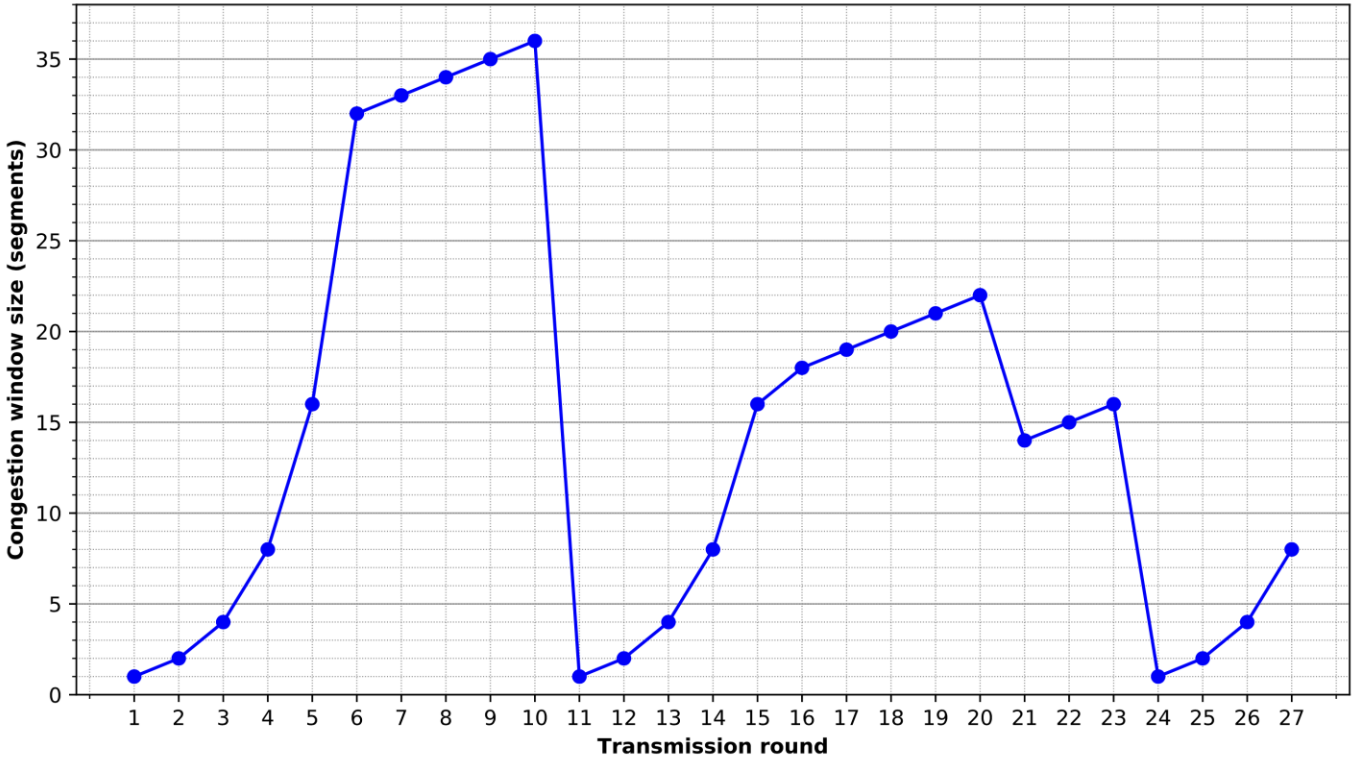
Question 1 (b)

Identify the time intervals when TCP congestion avoidance is used.



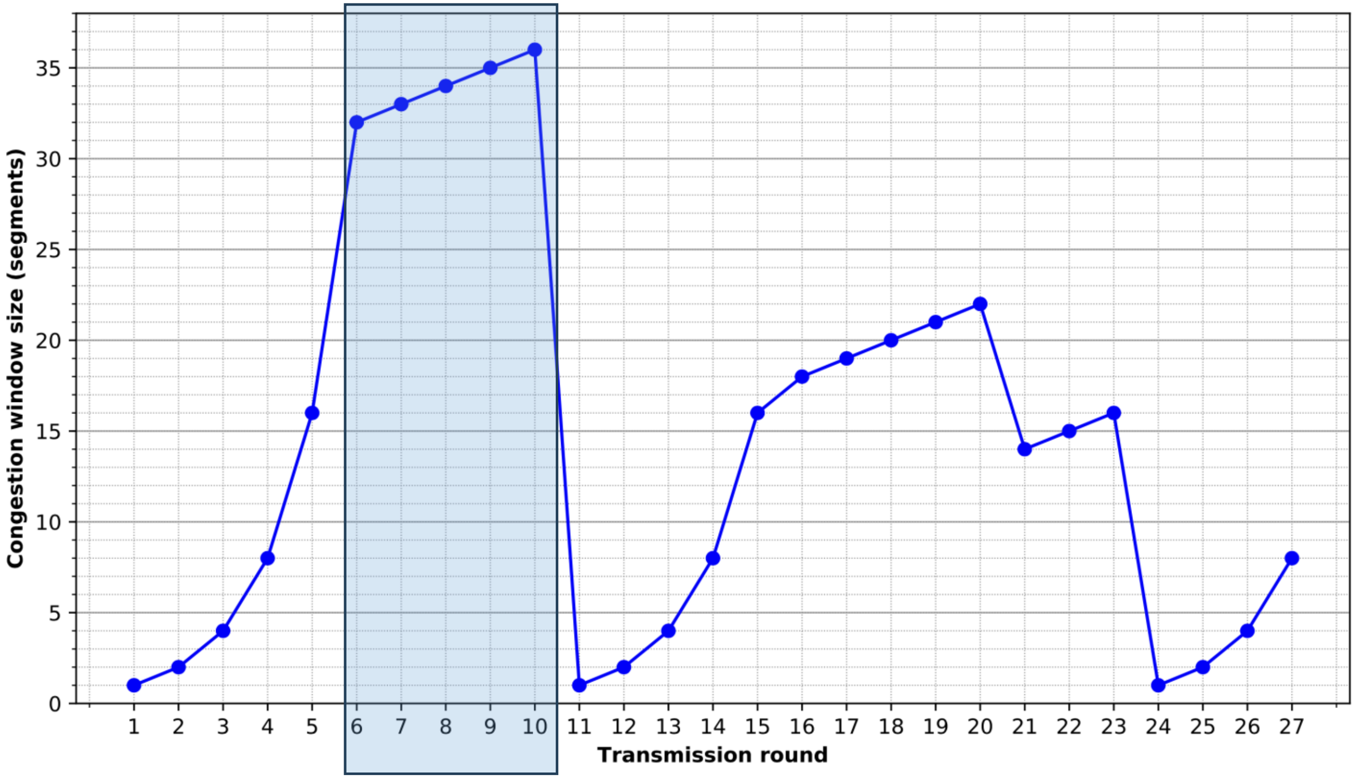
Question 1 (b)

Identify the time intervals when TCP congestion avoidance is used.



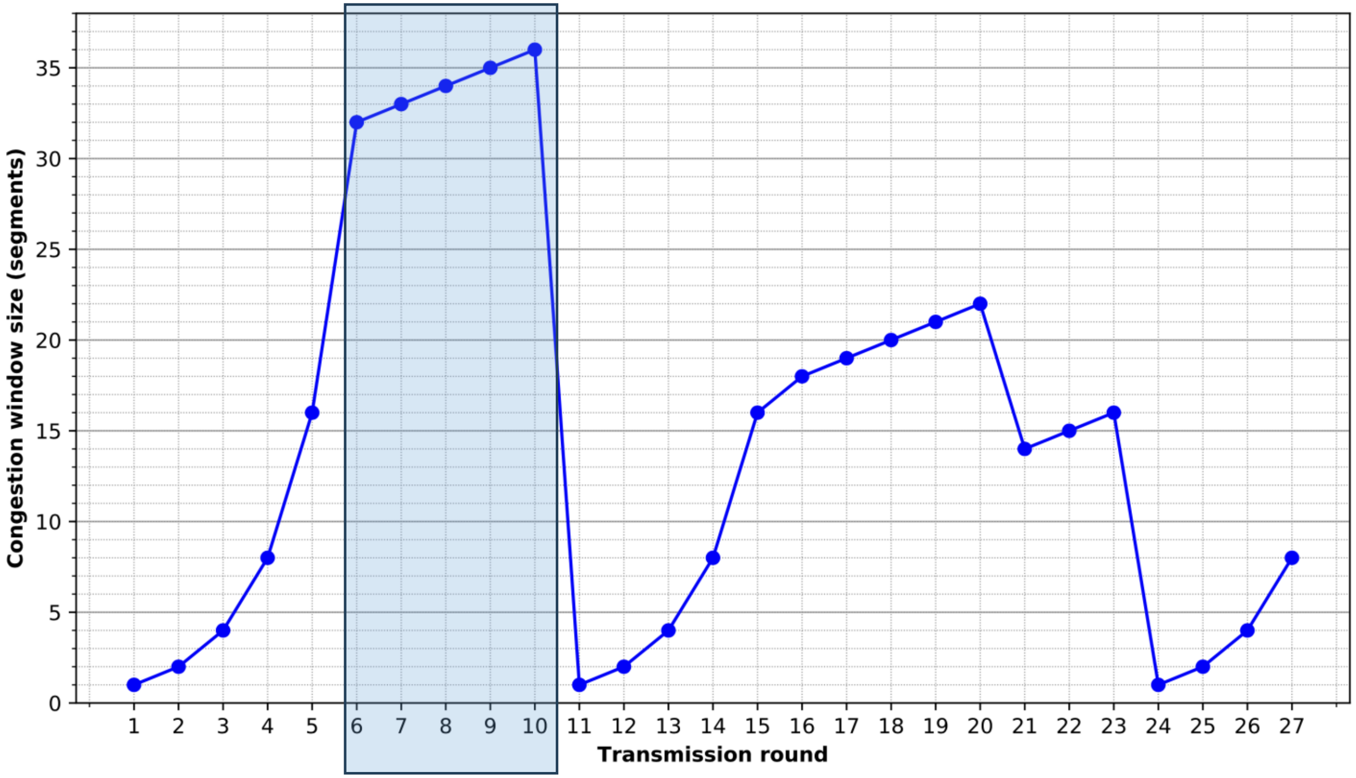
Question 1 (b)

Identify the time intervals when TCP congestion avoidance is used.



Question 1 (b)

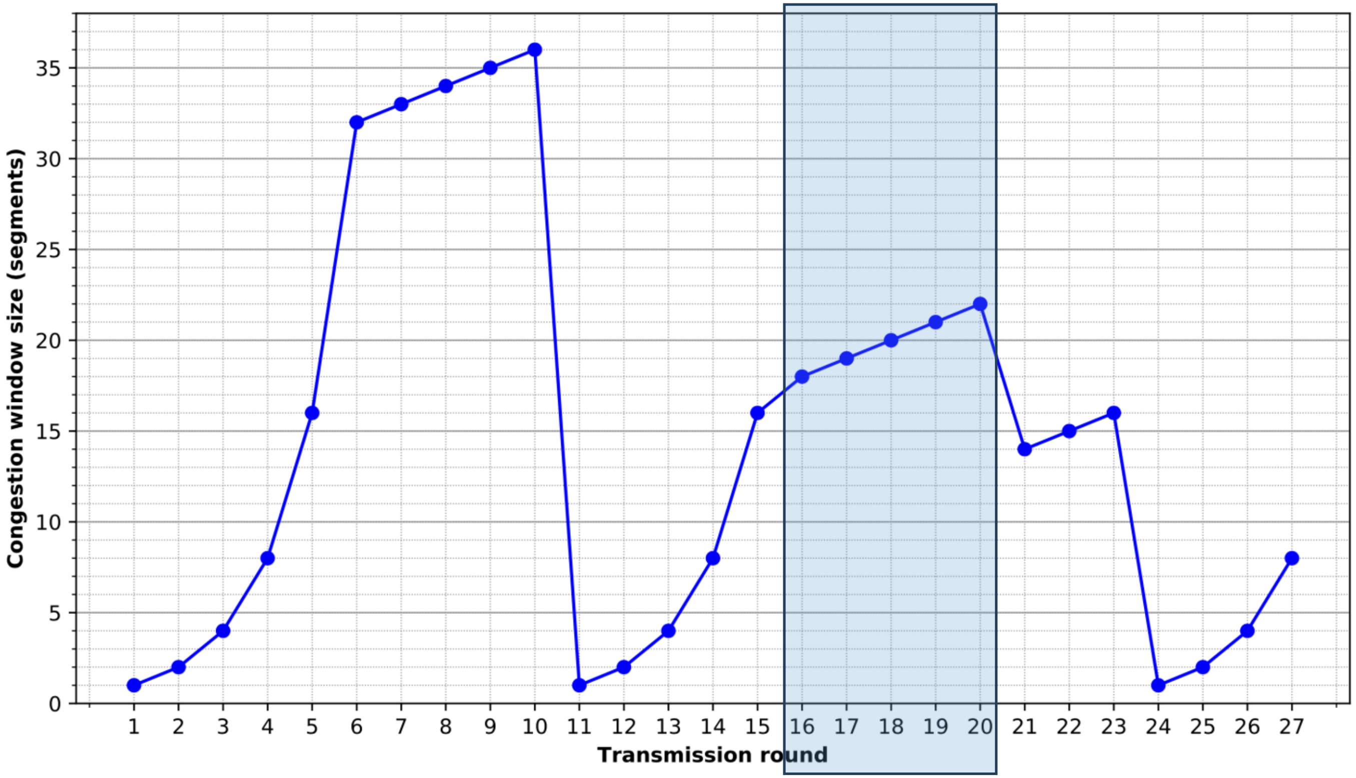
Identify the time intervals when TCP congestion avoidance is used.



[6,10]

Question 1 (b)

Identify the time intervals when TCP congestion avoidance is used.

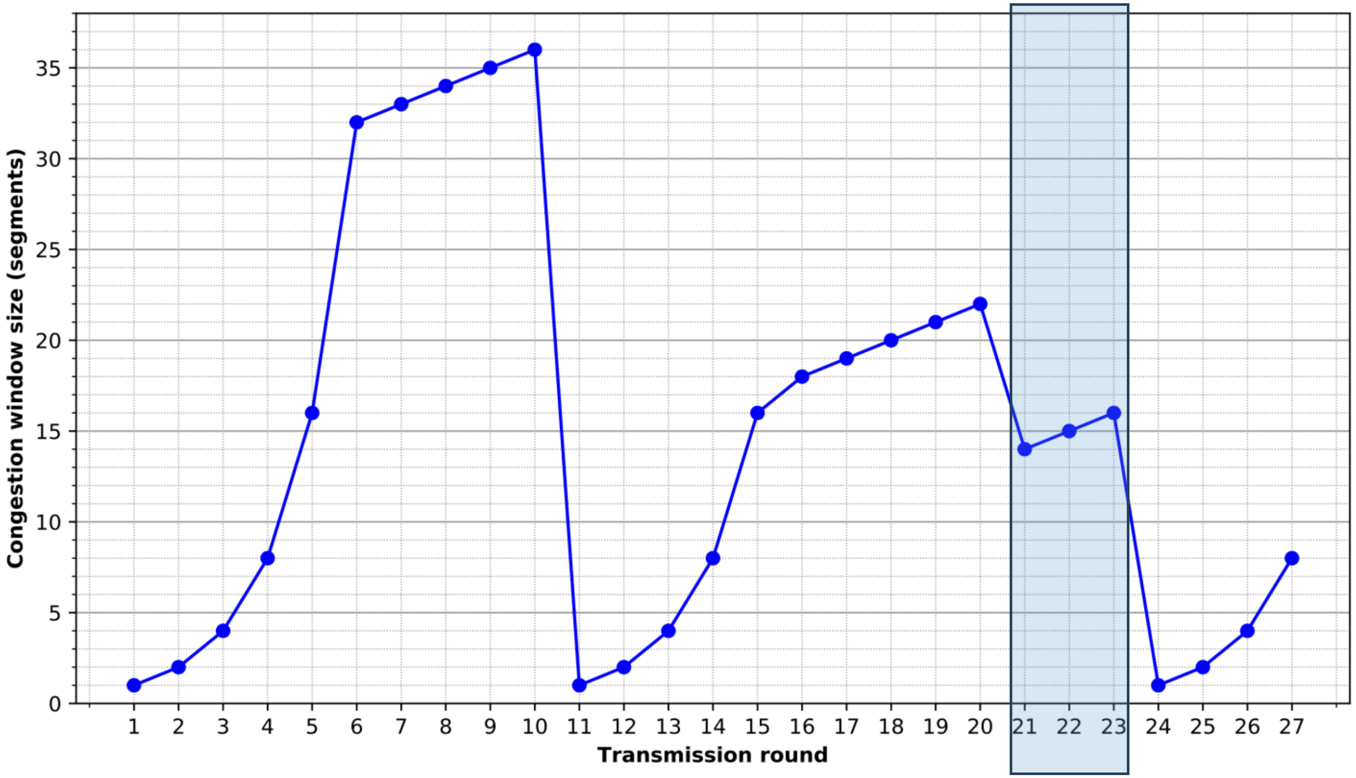


[6,10]

[16,20]

Question 1 (b)

Identify the time intervals when TCP congestion avoidance is used.

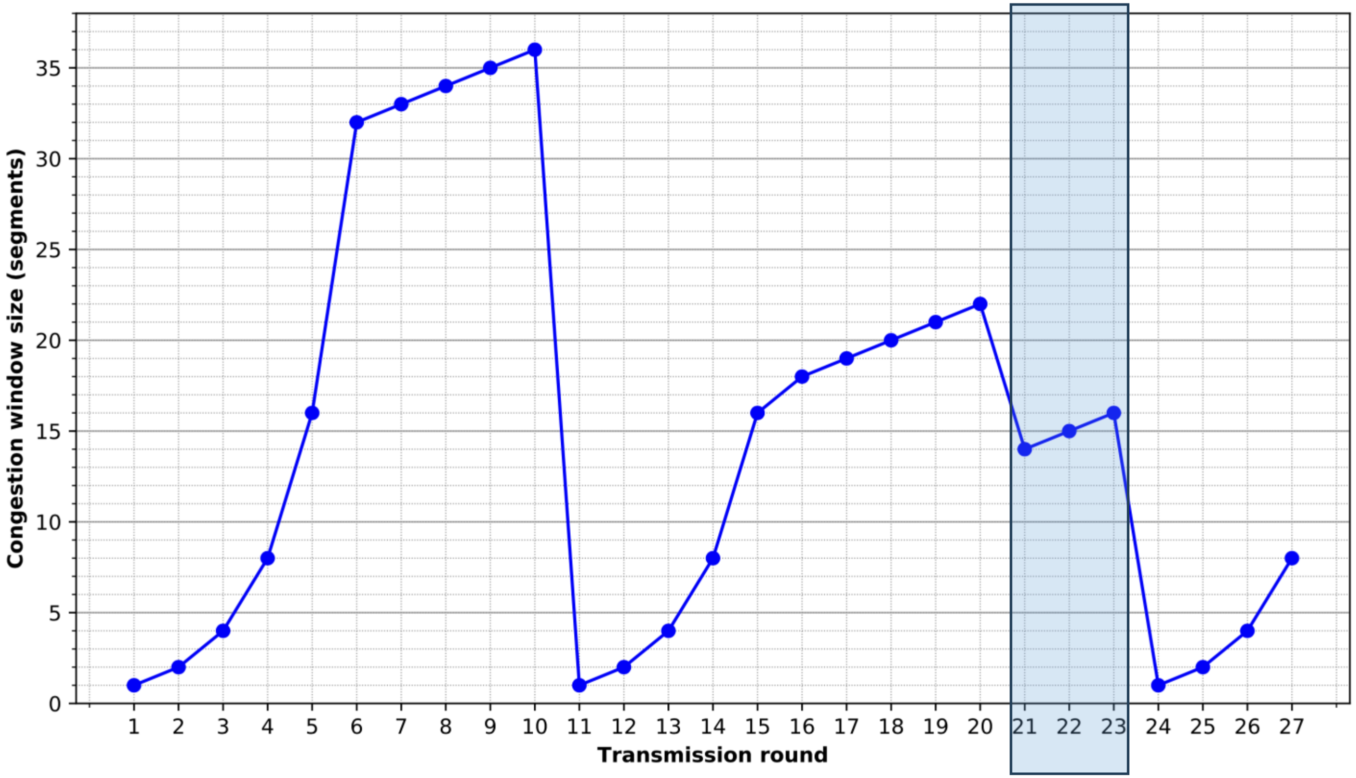


[6,10]

[16,20]

Question 1 (b)

Identify the time intervals when TCP congestion avoidance is used.



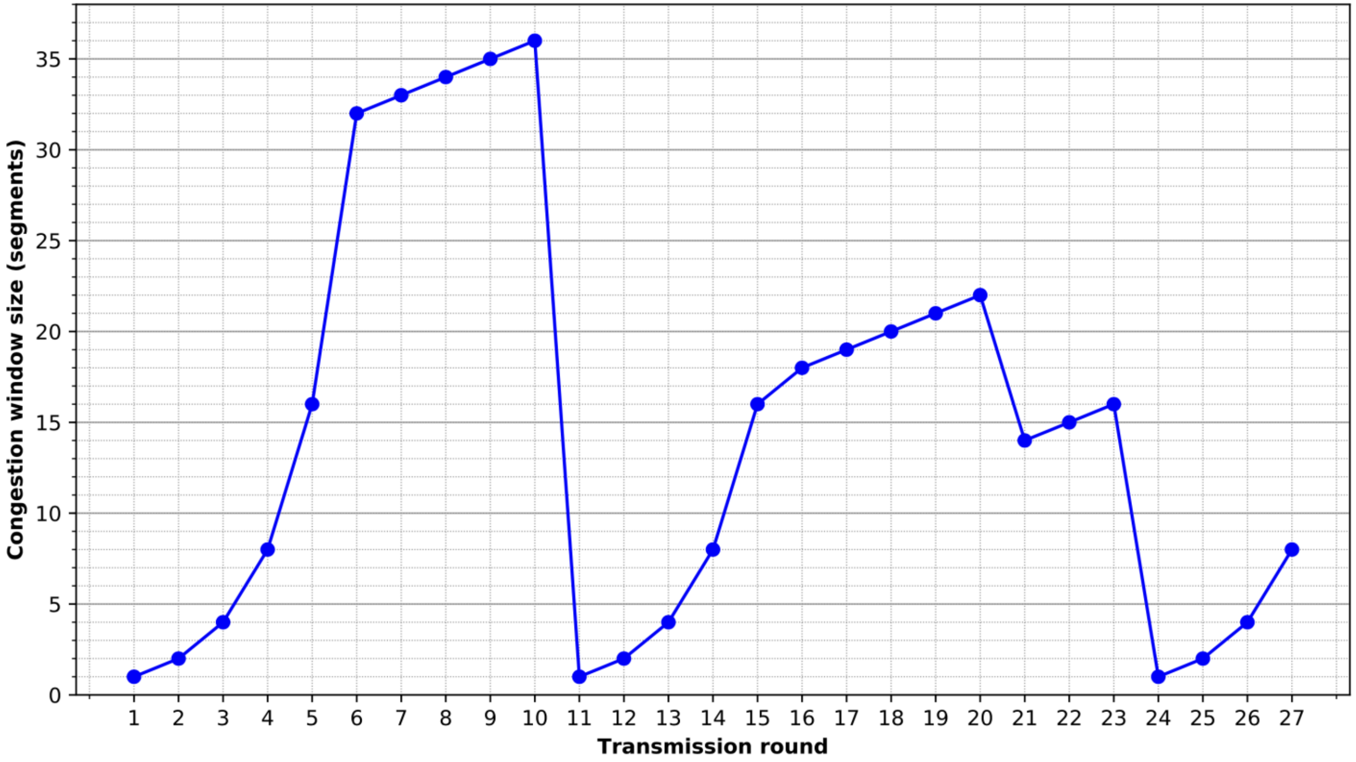
[6,10]

[16,20]

[21,23]

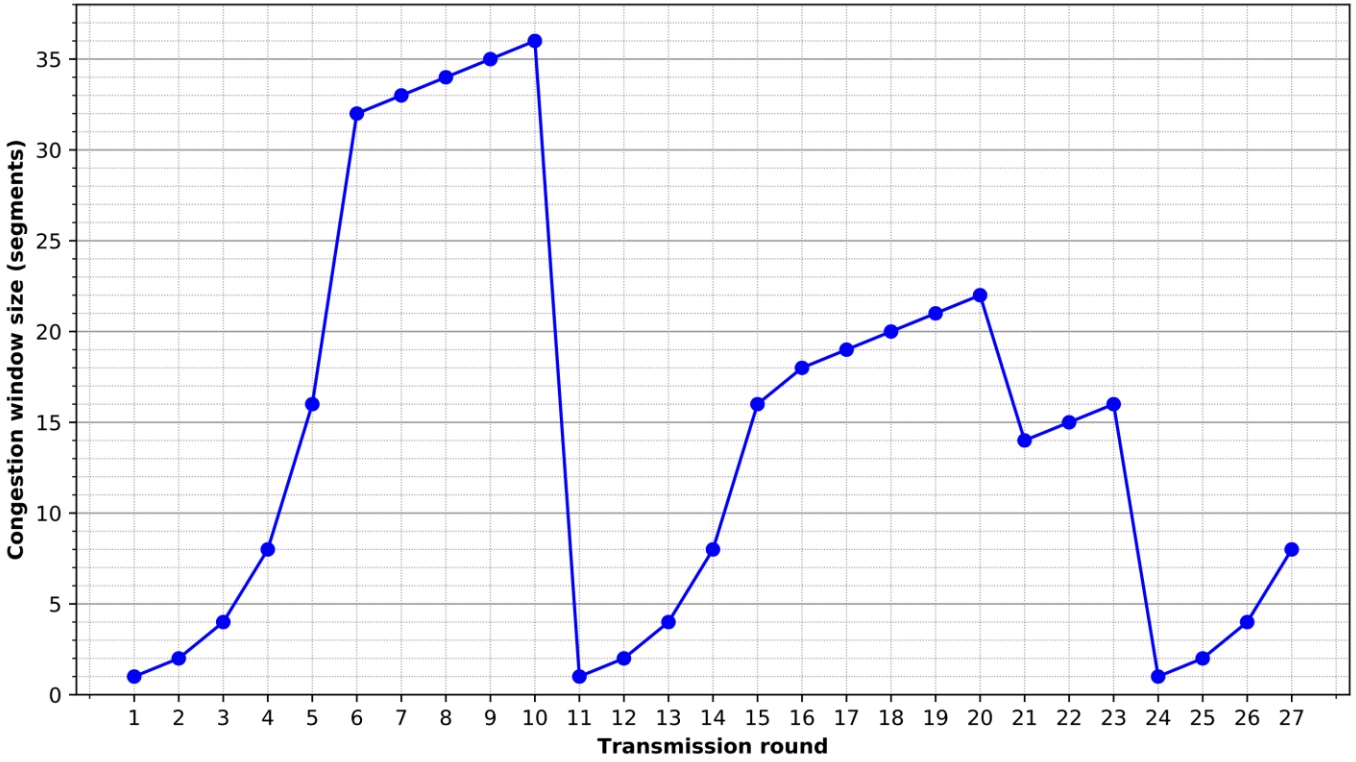
Question 1 (c)

After the 10th transmission round, how is the segment loss detected by the sender? Justify your answer.



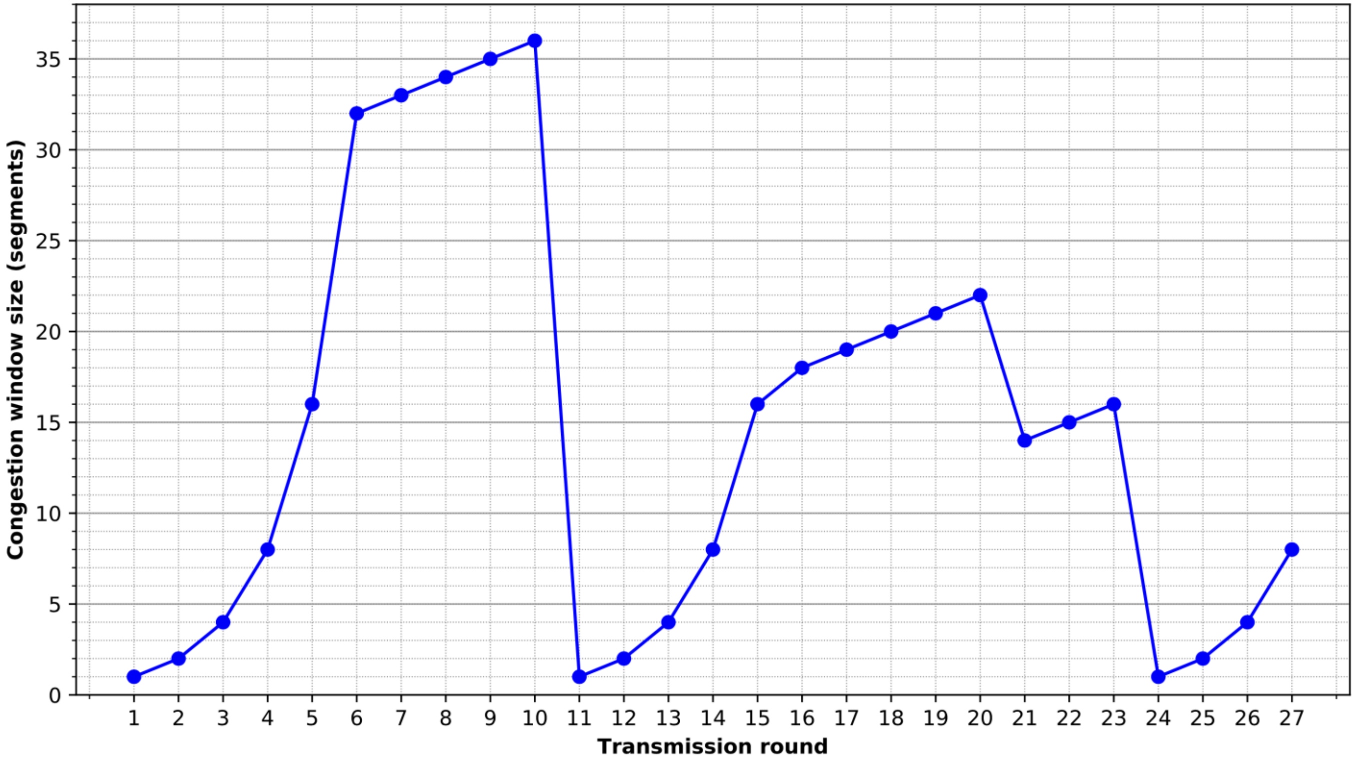
Question 1 (c)

After the 10th transmission round, how is the segment loss detected by the sender?



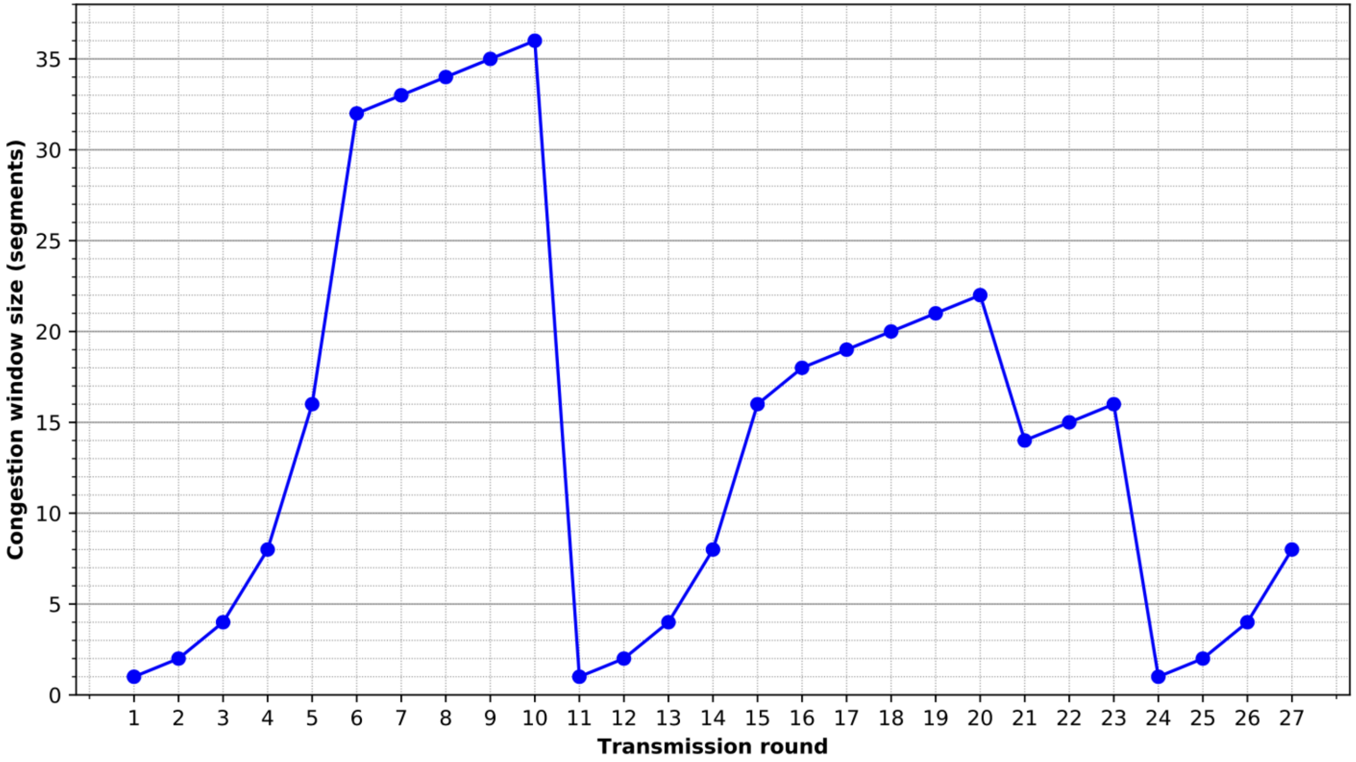
Question 1 (c)

After the 10th transmission round, how is the segment loss detected by the sender?



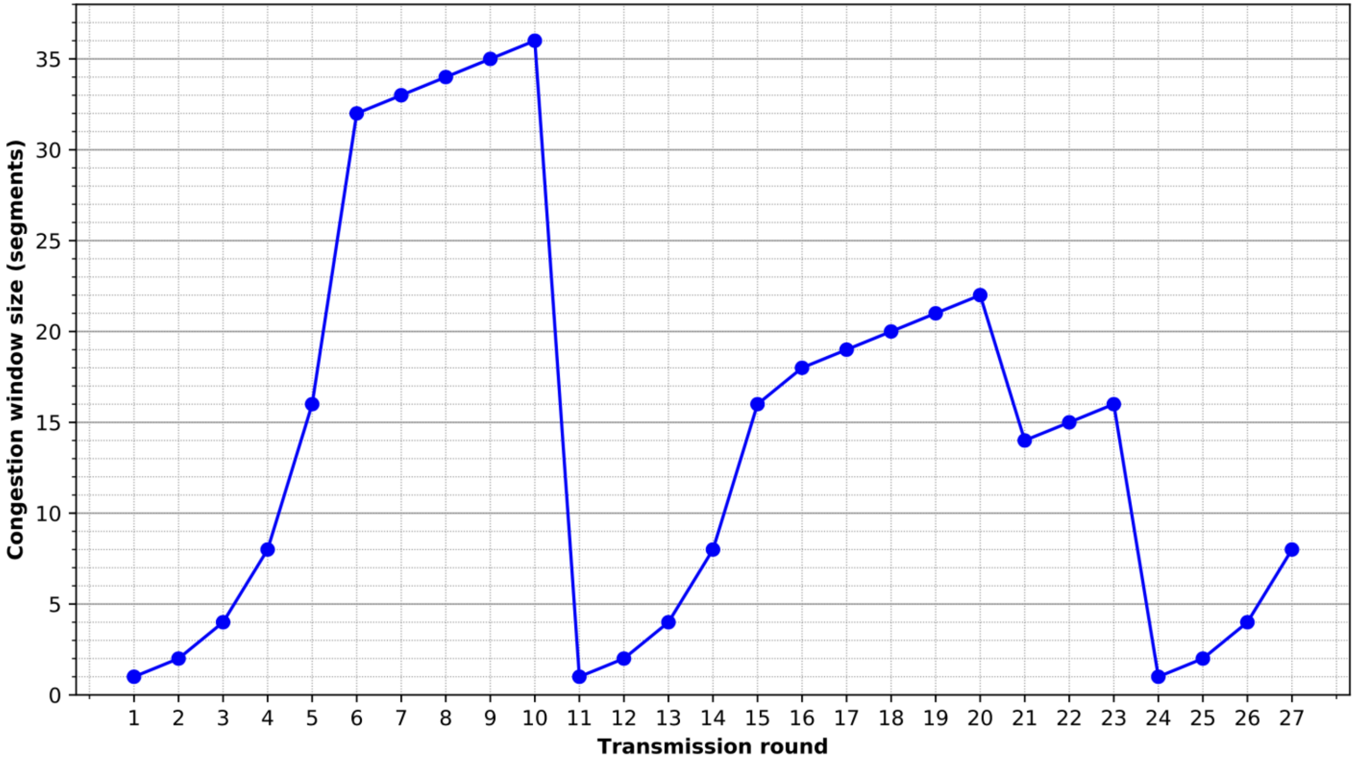
Question 1 (c)

After the 10th transmission round, how is the segment loss detected by the sender?



Question 1 (c)

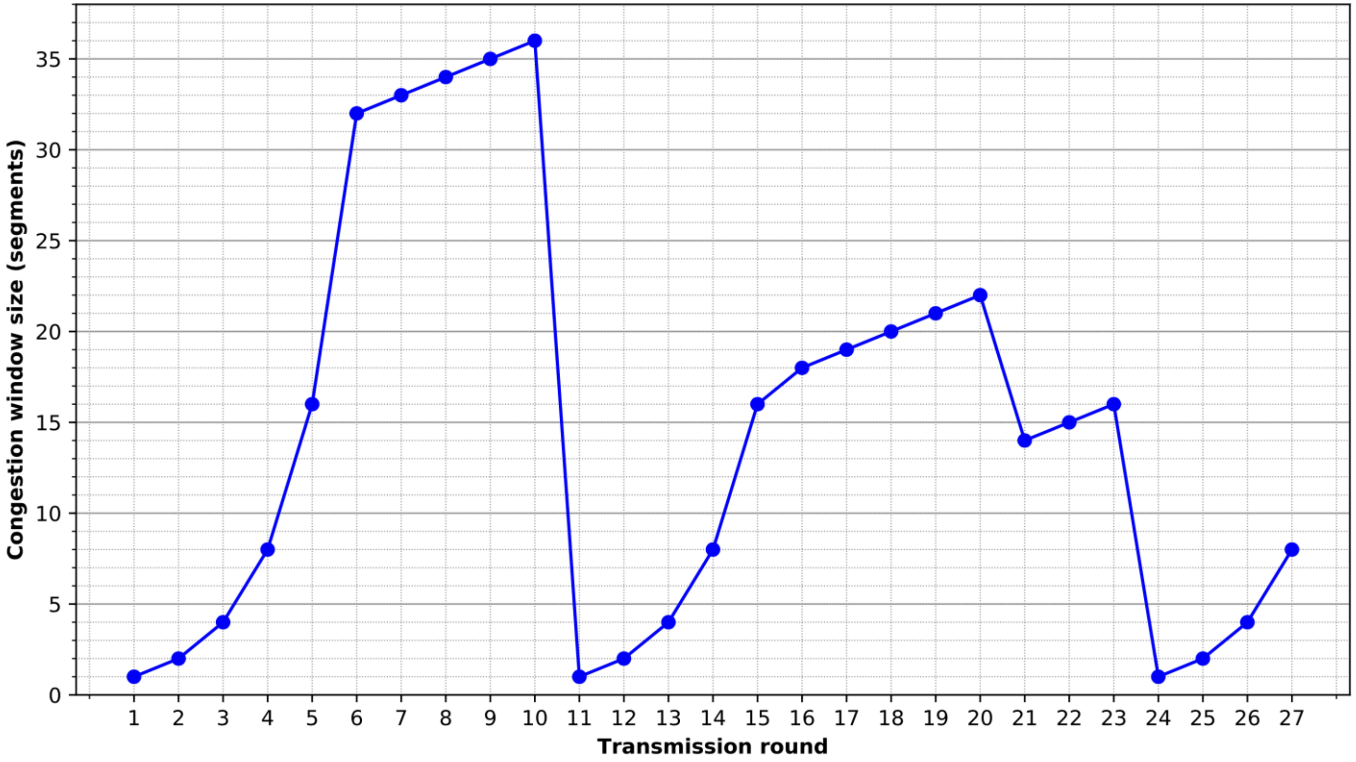
After the 10th transmission round, how is the segment loss detected by the sender?



The sender detected a timeout, because the CWND dropped to 1.

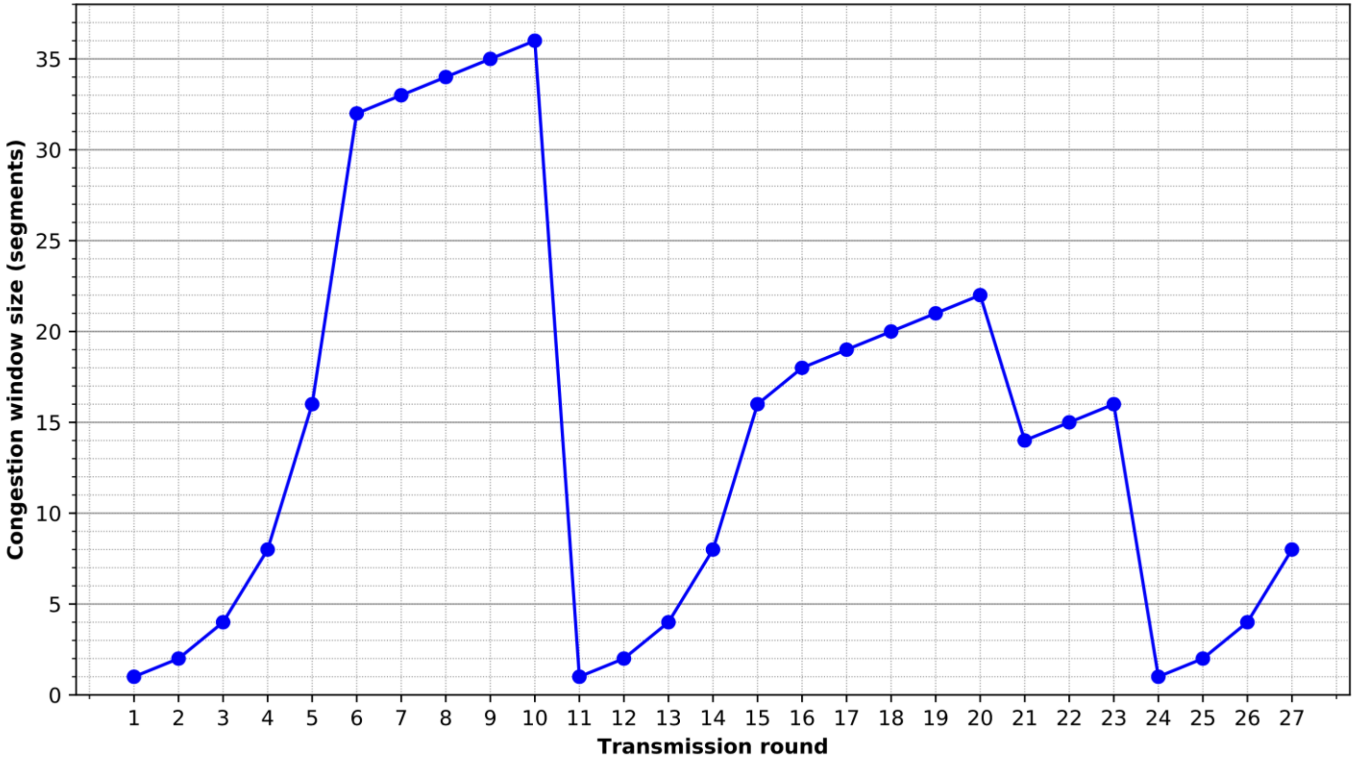
Question 1 (d)

After the 20th transmission round, how is the segment loss detected by the sender?



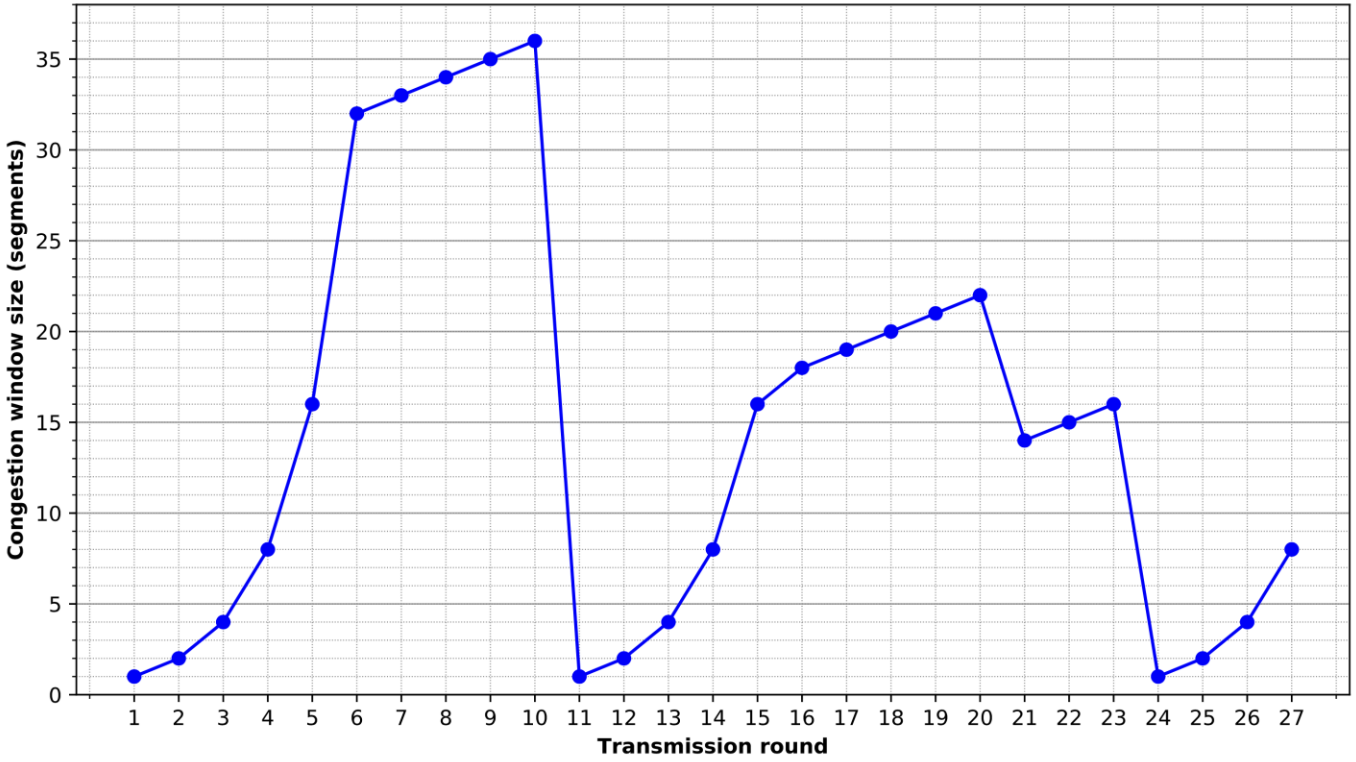
Question 1 (d)

After the 20th transmission round, how is the segment loss detected by the sender?



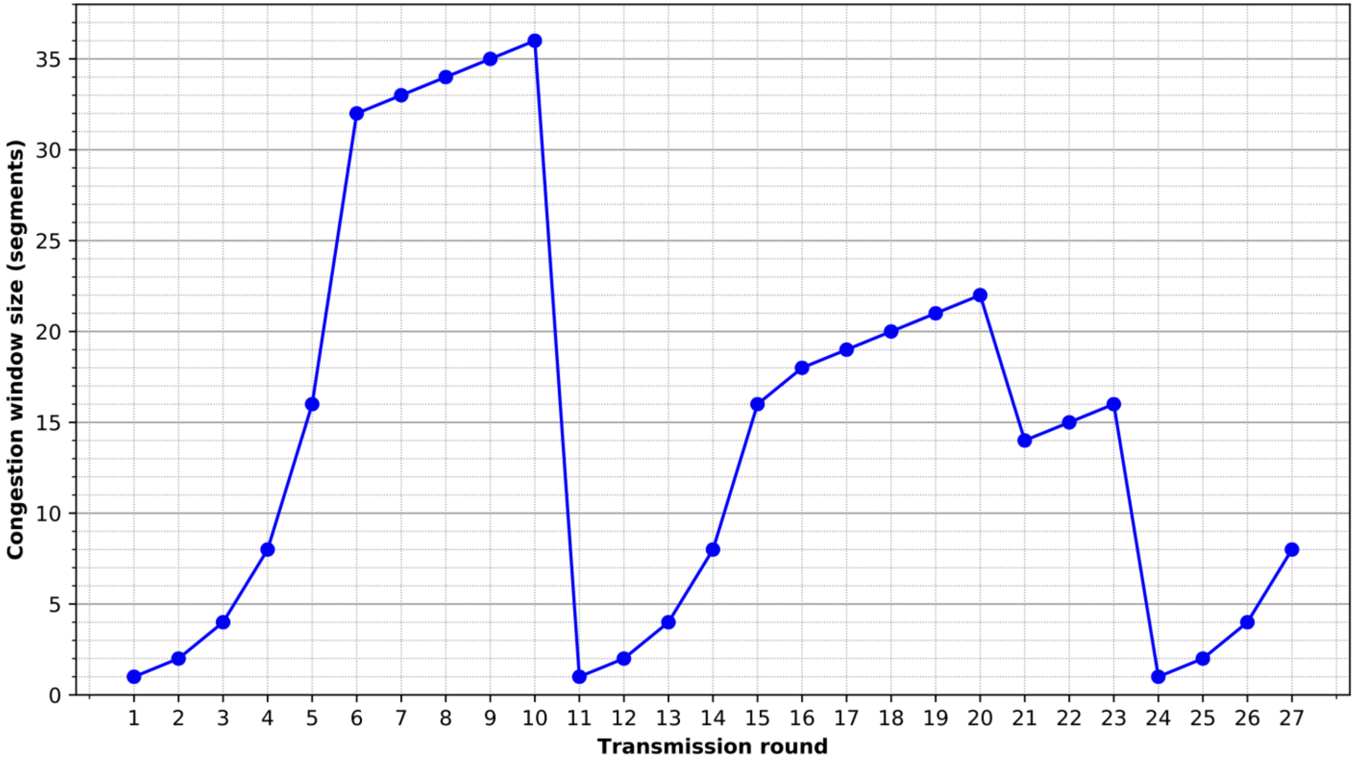
Question 1 (d)

After the 20th transmission round, how is the segment loss detected by the sender?



Question 1 (d)

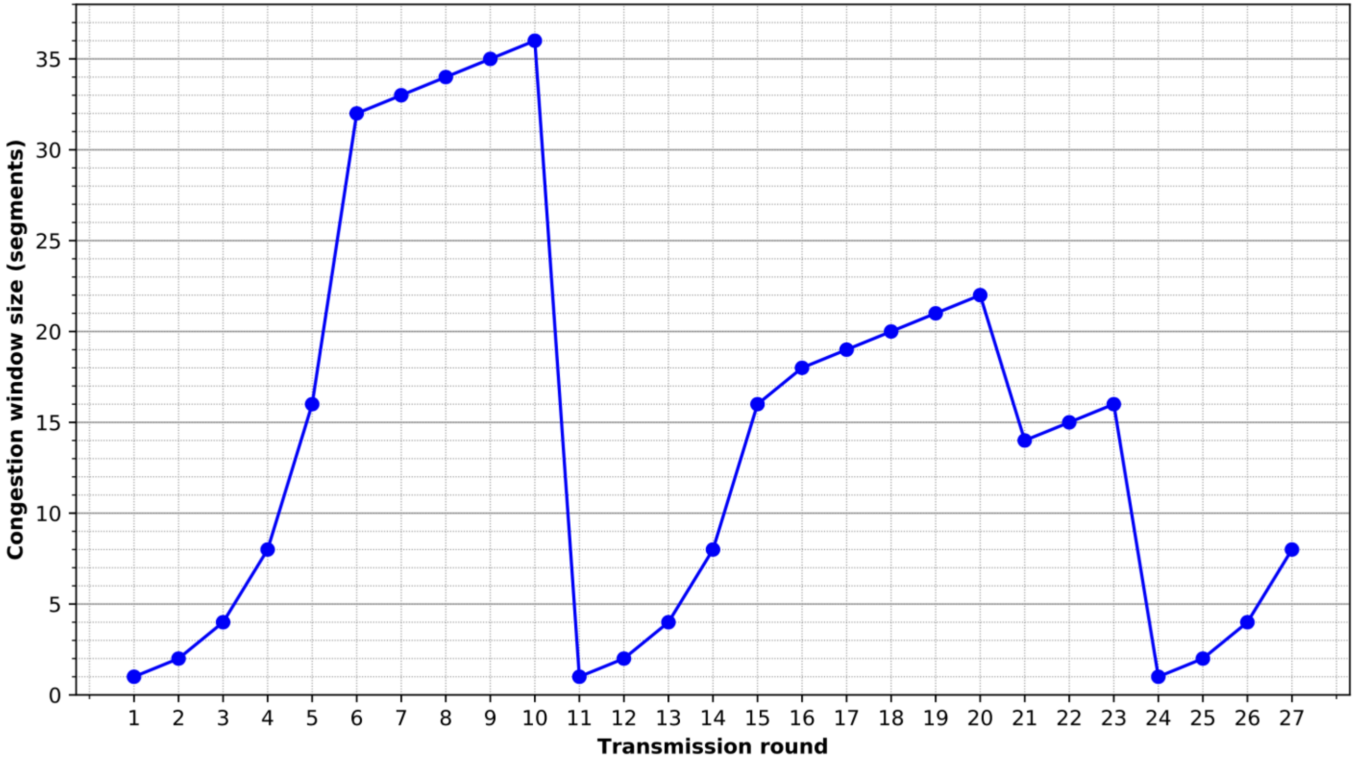
After the 20th transmission round, how is the segment loss detected by the sender?



The sender detected triple duplicated ACKs, because the CWND is halved.

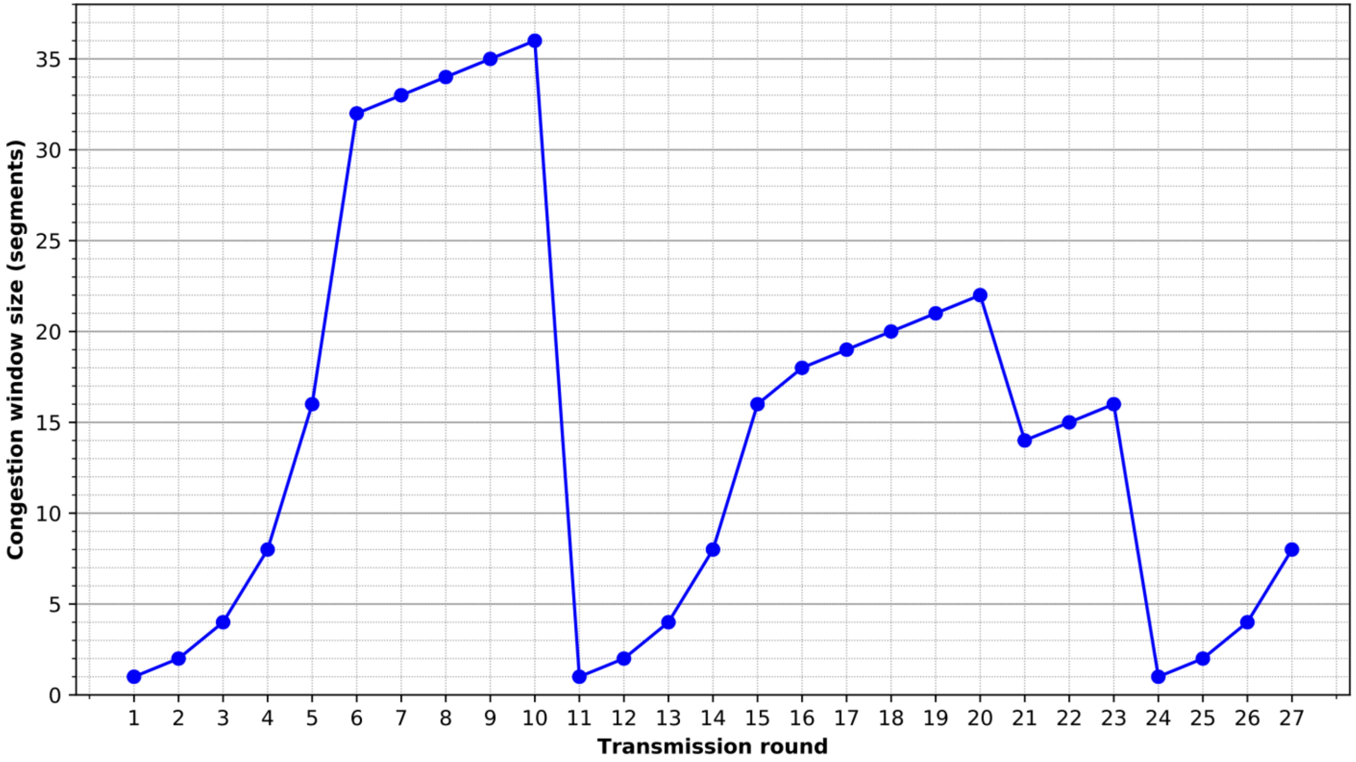
Question 1 (e)

What is the value of Threshold at the 5th, 13th, and 21st transmission round?



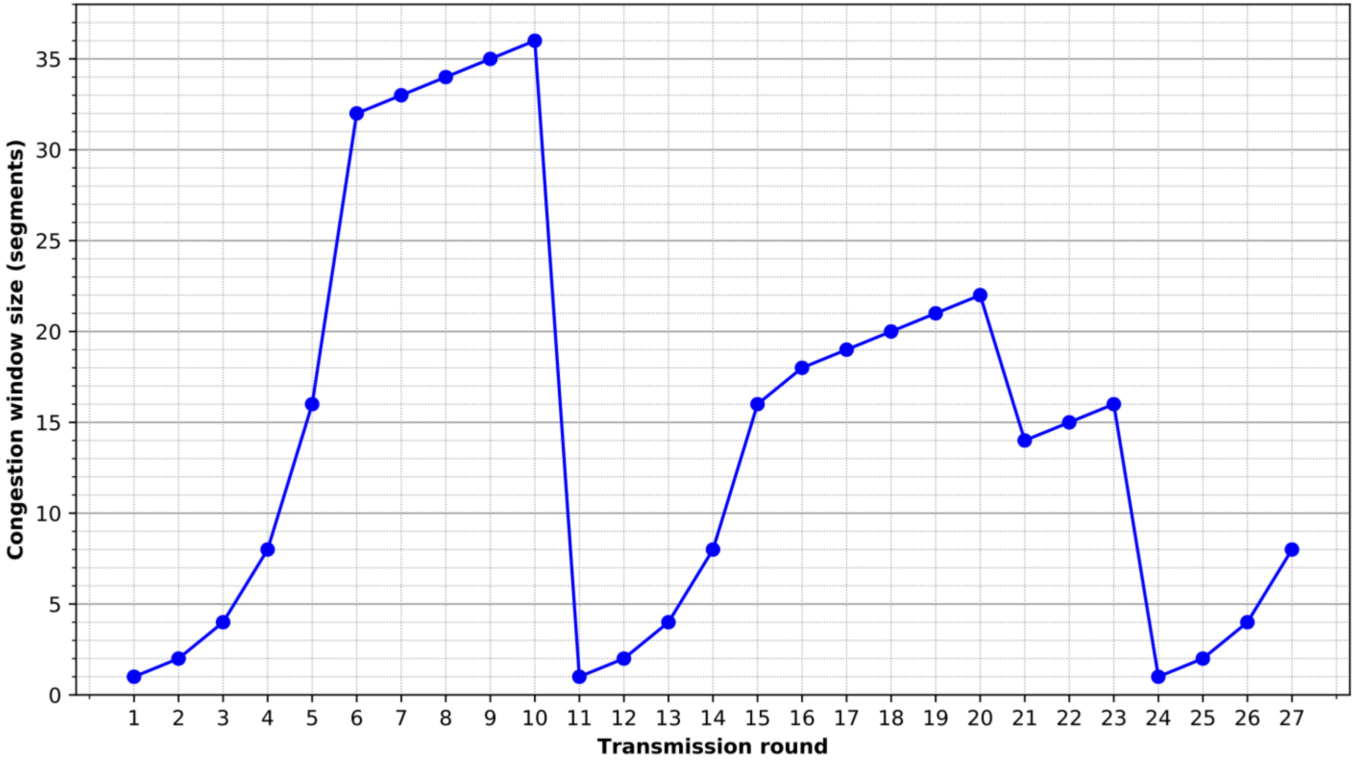
Question 1 (e)

What is the value of Threshold at the 5th, 13th, and 21st transmission round?



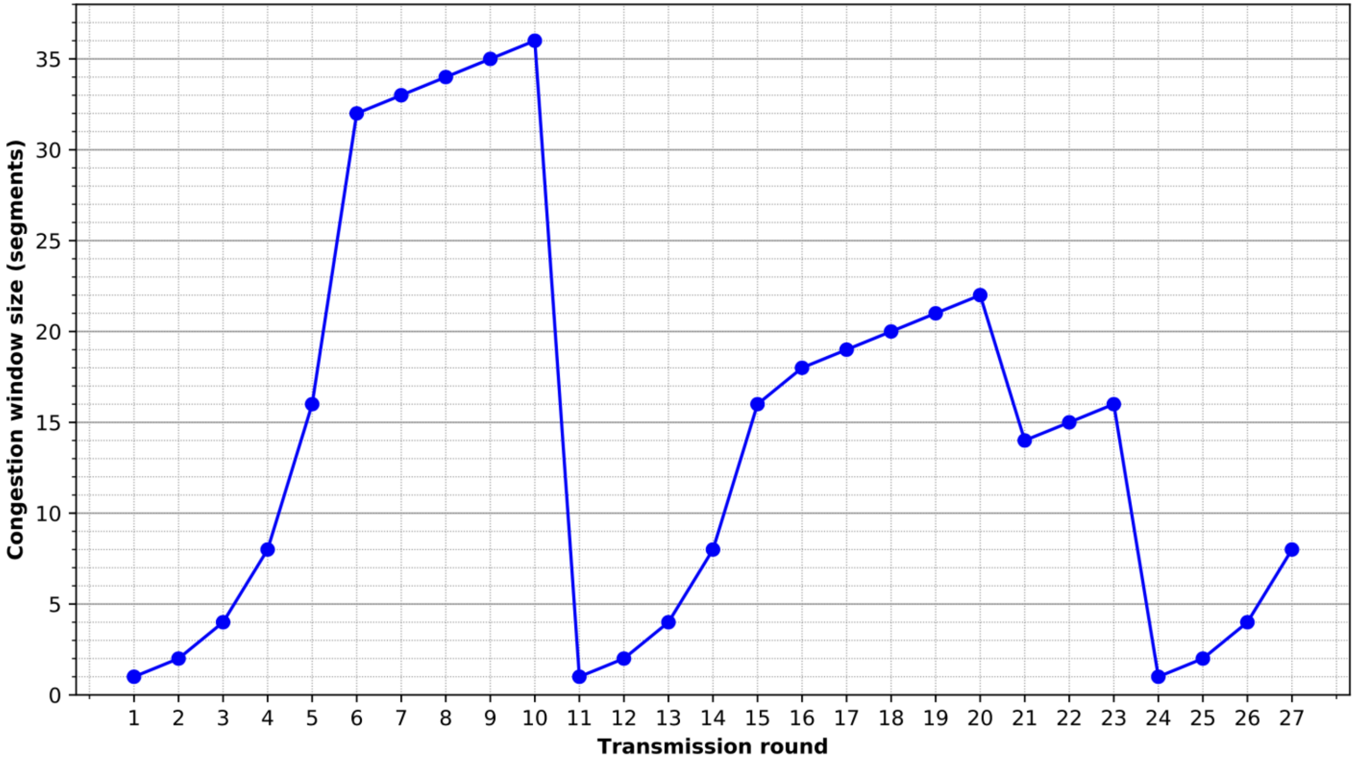
Question 1 (e)

What is the value of Threshold at the 5th, 13th, and 21st transmission round?



Question 1 (e)

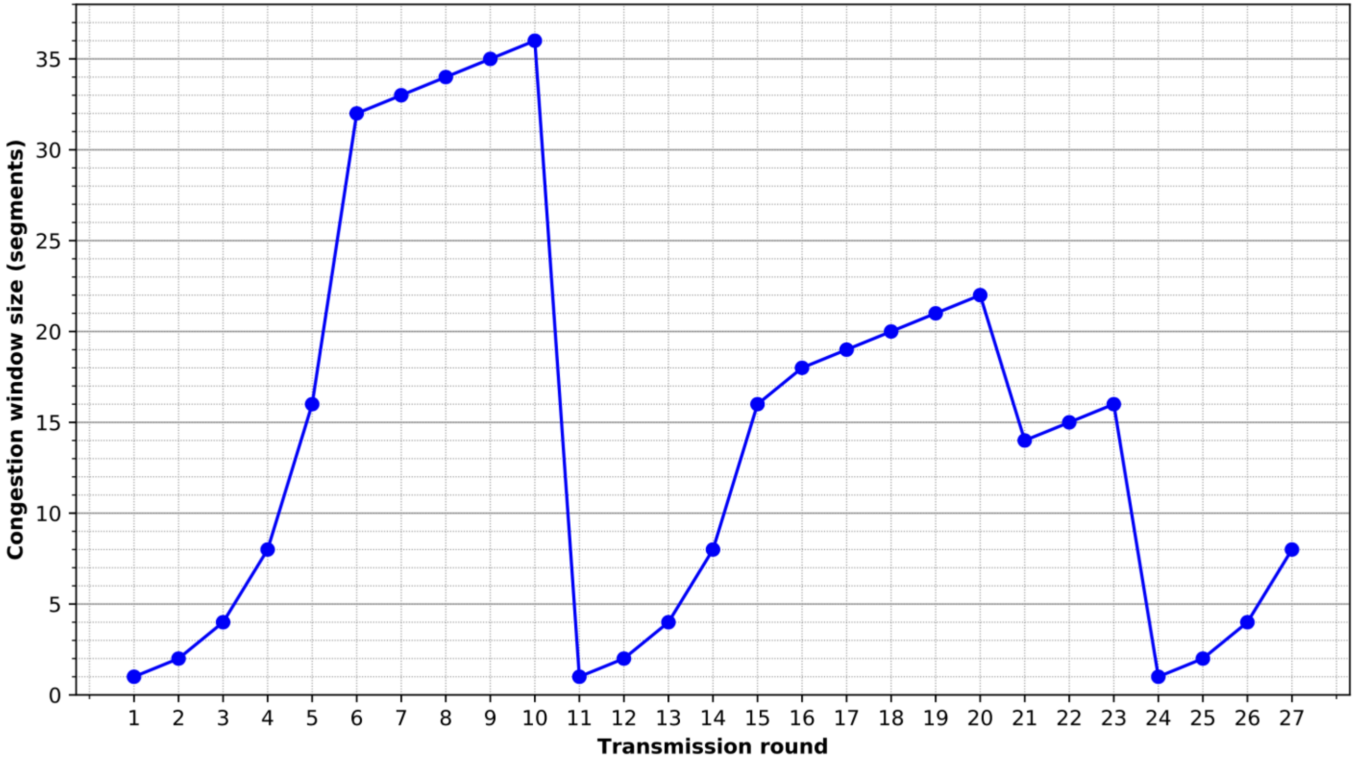
What is the value of Threshold at the 5th, 13th, and 21st transmission round?



For TCP Reno:
ssthresh=cwnd/2;

Question 1 (e)

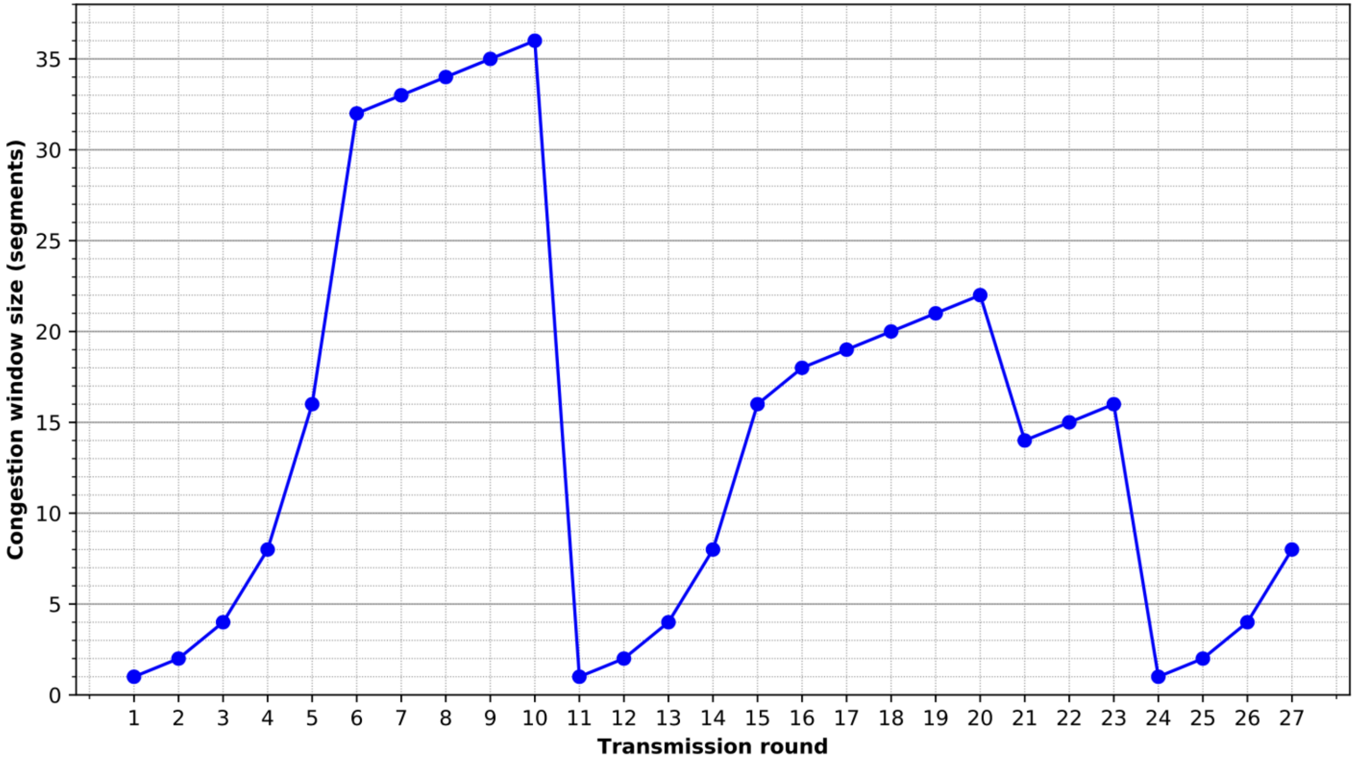
What is the value of Threshold at the 5th, 13th, and 21st transmission round?



For TCP Reno:
ssthresh=cwnd/2;
cwnd=ssthresh+3MSS for fast retransmits,
cwnd=1 for timeouts

Question 1 (e)

What is the value of Threshold at the 5th, 13th, and 21st transmission round?

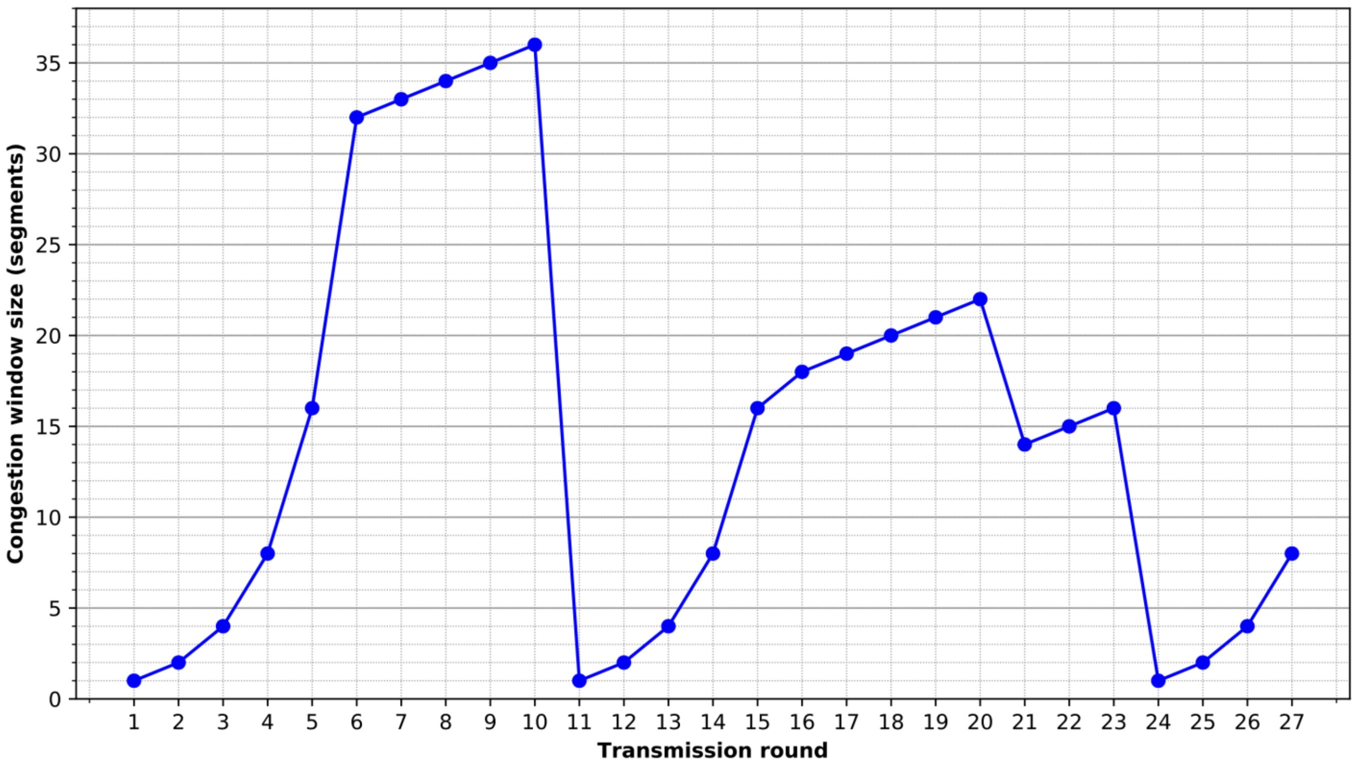


For TCP Reno:
ssthresh=cwnd/2;
cwnd=ssthresh+3MSS for fast retransmits,
cwnd=1 for timeouts

5th: 32
13th: 36/2=18
21st: 22/2=11

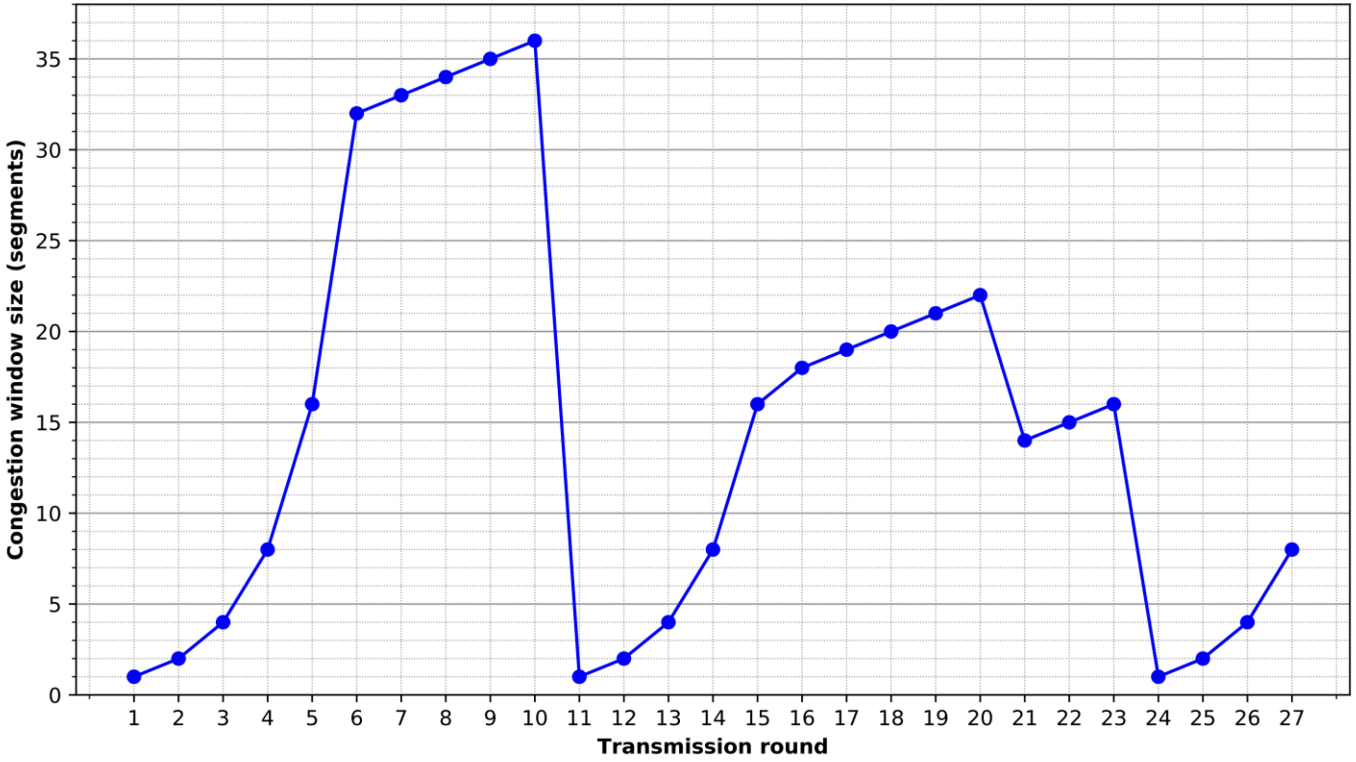
Question 1 (f)

During which transmission round is the 30th segment sent?



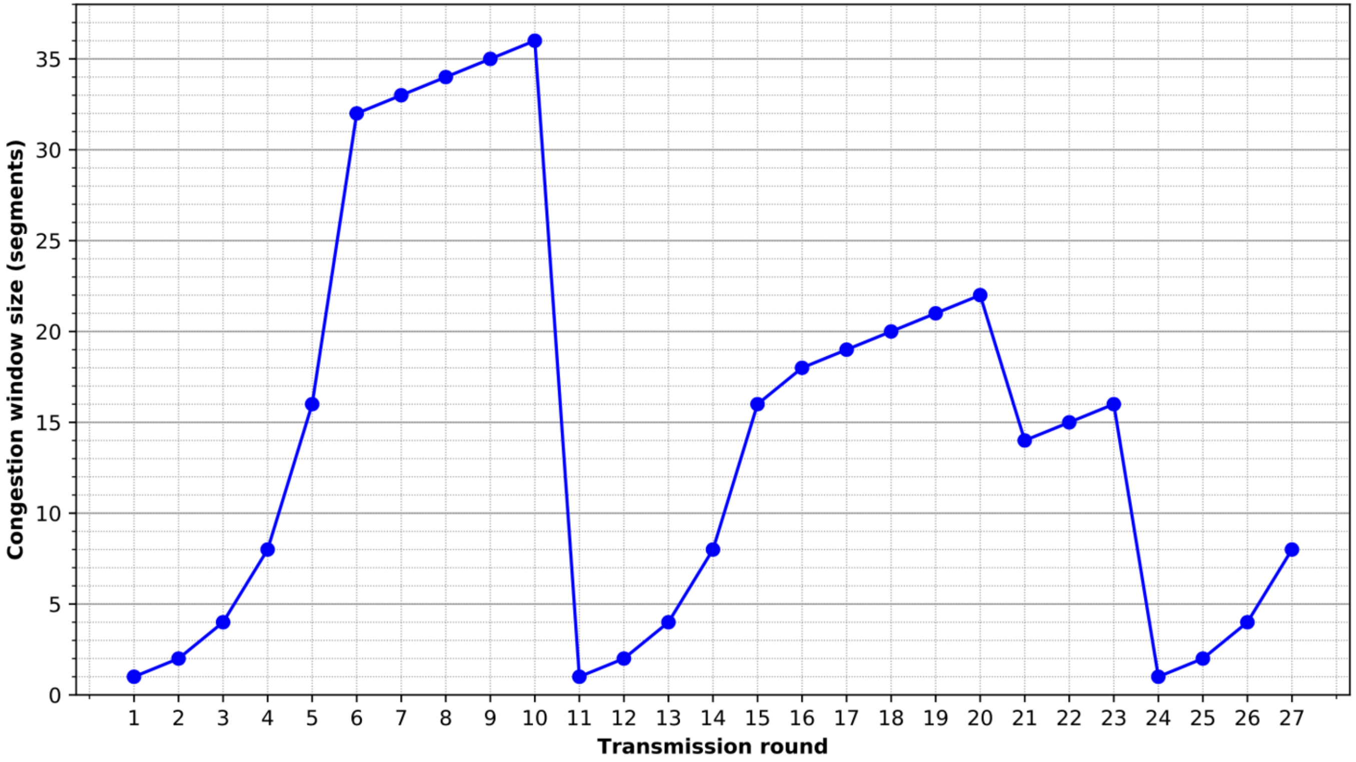
Question 1 (f)

During which transmission round is the 30th segment sent?



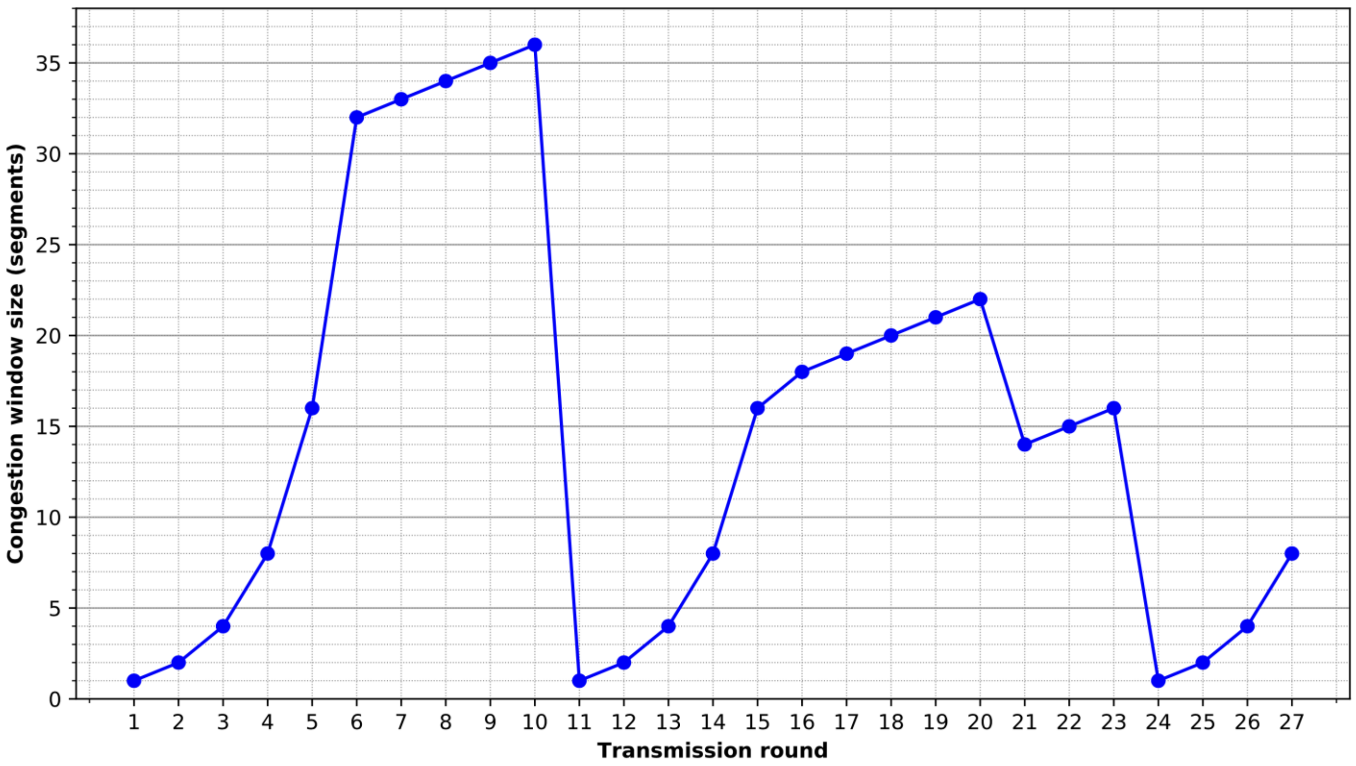
Question 1 (f)

During which transmission round is the 30th segment sent?



Question 1 (f)

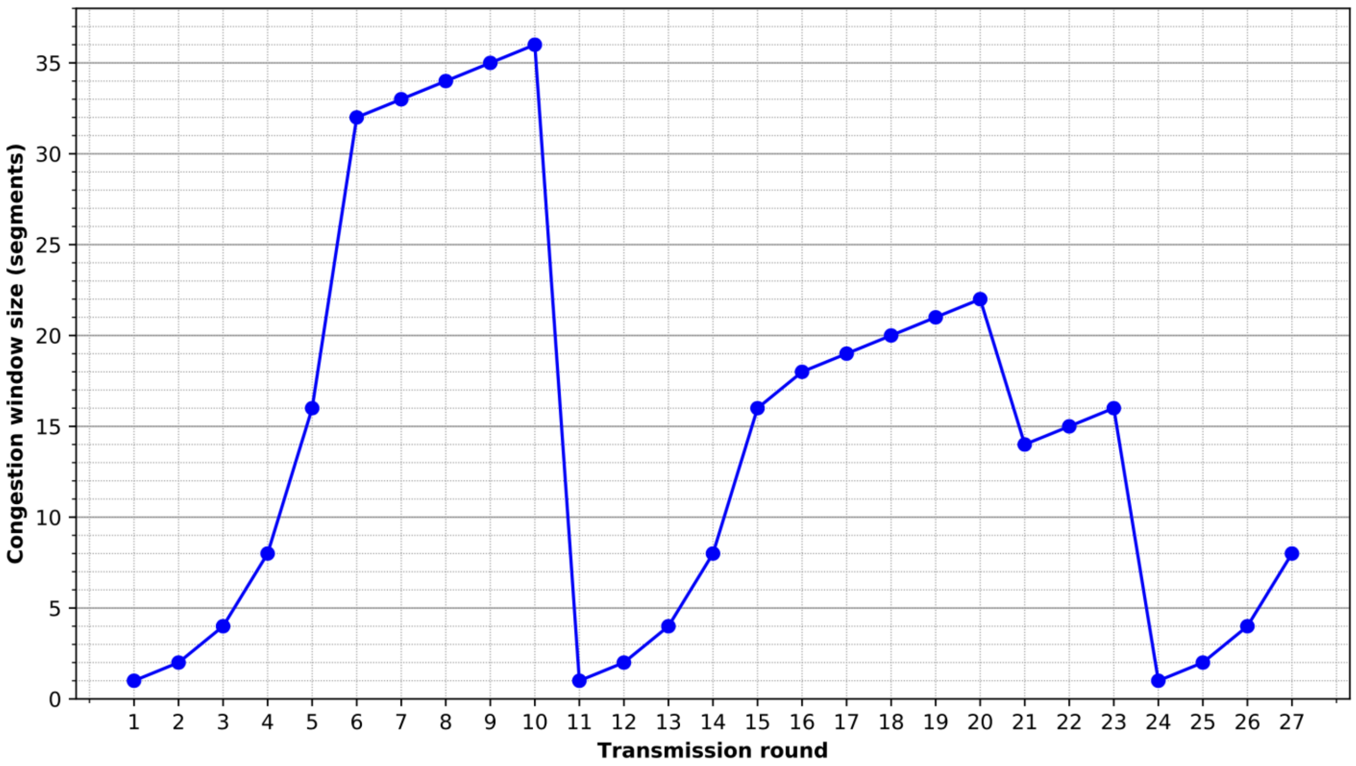
During which transmission round is the 30th segment sent?



During 5th transmission round.

Question 1 (f)

During which transmission round is the 30th segment sent?

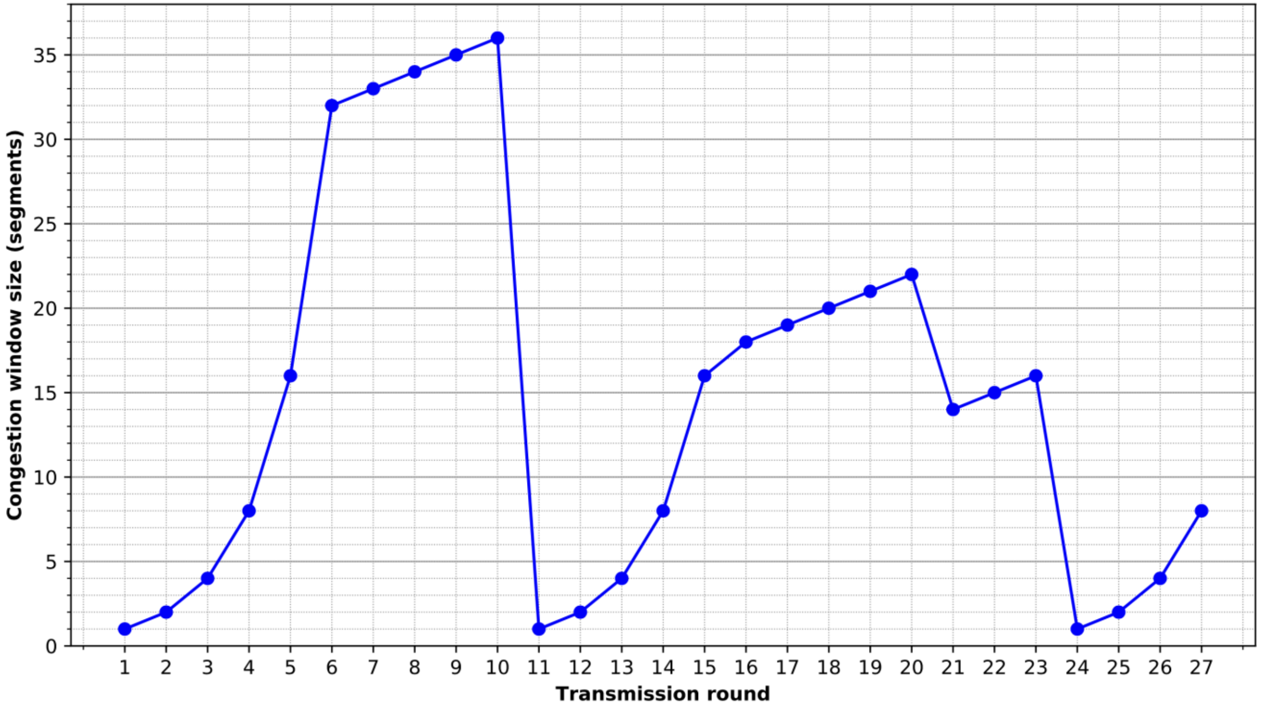


During 5th transmission round.

1st round: 1 sent in total
2nd round: 3 sent in total
3rd round: 7 sent in total
4th round: 15 sent in total
5th round: 31 sent in total

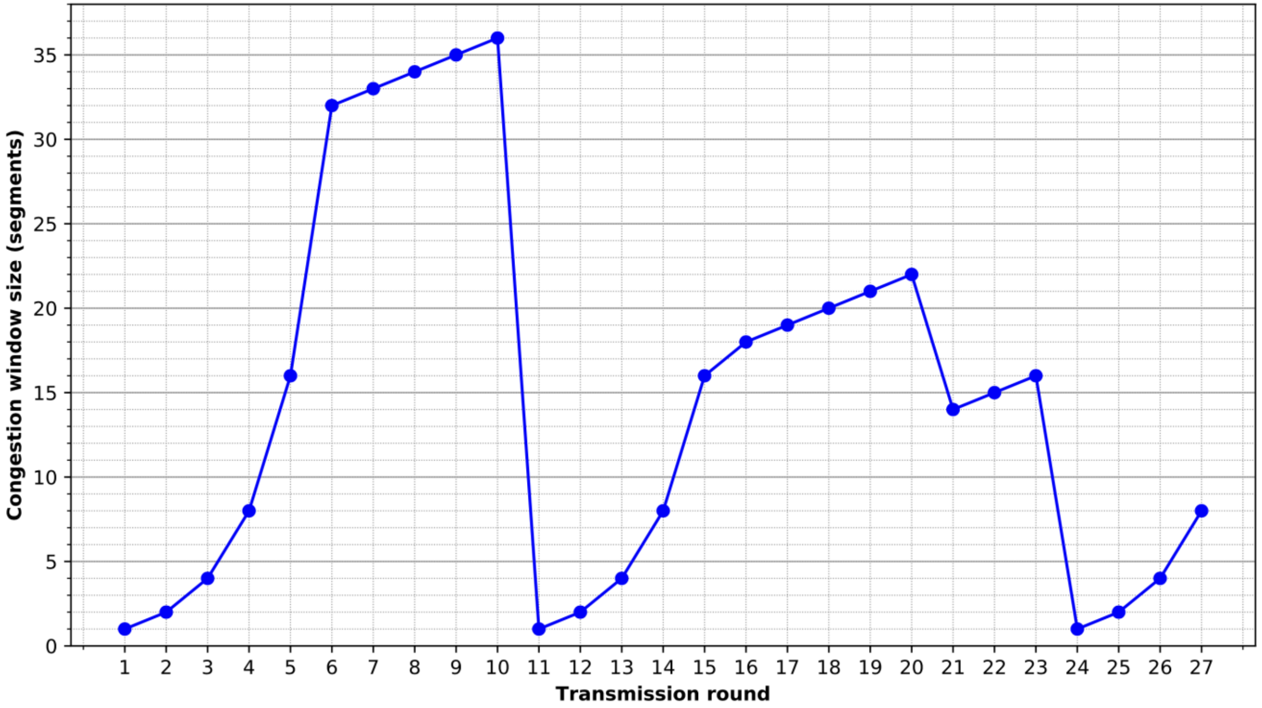
Question 1 (g)

Assuming a packet loss is detected after the 27th round by the reception of a triple duplicate acknowledgement, what will be the values of the congestion window size and Threshold?



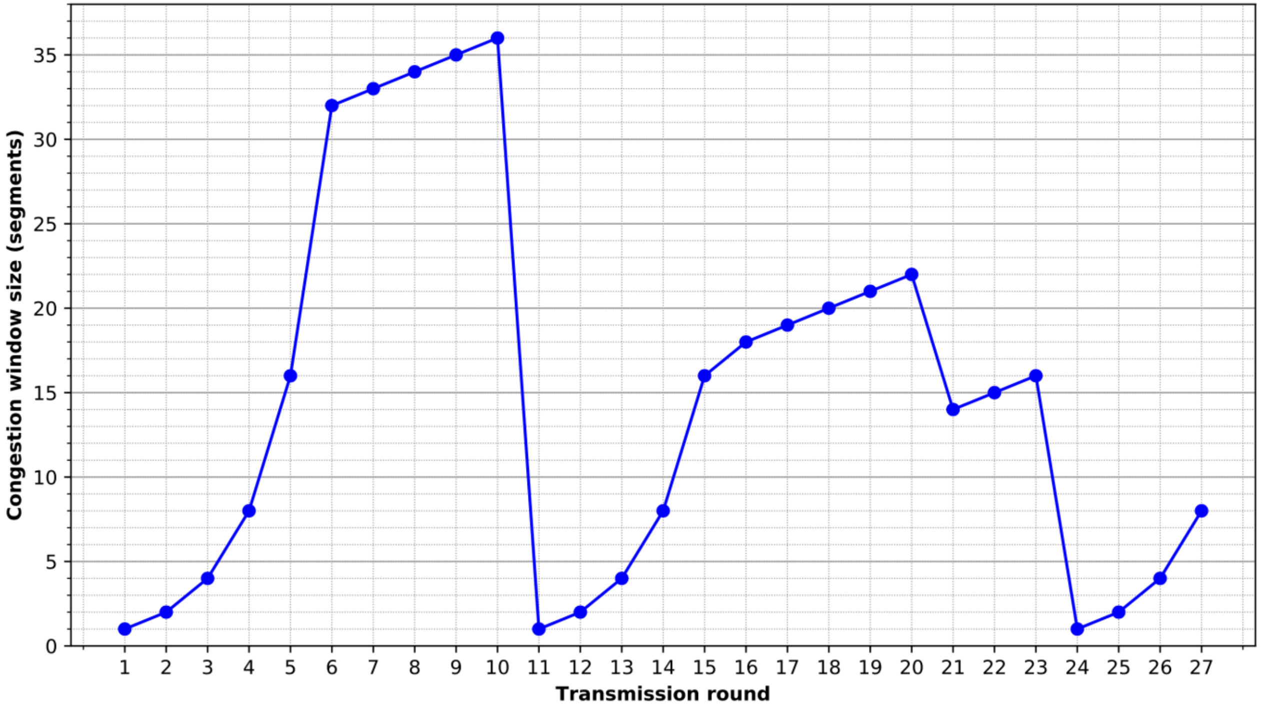
Question 1 (g)

Assuming a packet loss is detected after the 27th round by the reception of a triple duplicate acknowledgement, what will be the values of the congestion window size and Threshold?



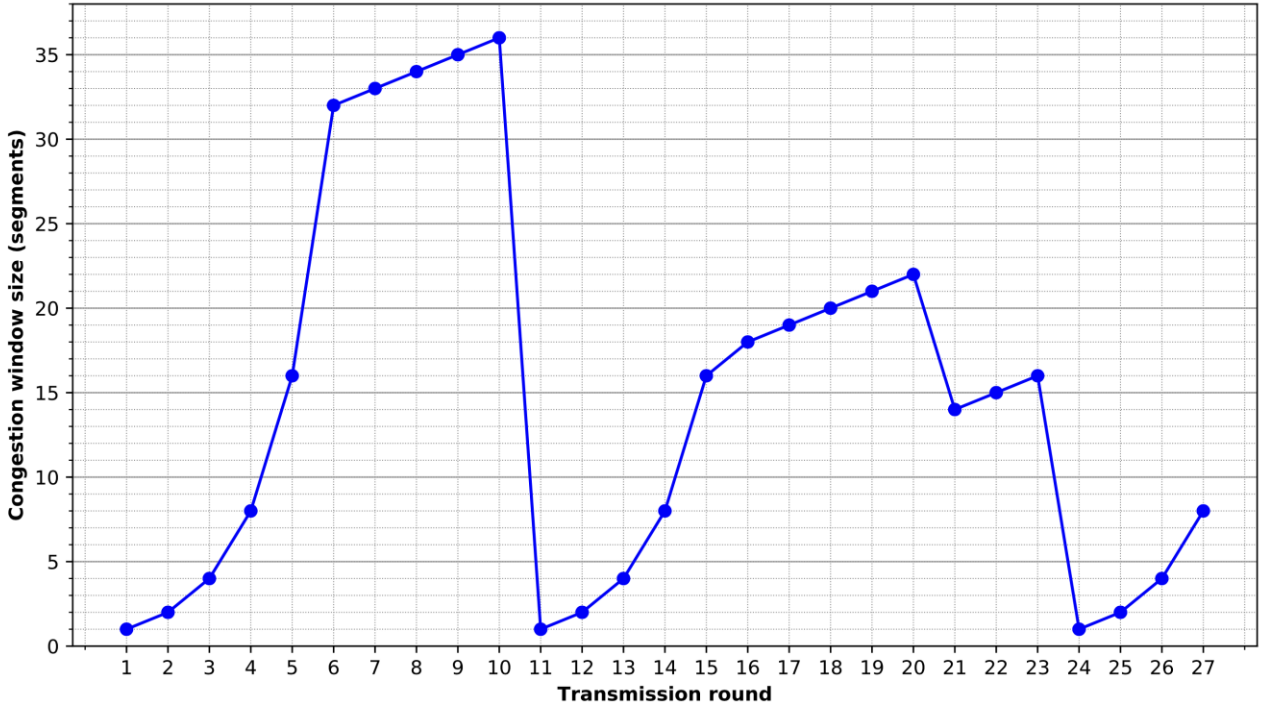
Question 1 (g)

Assuming a packet loss is detected after the 27th round by the reception of a triple duplicate acknowledgement, what will be the values of the congestion window size and Threshold?



Question 1 (g)

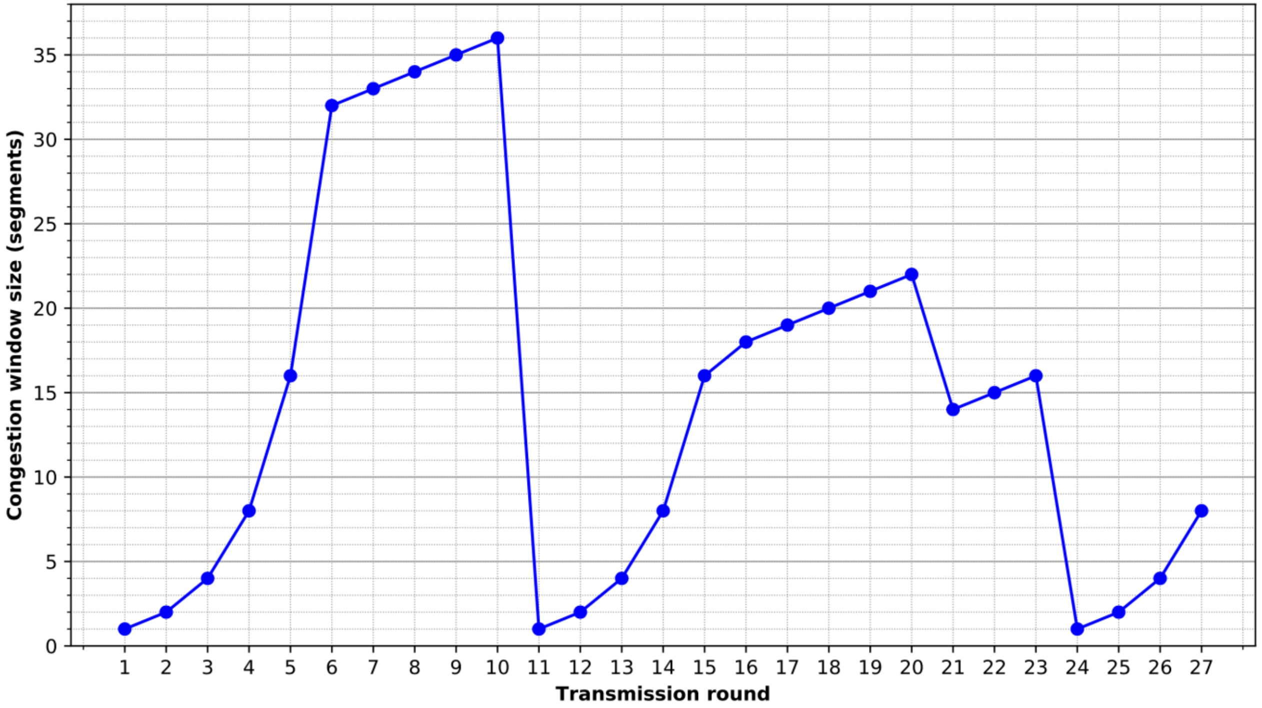
Assuming a packet loss is detected after the 27th round by the reception of a triple duplicate acknowledgement, what will be the values of the congestion window size and Threshold?



For TCP Reno:
ssthresh=cwnd/2;
cwnd=ssthresh+3MSS for fast retransmits,
cwnd=1 for timeouts

Question 1 (g)

Assuming a packet loss is detected after the 27th round by the reception of a triple duplicate acknowledgement, what will be the values of the congestion window size and Threshold?



For TCP Reno:
ssthresh=cwnd/2;
cwnd=ssthresh+3MSS for fast retransmits,
cwnd=1 for timeouts

ssthresh = 8/2=4;
cwnd = sshthresh+3 = 7

Questions?

Question 2 (a)

- TCP BBR, introduced by Google in 2016 is one of the new congestion control algorithms that uses delay as a way of detecting a congested link. During testing, it was shown that BBR was able to achieve lower round trip times compared to New Reno. How does BBR achieve this? The work by Cardwell et al might provide hints to solve this question.

Question 2 (a)

- TCP BBR, introduced by Google in 2016 is one of the new congestion control algorithms that uses delay as a way of detecting a congested link. During testing, it was shown that BBR was able to achieve lower round trip times compared to New Reno. How does BBR achieve this? The work by Cardwell et al might provide hints to solve this question.

Question 2 (a)

- TCP BBR, introduced by Google in 2016 is one of the new congestion control algorithms that uses delay as a way of detecting a congested link. During testing, it was shown that BBR was able to achieve lower round trip times compared to New Reno. How does BBR achieve this? The work by Cardwell et al might provide hints to solve this question.

Question 2 (a)

- BBR periodically estimates the available bandwidth and minimal round-trip time (RTT). It then uses the estimated bandwidth and RTT to estimate BDP. BBR keeps one BDP in flight to minimize delay.

Question 2 (a)

- BBR periodically estimates the available bandwidth and minimal round-trip time (RTT). It then uses the estimated bandwidth and RTT to estimate BDP. BBR keeps one BDP in flight to minimize delay.
- BBR has a "Drain" phase after its "Probe Bandwidth" phase, where it temporarily reduces its sending rate to get rid of the queue created at the end of the "Probe Bandwidth" phase. This prevents the creation of queues, keeping the delay minimal.

Question 2 (b)

- Nowadays, TCP flows usually start with an initial congestion window size larger than one. Explain possible advantages and disadvantages of choosing higher initial congestion window sizes. The work by Dukkipati et al. might provide hints to solve this question.

Question 2 (b)

- Nowadays, TCP flows usually start with an initial congestion window size larger than one. Explain possible advantages and disadvantages of choosing higher initial congestion window sizes. The work by Dukkipati et al. might provide hints to solve this question.

Question 2 (b)

- Nowadays, TCP flows usually start with an initial congestion window size larger than one. Explain possible advantages and disadvantages of choosing higher initial congestion window sizes. The work by Dukkipati et al. might provide hints to solve this question.

Question 2 (b)

Advantages:

Disadvantages:

Question 2 (b)

Advantages:

- Flows complete much faster, i.e flows require less RTTs in slow start phase

Disadvantages:

Question 2 (b)

Advantages:

- Flows complete much faster, i.e flows require less RTTs in slow start phase
- Reduce the need for starting multiple TCP connections

Disadvantages:

Question 2 (b)

Advantages:

- Flows complete much faster, i.e flows require less RTTs in slow start phase
- Reduce the need for starting multiple TCP connections
- Allow fair competition between short and long-lived flows

Disadvantages:

Question 2 (b)

Advantages:

- Flows complete much faster, i.e flows require less RTTs in slow start phase
- Reduce the need for starting multiple TCP connections
- Allow fair competition between short and long-lived flows
- Allow faster recovery from losses

Disadvantages:

Question 2 (b)

Advantages:

- Flows complete much faster, i.e flows require less RTTs in slow start phase
- Reduce the need for starting multiple TCP connections
- Allow fair competition between short and long-lived flows
- Allow faster recovery from losses

Disadvantages:

- May be unfair to flows operating with smaller congestion window settings

Question 2 (b)

Advantages:

- Flows complete much faster, i.e flows require less RTTs in slow start phase
- Reduce the need for starting multiple TCP connections
- Allow fair competition between short and long-lived flows
- Allow faster recovery from losses

Disadvantages:

- May be unfair to flows operating with smaller congestion window settings
- Sending large amounts of data may cause bloated buffers at bottlenecks leading to increased latency

Question 2 (c)

- During TCP Reno's slow start phase the congestion window size is doubled upon successful transmission of a full window. Explain disadvantages and advantages of increasing the multiplier during the slow start phase. The work by Ha et al. might provide hints to solve this question.

Question 2 (c)

- During TCP Reno's slow start phase the congestion window size is doubled upon successful transmission of a full window. Explain disadvantages and advantages of increasing the multiplier during the slow start phase. The work by Ha et al. might provide hints to solve this question.

Question 2 (c)

- During TCP Reno's slow start phase the congestion window size is doubled upon successful transmission of a full window. Explain disadvantages and advantages of increasing the multiplier during the slow start phase. The work by Ha et al. might provide hints to solve this question.

Question 2 (c)

Advantages:

Disadvantages:

Question 2 (c)

Advantages:

- Faster convergence to available link bandwidth solves under-utilisation issue

Disadvantages:

Question 2 (c)

Advantages:

- Faster convergence to available link bandwidth solves under-utilisation issue

Disadvantages:

- Aggressive increases may lead to bursts, bloated buffers (latency) and packet losses

Questions?

Question 3-5: Analyzing Network Traffic

Analyze real traffic using the traffic analysis tool Wireshark .

The simplest functionality of Wireshark are display filters. The display filters restrict the trace presented to the packets fulfilling a specific condition entered by the user. Wireshark also provides a large set of sophisticated automatic analyzers that are generally more powerful and convenient than display filters and useful for various analysis tasks.

The following analyzers will be particularly relevant for us:

- Select a single flow: right click on a packet and select Follow TCP Stream in the context menu
- Plot sequence diagrams: Statistics → Flow Graph → TCP flow → OK
- Plotting functions, like Statistics → I/O-Graphs

Question 3-5: Analyzing Network Traffic

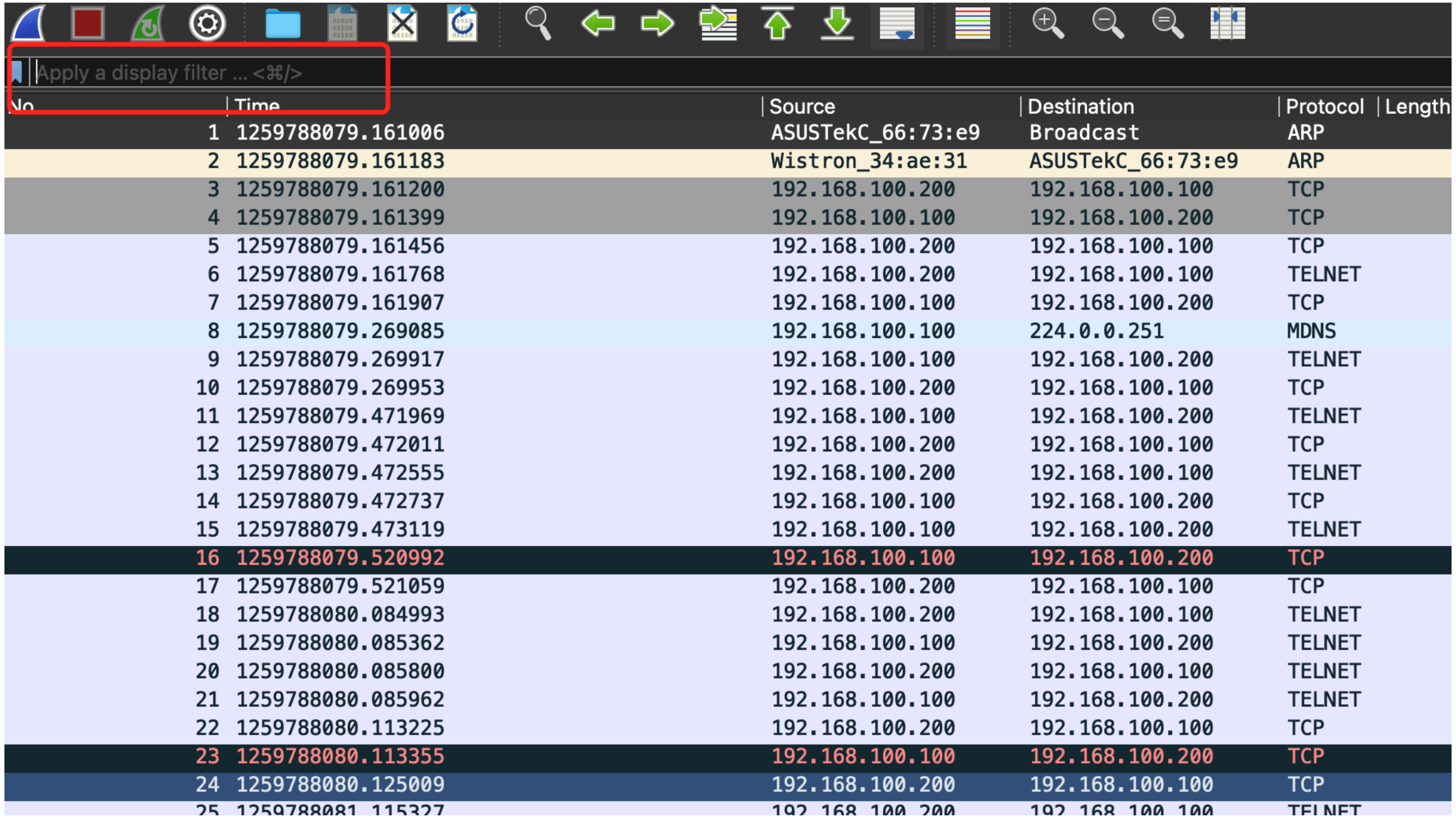
Analyze real traffic using the traffic analysis tool Wireshark .

The simplest functionality of Wireshark are display filters. The display filters restrict the trace presented to the packets fulfilling a specific condition entered by the user. Wireshark also provides a large set of sophisticated automatic analyzers that are generally more powerful and convenient than display filters and useful for various analysis tasks.

The following analyzers will be particularly relevant for us:

- Select a single flow: right click on a packet and select Follow TCP Stream in the context menu
- Plot sequence diagrams: Statistics → Flow Graph → TCP flow → OK
- Plotting functions, like Statistics → I/O-Graphs

Question 3-5: Analyzing Network Traffic



No.	Time	Source	Destination	Protocol	Length
1	1259788079.161006	ASUSTekC_66:73:e9	Broadcast	ARP	
2	1259788079.161183	Wistron_34:ae:31	ASUSTekC_66:73:e9	ARP	
3	1259788079.161200	192.168.100.200	192.168.100.100	TCP	
4	1259788079.161399	192.168.100.100	192.168.100.200	TCP	
5	1259788079.161456	192.168.100.200	192.168.100.100	TCP	
6	1259788079.161768	192.168.100.200	192.168.100.100	TELNET	
7	1259788079.161907	192.168.100.100	192.168.100.200	TCP	
8	1259788079.269085	192.168.100.100	224.0.0.251	MDNS	
9	1259788079.269917	192.168.100.100	192.168.100.200	TELNET	
10	1259788079.269953	192.168.100.200	192.168.100.100	TCP	
11	1259788079.471969	192.168.100.100	192.168.100.200	TELNET	
12	1259788079.472011	192.168.100.200	192.168.100.100	TCP	
13	1259788079.472555	192.168.100.200	192.168.100.100	TELNET	
14	1259788079.472737	192.168.100.100	192.168.100.200	TCP	
15	1259788079.473119	192.168.100.100	192.168.100.200	TELNET	
16	1259788079.520992	192.168.100.100	192.168.100.200	TCP	
17	1259788079.521059	192.168.100.200	192.168.100.100	TCP	
18	1259788080.084993	192.168.100.200	192.168.100.100	TELNET	
19	1259788080.085362	192.168.100.100	192.168.100.200	TELNET	
20	1259788080.085800	192.168.100.200	192.168.100.100	TELNET	
21	1259788080.085962	192.168.100.100	192.168.100.200	TELNET	
22	1259788080.113225	192.168.100.200	192.168.100.100	TCP	
23	1259788080.113355	192.168.100.100	192.168.100.200	TCP	
24	1259788080.125009	192.168.100.200	192.168.100.100	TCP	
25	1259788081.115327	192.168.100.200	192.168.100.100	TELNET	

Question 3-5: Analyzing Network Traffic

Analyze real traffic using the traffic analysis tool Wireshark .

The simplest functionality of Wireshark are display filters. The display filters restrict the trace presented to the packets fulfilling a specific condition entered by the user. Wireshark also provides a large set of sophisticated automatic analyzers that are generally more powerful and convenient than display filters and useful for various analysis tasks.

The following analyzers will be particularly relevant for us:

- Select a single flow: right click on a packet and select Follow TCP Stream in the context menu
- Plot sequence diagrams: Statistics → Flow Graph → TCP flow → OK
- Plotting functions, like Statistics → I/O-Graphs

Question 3-5: Analyzing Network Traffic

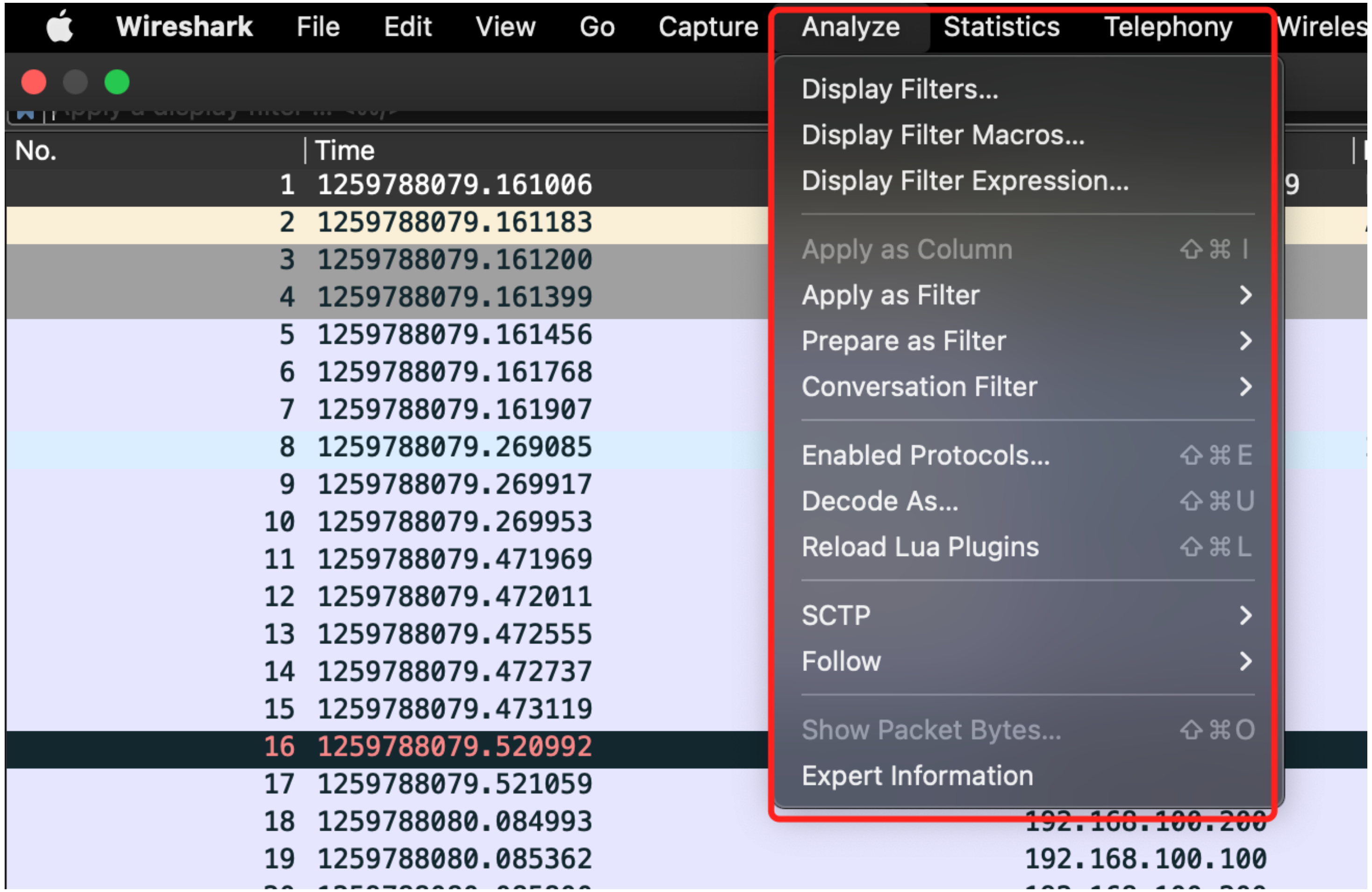
Analyze real traffic using the traffic analysis tool Wireshark .

The simplest functionality of Wireshark are display filters. The display filters restrict the trace presented to the packets fulfilling a specific condition entered by the user. Wireshark also provides a large set of sophisticated automatic analyzers that are generally more powerful and convenient than display filters and useful for various analysis tasks.

The following analyzers will be particularly relevant for us:

- Select a single flow: right click on a packet and select Follow TCP Stream in the context menu
- Plot sequence diagrams: Statistics → Flow Graph → TCP flow → OK
- Plotting functions, like Statistics → I/O-Graphs

Question 3-5: Analyzing Network Traffic



Question 3-5: Analyzing Network Traffic

Analyze real traffic using the traffic analysis tool Wireshark .

The simplest functionality of Wireshark are display filters. The display filters restrict the trace presented to the packets fulfilling a specific condition entered by the user. Wireshark also provides a large set of sophisticated automatic analyzers that are generally more powerful and convenient than display filters and useful for various analysis tasks.

The following analyzers will be particularly relevant for us:

- Select a single flow: right click on a packet and select Follow TCP Stream in the context menu
- Plot sequence diagrams: Statistics → Flow Graph → TCP flow → OK
- Plotting functions, like Statistics → I/O-Graphs

Question 3-5: Analyzing Network Traffic

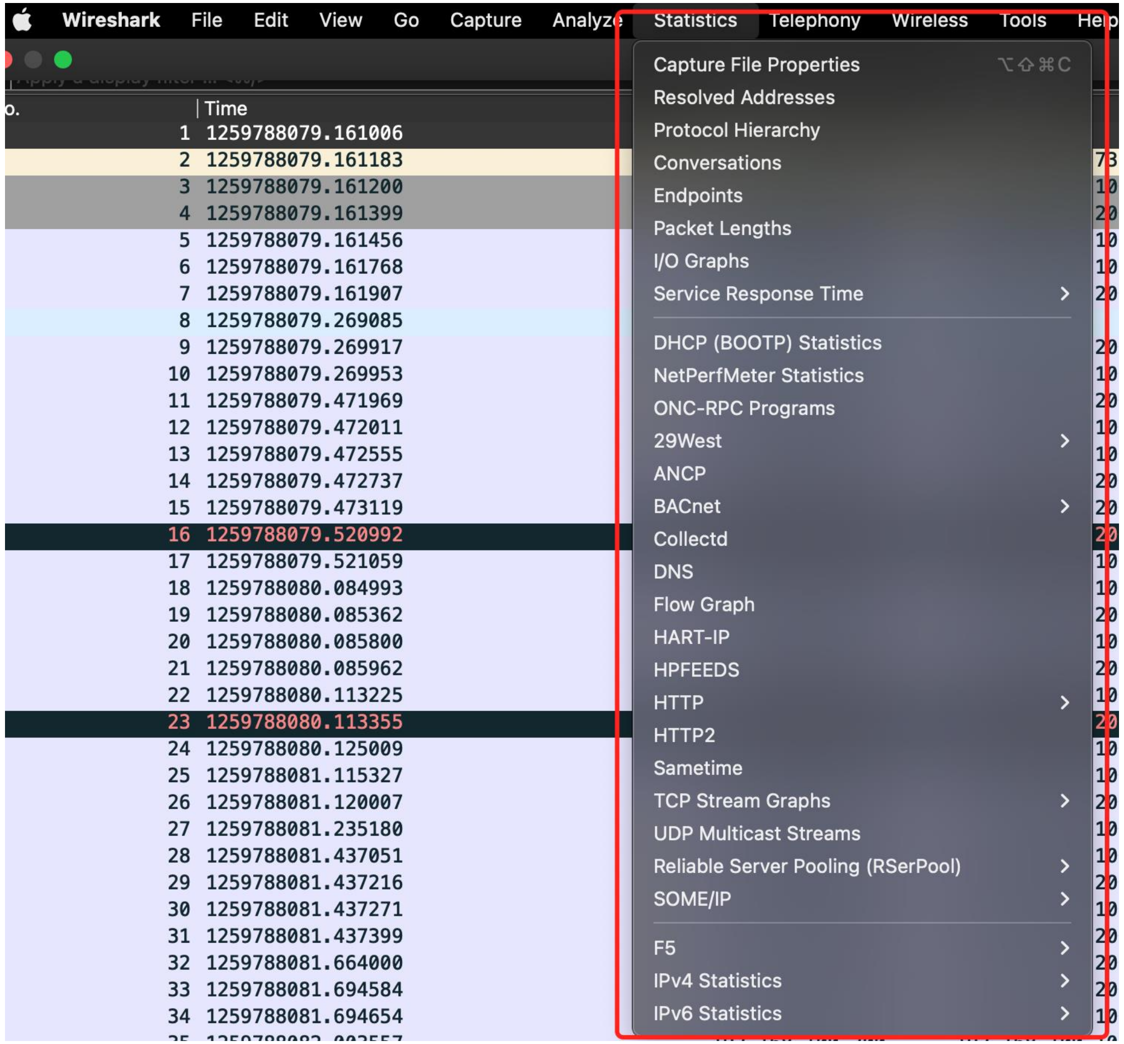
Analyze real traffic using the traffic analysis tool Wireshark .

The simplest functionality of Wireshark are display filters. The display filters restrict the trace presented to the packets fulfilling a specific condition entered by the user. Wireshark also provides a large set of sophisticated automatic analyzers that are generally more powerful and convenient than display filters and useful for various analysis tasks.

The following analyzers will be particularly relevant for us:

- Select a single flow: right click on a packet and select Follow TCP Stream in the context menu
- Plot sequence diagrams: Statistics → Flow Graph → TCP flow → OK
- Plotting functions, like Statistics → I/O-Graphs

Question 3-5: Analyzing Network Traffic



Question 3-5: Analyzing Network Traffic

Analyze real traffic using the traffic analysis tool Wireshark .

The simplest functionality of Wireshark are display filters. The display filters restrict the trace presented to the packets fulfilling a specific condition entered by the user. Wireshark also provides a large set of sophisticated automatic analyzers that are generally more powerful and convenient than display filters and useful for various analysis tasks.

The following analyzers will be particularly relevant for us:

- Select a single flow: right click on a packet and select Follow TCP Stream in the context menu
- Plot sequence diagrams: Statistics → Flow Graph → TCP flow → OK
- Plotting functions, like Statistics → I/O-Graphs

Question 3-5: Analyzing Network Traffic

Familiarize yourself with the tool and try out different statistics and tools on the trace file we provide below.

In the following questions we ask you to do similar tasks by using display filters as well as the automatic analyzers in order to familiarize yourself with both techniques.

We will often refer to the Stream Index of a TCP connection. Keep in mind that this identifier can be obtained by the Follow TCP stream function.

Question 3-5: Analyzing Network Traffic

Familiarize yourself with the tool and try out different statistics and tools on the trace file we provide below.

In the following questions we ask you to do similar tasks by using display filters as well as the automatic analyzers in order to familiarize yourself with both techniques.

We will often refer to the Stream Index of a TCP connection. Keep in mind that this identifier can be obtained by the Follow TCP stream function.

Question 3-5: Analyzing Network Traffic

Familiarize yourself with the tool and try out different statistics and tools on the trace file we provide below.

In the following questions we ask you to do similar tasks by using display filters as well as the automatic analyzers in order to familiarize yourself with both techniques.

We will often refer to the Stream Index of a TCP connection. Keep in mind that this identifier can be obtained by the Follow TCP stream function.

Question 3-5: Analyzing Network Traffic

Important note: The journey is the reward; just stating the solution to the questions posed below is not a sufficient answer but you should include a description of your reasoning and how the results were obtained — for instance, when you use display filters for a question, copy them into your answer of the question.

Question 3-5: Analyzing Network Traffic

Important note: The journey is the reward; just stating the solution to the questions posed below is not a sufficient answer but you should include a description of your reasoning and how the results were obtained — for instance, when you use display filters for a question, copy them into your answer of the question.

Question 3 (a): TCP connections

How many TCP connections are at least in part contained in the trace?

Question 3 (a): TCP connections

How many TCP connections are at least in part contained in the trace?

Question 3 (a): TCP connections

How many TCP connections are at least in part contained in the trace?

Answer: 9 connections.

Question 3 (a): TCP connections

How many TCP connections are at least in part contained in the trace?

Answer: 9 connections.

Steps" Statistics -> Conversions -> Select TCP

TCP · 9

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
130.149.220.42	22	130.149.220.164	47191	25	2,404 KiB	2	16	1,543 KiB	9	882 bytes	19.131856	280.6124	45 bytes	25 bytes
130.149.220.42	39050	130.149.220.164	22	899	170,479 KiB	5	382	28,516 KiB	517	141,963 KiB	112.522806	175.5260	1,299 KiB	6,470 KiB
130.149.220.164	40817	130.149.220.42	22	470	172,242 KiB	6	256	23,027 KiB	214	149,215 KiB	122.225324	159.1251	1,157 KiB	7,501 KiB
130.149.220.164	49241	130.149.220.251	80	35	21,847 KiB	1	17	1,215 KiB	18	20,632 KiB	9.616225	0.1760	55,221 KiB	937,826 KiB
130.149.220.164	49243	130.149.220.251	80	33	21,729 KiB	4	15	1,098 KiB	18	20,632 KiB	70.611999	0.1081	81,244 KiB	1,491 MiB
130.149.220.164	52142	130.149.220.251	80	12	1,968 KiB	8	6	642 bytes	6	1,341 KiB	278.624010	2.6325	1,905 KiB	4,074 KiB
130.149.220.164	47001	130.149.220.252	25	32	2,623 KiB	3	17	1,412 KiB	15	1,211 KiB	49.666050	263.8858	43 bytes	37 bytes
192.168.100.200	42700	192.168.100.100	22	724	585,995 KiB	7	306	23,996 KiB	418	561,999 KiB	133.169129	80.5474	2,383 KiB	55,817 KiB
192.168.100.200	59142	192.168.100.100	23	199	14,910 KiB	0	116	7,794 KiB	83	7,116 KiB	0.000194	295.8402	215 bytes	197 bytes

Question 3 (b): TCP connections

Using display filters, fill Table 1 below for the first TCP connection starting in the trace. Briefly explain your approach! Hint: Filter by TCP flags and then identify the first connection.

Stream Index	source IP	destination IP	conn. start	conn. end	display filter

Table 1: Single-Entry Connection Table

Question 3 (b): TCP connections

Using display filters, fill Table 1 below for the first TCP connection starting in the trace. Briefly explain your approach! Hint: Filter by TCP flags and then identify the first connection.

Stream Index	source IP	destination IP	conn. start	conn. end	display filter

Table 1: Single-Entry Connection Table

Question 3 (b): TCP connections

Use filter: (tcp.stream eq 0)

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000194	192.168.100.200	192.168.100.100	TCP	74	59142 → 23 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM TSval=608939464 TSecr=0 WS=128
4	0.000393	192.168.100.100	192.168.100.200	TCP	74	23 → 59142 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=6919129 TSecr=608939464 WS=512
5	0.000450	192.168.100.200	192.168.100.100	TCP	66	59142 → 23 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=608939465 TSecr=6919129
6	0.000762	192.168.100.200	192.168.100.100	TELNET	93	Telnet Data

2548	295.549763	192.168.100.200	192.168.100.100	TELNET	68	Telnet Data ...
2549	295.811507	192.168.100.100	192.168.100.200	TCP	66	23 → 59142 [FIN, ACK] Seq=1347 Ack=220 Win=6144 Len=0 TSval=6993017 TSecr=609013352
2550	295.811731	192.168.100.100	192.168.100.200	TCP	66	[TCP Retransmission] 23 → 59142 [FIN, ACK] Seq=1347 Ack=220 Win=6144 Len=0 TSval=6993081 TSecr=609013352
2551	295.811758	192.168.100.200	192.168.100.100	TCP	78	[TCP Previous segment not captured] 59142 → 23 [ACK] Seq=221 Ack=1348 Win=8064 Len=0 TSval=609013417 TSecr=6993081 SLE=1347 SRE=1348
2552	295.840216	192.168.100.200	192.168.100.100	TCP	66	[TCP Retransmission] 59142 → 23 [FIN, ACK] Seq=220 Ack=1348 Win=8064 Len=0 TSval=609013417 TSecr=6993017
2553	295.840417	192.168.100.100	192.168.100.200	TCP	66	23 → 59142 [ACK] Seq=1348 Ack=221 Win=6144 Len=0 TSval=6993088 TSecr=609013417

Stream Index	source IP	destination IP	conn. start	conn. end
0	192.168.100.200	192.168.100.100	0.000194	295.840417

Question 3 (c): TCP connections

Using automatic analyzers, fill Table 2 below for all TCP connections in the trace (one row per connection). Sort the connections in increasing order of Stream Index. Additionally, specify the analyzers used, how they are used and explain your approach.

Stream Index	source IP	destination IP	conn. start	conn. end

Table 2: Full Connection Table

Question 3 (c): TCP connections

Using automatic analyzers, fill Table 2 below for all TCP connections in the trace (one row per connection). Sort the connections in increasing order of Stream Index. Additionally, specify the analyzers used, how they are used and explain your approach.

Stream Index	source IP	destination IP	conn. start	conn. end

Table 2: Full Connection Table

Question 3 (c): TCP connections

See Conversations Filter → TCP for the information needed to fill in the table — end can be calculated as start + duration

Question 3 (c): TCP connections

Stream Index	source IP	destination IP	conn. start	conn. end
0	192.168.100.200	192.168.100.100	0.000194	295.840417
1	130.149.220.164	130.149.220.251	9.616225	9.792222
2	130.149.220.42	130.149.220.164	19.131856	299.744244
3	130.149.220.164	130.149.220.252	49.66605	313.551888
4	130.149.220.164	130.149.220.251	70.611999	70.720083
5	130.149.220.42	130.149.220.164	112.522806	288.048851
6	130.149.220.164	130.149.220.42	122.225324	281.350445
7	192.168.100.200	192.168.100.100	133.169129	213.716505
8	130.149.220.164	130.149.220.251	278.62401	281.256498

Question 3 (d): TCP connections

How many UDP flows are there? Briefly explain how you found this information.

Question 3 (d): TCP connections

How many UDP flows are there? Briefly explain how you found this information.

Question 3 (d): TCP connections

How many UDP flows are there? Briefly explain how you found this information.

Answer: 68 flows

Steps: Statistics -> Conversations-> Select UDP

UDP · 68

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
130.149.220.164	32956	130.149.220.253	53	2	226 bytes	2	1	88 bytes	1	138 bytes	49.664915	0.0007		
130.149.220.164	34364	130.149.220.253	53	2	232 bytes	5	1	91 bytes	1	141 bytes	112.852512	0.0004		
130.149.220.164	34879	130.149.220.253	53	2	234 bytes	44	1	92 bytes	1	142 bytes	113.834931	0.0004		
130.149.220.164	35045	130.149.220.253	53	2	253 bytes	4	1	87 bytes	1	166 bytes	112.851200	0.0008		
130.149.220.164	35487	130.149.220.253	53	2	224 bytes	1	1	87 bytes	1	137 bytes	9.589148	0.0008		
130.149.220.164	35830	130.149.220.253	53	2	312 bytes	37	1	105 bytes	1	207 bytes	113.830009	0.0005		
130.149.220.164	36265	130.149.220.253	53	2	368 bytes	55	1	98 bytes	1	270 bytes	122.860380	0.0006		
130.149.220.164	36487	130.149.220.253	53	2	234 bytes	22	1	92 bytes	1	142 bytes	113.799424	0.0005		
130.149.220.164	37203	130.149.220.253	53	2	234 bytes	28	1	92 bytes	1	142 bytes	113.803666	0.0005		
130.149.220.164	37301	130.149.220.253	53	2	224 bytes	67	1	87 bytes	1	137 bytes	281.085223	0.0007		

Question 3 (e): TCP connections

Give an example of a TCP connection exhibiting packet loss, specified by its Stream Index.

Question 3 (e): TCP connections

Give an example of a TCP connection exhibiting packet loss, specified by its Stream Index.

Question 3 (e): TCP connections

Give an example of a TCP connection exhibiting packet loss, specified by its Stream Index.

Answer: Stream Index 5; No. 226–228 between 130.149.220.164 and 130.149.220.42

With display filter : tcp.analysis.lost_segment

Question 4 (a): DNS Resolution

Manually obtain the DNS name of a single host, specified by its IP address, using only information contained in the trace. Explain your approach. Hint: Look at the DNS traffic.

Question 4 (a): DNS Resolution

Manually obtain the DNS name of a single host, specified by its IP address, using only information contained in the trace. Explain your approach. Hint: Look at the DNS traffic.

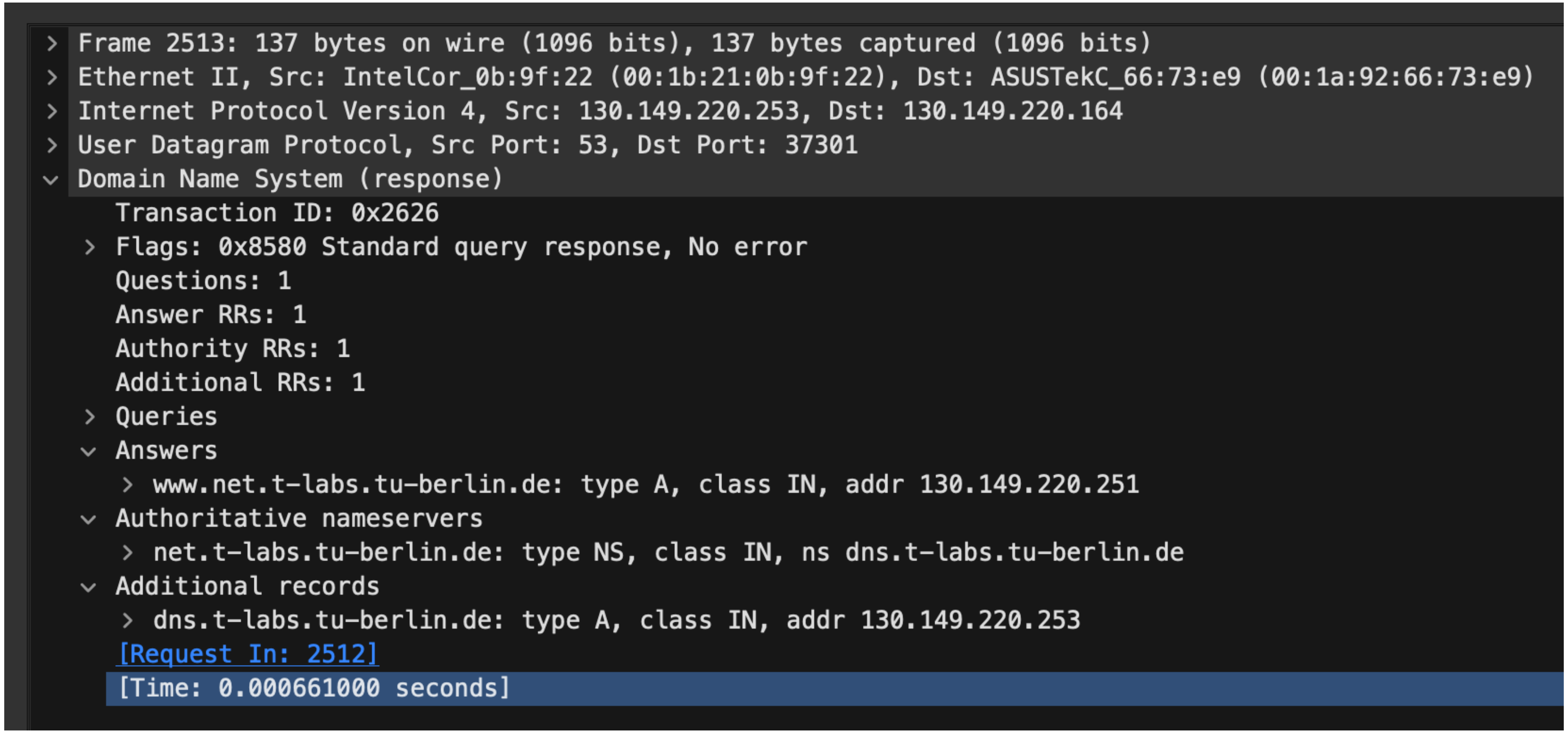
Question 4 (a): DNS Resolution

Manually obtain the DNS name of a single host, specified by its IP address, using only information contained in the trace. Explain your approach. Hint: Look at the DNS traffic.

Packet No. 2513, double click, scroll down to Domain Name System (response),

Answers: www.net.t-labs.tu-berlin.de: type A, class IN, addr 130.149.220.251

Question 4 (a): DNS Resolution



Question 4 (b): DNS Resolution

Now use automatic analyzers of Wireshark to resolve the names of all hosts (including the previous one). Present your results in Table 3 below. Write a hyphen – if the host does not have a DNS name.

host IP	DNS name

Table 3: DNS Translation Table

Question 4 (b): DNS Resolution

Now use automatic analyzers of Wireshark to resolve the names of all hosts (including the previous one). Present your results in Table 3 below. Write a hyphen – if the host does not have a DNS name.

host IP	DNS name

Table 3: DNS Translation Table

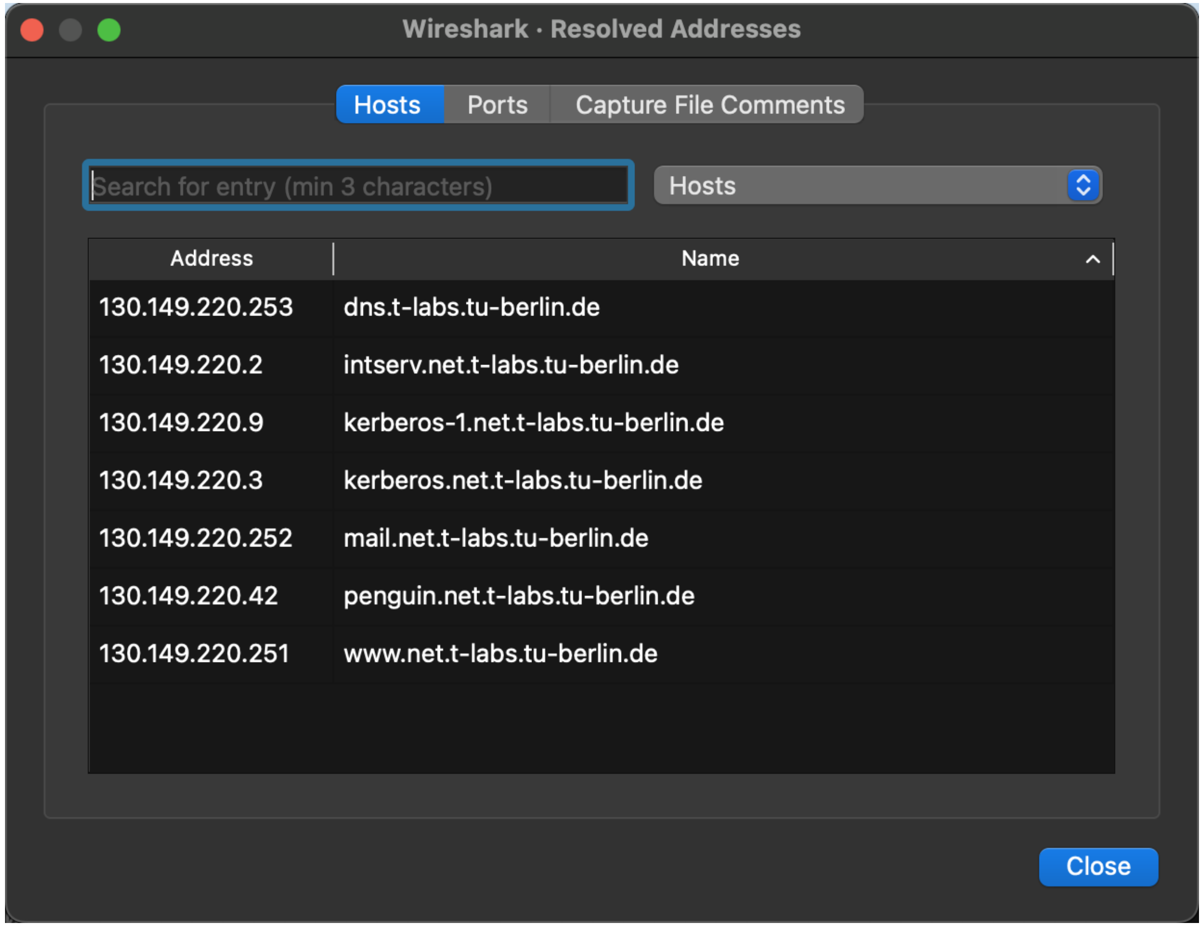
Question 4 (b): DNS Resolution

Statistics → Resolved Addresses

host IP	DNS name
130.149.220.9	kerberos-1.net.t-labs.tu-berlin.de
130.149.220.2	intserv.net.t-labs.tu-berlin.de
130.149.220.251	www.net.t-labs.tu-berlin.de
130.149.220.42	penguin.net.t-labs.tu-berlin.de
130.149.220.3	kerberos.net.t-labs.tu-berlin.de
130.149.220.252	mail.net.t-labs.tu-berlin.de
130.149.220.253	dns.t-labs.tu-berlin.de

Question 4 (b): DNS Resolution

Statistics -> Resolved Addresses -> Select Hosts



Questions?

Question 5 (a): Application Layer

Sorting the connections in increasing order by Stream Index, answer in 2-3 sentences per connection the following questions:

(i) What is the user doing / what is requested?

(ii) Which information is disclosed (passwords, etc.)?

If you cannot find this information, justify why it is not possible. When private information is disclosed, what would be an alternative application layer protocol fulfilling the same functionality but without information disclosure.

Question 5 (a): Application Layer

Sorting the connections in increasing order by Stream Index, answer in 2-3 sentences per connection the following questions:

(i) What is the user doing / what is requested?

(ii) Which information is disclosed (passwords, etc.)?

If you cannot find this information, justify why it is not possible. When private information is disclosed, what would be an alternative application layer protocol fulfilling the same functionality but without information disclosure.

Question 5 (a): Application Layer

tcp.stream eq 3

No.	Time	Source	Destination	Protocol	Length	Info
128	49.666050	130.149.220.164	130.149.220.252	TCP	74	47001 → 25 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM TSval=608951881 TSecr=0 WS=128
129	49.666643	130.149.220.252	130.149.220.164	TCP	74	25 → 47001 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=1009706860 TSecr=608951881 WS=128
130	49.666686	130.149.220.164	130.149.220.252	TCP	66	47001 → 25 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=608951881 TSecr=1009706860
131	49.668640	130.149.220.252	130.149.220.164	SMTP	127	S: 220 mail.net.t-labs.tu-berlin.de ESMTP Postfix (Debian/GNU)
132	49.668669	130.149.220.164	130.149.220.252	TCP	66	47001 → 25 [ACK] Seq=1 Ack=62 Win=5888 Len=0 TSval=608951882 TSecr=1009706861
133	59.027557	130.149.220.164	130.149.220.252	SMTP	101	C: HELO mail.net.t-labs.tu-berlin.de
134	59.028060	130.149.220.252	130.149.220.164	TCP	66	25 → 47001 [ACK] Seq=62 Ack=36 Win=5888 Len=0 TSval=1009709201 TSecr=608954221
135	59.028067	130.149.220.252	130.149.220.164	SMTP	100	S: 250 mail.net.t-labs.tu-berlin.de
136	59.028098	130.149.220.164	130.149.220.252	TCP	66	47001 → 25 [ACK] Seq=36 Ack=96 Win=5888 Len=0 TSval=608954222 TSecr=1009709201
141	60.155318	130.149.220.164	130.149.220.252	SMTP	112	C: MAIL FROM: chewbacca@net.t-labs.tu-berlin.de
142	60.156389	130.149.220.252	130.149.220.164	SMTP	80	S: 250 2.1.0 Ok
143	60.156418	130.149.220.164	130.149.220.252	TCP	66	47001 → 25 [ACK] Seq=82 Ack=110 Win=5888 Len=0 TSval=608954504 TSecr=1009709483
179	80.369228	130.149.220.164	130.149.220.252	SMTP	104	C: RCPT TO: jan@net.t-labs.tu-berlin.de
180	80.400328	130.149.220.252	130.149.220.164	TCP	66	25 → 47001 [ACK] Seq=110 Ack=120 Win=5888 Len=0 TSval=1009714544 TSecr=608959557
181	80.400425	130.149.220.164	130.149.220.252	SMTP	137	C: DATA fragment, 71 bytes
182	80.400939	130.149.220.252	130.149.220.164	TCP	66	25 → 47001 [ACK] Seq=110 Ack=191 Win=5888 Len=0 TSval=1009714544 TSecr=608959565
183	80.433181	130.149.220.252	130.149.220.164	SMTP	80	S: 250 2.1.5 Ok
184	80.433243	130.149.220.164	130.149.220.252	TCP	66	47001 → 25 [ACK] Seq=191 Ack=124 Win=5888 Len=0 TSval=608959573 TSecr=1009714551
185	80.433794	130.149.220.252	130.149.220.164	SMTP	103	S: 354 End data with <CR><LF>.<CR><LF>
186	80.433810	130.149.220.164	130.149.220.252	TCP	66	47001 → 25 [ACK] Seq=191 Ack=161 Win=5888 Len=0 TSval=608959573 TSecr=1009714551
2557	306.548054	130.149.220.164	130.149.220.252	SMTP	89	C: DATA fragment, 23 bytes
2558	306.586931	130.149.220.252	130.149.220.164	TCP	66	25 → 47001 [ACK] Seq=161 Ack=214 Win=5888 Len=0 TSval=1009771095 TSecr=609016102
2559	306.587016	130.149.220.164	130.149.220.252	SMTP/IMF	163	subject: Invasion 2.0, , Will support you. Give orders, we follow. , , greetings to The Emperor too! , best. Angie
2560	306.587416	130.149.220.252	130.149.220.164	TCP	66	25 → 47001 [ACK] Seq=161 Ack=311 Win=5888 Len=0 TSval=1009771095 TSecr=609016111
2561	306.595790	130.149.220.252	130.149.220.164	SMTP	133	S: 250 2.0.0 Ok: queued as 91753700D2A9 | 500 5.5.2 Error: bad syntax
2562	306.595821	130.149.220.164	130.149.220.252	TCP	66	47001 → 25 [ACK] Seq=311 Ack=228 Win=5888 Len=0 TSval=609016113 TSecr=1009771095
2563	313.550092	130.149.220.164	130.149.220.252	SMTP	72	C: quit
2564	313.550761	130.149.220.252	130.149.220.164	SMTP	81	S: 221 2.0.0 Bye
2565	313.550811	130.149.220.164	130.149.220.252	TCP	66	47001 → 25 [ACK] Seq=317 Ack=243 Win=5888 Len=0 TSval=609017852 TSecr=1009772835
2566	313.551011	130.149.220.252	130.149.220.164	TCP	66	25 → 47001 [FIN, ACK] Seq=243 Ack=317 Win=5888 Len=0 TSval=1009772835 TSecr=609017852
2567	313.551253	130.149.220.164	130.149.220.252	TCP	66	47001 → 25 [FIN, ACK] Seq=317 Ack=244 Win=5888 Len=0 TSval=609017852 TSecr=1009772835
2568	313.551888	130.149.220.252	130.149.220.164	TCP	66	25 → 47001 [ACK] Seq=244 Ack=318 Win=5888 Len=0 TSval=1009772835 TSecr=609017852

Question 5 (a): Application Layer

- Stream 0: Remote Login on a Linux Machine with xterm; everything revealed, including remote system status and user credentials (SSH!)
- Stream 1: Regular HTTP; content retrieved is visible (HTTPS!)
- Stream 2: Traffic encrypted, therefore no information available
- Stream 3: Email, addresses and content visible (TLS!)
- Stream 4: Again, HTTP, see Stream 1
- Stream 5: SSH, traffic encrypted, therefore no information available
- Stream 6: Again, SSH, see Stream 5
- Stream 7: Again, SSH, see Stream 5
- Stream 8: Again, HTTP, see Stream 1

Question 5 (b): Application Layer

Take a look at packets 18 to 20. What is in your opinion the application layer semantic of the three packets? Additionally, name the IETF standards document in which the semantic of these packets is defined. How did you find it?

Question 5 (b): Application Layer

Take a look at packets 18 to 20. What is in your opinion the application layer semantic of the three packets? Additionally, name the IETF standards document in which the semantic of these packets is defined. How did you find it?

Question 5 (b): Application Layer

Take a look at packets 18 to 20. What is in your opinion the application layer semantic of the three packets? Additionally, name the IETF standards document in which the semantic of these packets is defined. How did you find it?

12	0.311005	192.168.100.200	192.168.100.100	TCP	66	59142 → 23 [ACK] Seq=28 Ack=52 Win=5888 Len=0 TSval=608939542 TSecr=691
13	0.311549	192.168.100.200	192.168.100.100	TELNET	188	Telnet Data ...
14	0.311731	192.168.100.100	192.168.100.200	TCP	66	23 → 59142 [ACK] Seq=52 Ack=150 Win=6144 Len=0 TSval=6919207 TSecr=6089
15	0.312113	192.168.100.100	192.168.100.200	TELNET	69	Telnet Data ...
16	0.359986	192.168.100.100	192.168.100.200	TCP	105	[TCP Spurious Retransmission] 23 → 59142 [PSH, ACK] Seq=13 Ack=28 Win=6
17	0.360053	192.168.100.200	192.168.100.100	TCP	78	59142 → 23 [ACK] Seq=150 Ack=55 Win=5888 Len=0 TSval=608939555 TSecr=69
18	0.923987	192.168.100.200	192.168.100.100	TELNET	69	Telnet Data ...
19	0.924356	192.168.100.100	192.168.100.200	TELNET	69	Telnet Data ...
20	0.924794	192.168.100.200	192.168.100.100	TELNET	69	Telnet Data ...
21	0.924956	192.168.100.100	192.168.100.200	TELNET	93	Telnet Data ...
22	0.952219	192.168.100.200	192.168.100.100	TCP	66	59142 → 23 [ACK] Seq=153 Ack=58 Win=5888 Len=0 TSval=608939696 TSecr=69
23	0.952349	192.168.100.100	192.168.100.200	TCP	66	[TCP Dup ACK 21#1] 23 → 59142 [ACK] Seq=85 Ack=156 Win=6144 Len=0 TSval=
24	0.964003	192.168.100.200	192.168.100.100	TCP	66	59142 → 23 [ACK] Seq=156 Ack=85 Win=5888 Len=0 TSval=608939706 TSecr=69

Question 5 (b): Application Layer

Take a look at packets 18 to 20. What is in your opinion the application layer semantic of the three packets? Additionally, name the IETF standards document in which the semantic of these packets is defined. How did you find it?

Application Layer Information for the three packets, already interpreted by Wireshark

```
⌄ Telnet
  ⌄ Won't Echo
      Command: Won't (252)
      Subcommand: Echo
```

18th packet

```
⌄ Telnet
  ⌄ Will Echo
      Command: Will (251)
      Subcommand: Echo
```

19th packet

```
⌄ Telnet
  ⌄ Do Echo
      Command: Do (253)
      Subcommand: Echo
```

20th packet

Question 5 (b): Application Layer

Take a look at packets 18 to 20. What is in your opinion the application layer semantic of the three packets? Additionally, name the IETF standards document in which the semantic of these packets is defined. How did you find it?

Telnet

RFC 854

(Google → tools.ietf.org)

Questions?

Feedback?