23.7 SUMMARY

- In the client/server paradigm, an application program on the local host, called the client, needs services from an application program on the remote host, called a server.

- Each application program has a port number that distinguishes it from other programs running at the same time on the same machine.
- The client program is assigned a random port number called an ephemeral port number; the server program is assigned a universal port number called a well-known port number.
- The ICANN has specified ranges for the different types of port numbers.
- The combination of the IP address and the port number, called the socket address, defines a process and a host.
- UDP is a connectionless, unreliable transport layer protocol with no embedded flow or error control mechanism except the checksum for error detection.
- The UDP packet is called a user datagram. A user datagram is encapsulated in the data field of an IP datagram.
- Transmission Control Protocol (TCP) is one of the transport layer protocols in the TCP/IP protocol suite.
- TCP provides process-to-process, full-duplex, and connection-oriented service.
- The unit of data transfer between two devices using TCP software is called a segment; it has 20 to 60 bytes of header, followed by data from the application program.
- A TCP connection normally consists of three phases: connection establishment, data transfer, and connection termination.
- Connection establishment requires three-way handshaking; connection termination requires three- or four-way handshaking.
- TCP uses flow control, implemented as a sliding window mechanism, to avoid overwhelming a receiver with data.

- The TCP window size is determined by the receiver-advertised window size *(rwnd)* or the congestion window size *(cwnd),* whichever is smaller. The window can be opened or closed by the receiver, but should not be shrunk.
- The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with a randomly generated number.
- TCP uses error control to provide a reliable service. Error control is handled by the checksum, acknowledgment, and time-out. Corrupted and lost segments are retransmitted, and duplicate segments are discarded. Data may arrive out of order and are temporarily stored by the receiving TCP, but TCP guarantees that no out-of-order segment is delivered to the process.
- In modem implementations, a retransmission occurs if the retransmission timer expires or three duplicate ACK segments have arrived.
- SCTP is a message-oriented, reliable protocol that combines the good features of UDP and TCP.
- SCTP provides additional services not provided by UDP or Tep, such as multiple-stream and multihoming services.
- SCTP is a connection-oriented protocol. An SCTP connection is called an association.
- SCTP uses the term *packet* to define a transportation unit.
- In SCTP, control information and data information are carried in separate chunks.

- An SCTP packet can contain control chunks and data chunks with control chunks coming before data chunks.
- In SCTP, each data chunk is numbered using a transmission sequence number (TSN).
- To distinguish between different streams, SCTP uses the sequence identifier (SI).
- To distinguish between different data chunks belonging to the same stream, SCTP uses the stream sequence number (SSN).
- Data chunks are identified by three identifiers: TSN, SI, and SSN. TSN is a cumulative number recognized by the whole association; SSN starts from 0 in each stream.
- SCTP acknowledgment numbers are used only to acknowledge data chunks; control chunks are acknowledged, if needed, by another control chunk.
- An SCTP association is normally established using four packets (four-way handshaking). An association is normally terminated using three packets (three-way handshaking).
- An SCTP association uses a cookie to prevent blind flooding attacks and a verification tag to avoid insertion attacks.
- SCTP provides flow control, error control, and congestion control.
- The SCTP acknowledgment SACK reports the cumulative TSN, the TSN of the last data chunk received in order, and selective TSNs that have been received.

23.8 PRACTICE SET

Review Questions

1. In cases where reliability is not of primary importance, UDP would make a good transport protocol. Give examples of specific cases.

1. ***Reliability*** is not of primary importance in applications such as echo, daytime, BOOTP, TFTP and SNMP. In custom software, reliability can be built into the client/server applications to provide a more reliable, low overhead service.

---

2. Are both UDP and IP unreliable to the same degree? Why or why not?

2. IP and UDP are both ***connectionless*** and ***unreliable protocols***. The main difference in their reliability is that IP only calculates a checksum for the IP header and not for the data while UDP calculates a checksum for the entire datagram.

---

3. Do port addresses need to be unique? Why or why not? Why are port addresses shorter than IP addresses?

3. ***Port addresses*** do not need to be universally unique as long as each IP address/port address pair uniquely identify a particular process running on a particular host. A good example would be a network consisting of 50 hosts, each running echo server software. Each server uses the well known port number 7, but the IP address, together with the port number of 7, uniquely identify a particular server program on a particular host. Port addresses are ***shorter*** than IP addresses because their domain, a single system, is smaller than the domain of IP addresses, all systems on the Internet.

---

4. What is the dictionary definition of the word *ephemeral?* How does it apply to the concept of the ephemeral port number?

4. ***Ephemeral*** is defined as short-lived or transitory. Ephemeral port numbers are only used for the duration of a single communication between client and server, so they are indeed short-lived.

---

5. What is the minimum size of a UDP datagram?

5. The minimum size of a UDP datagram is **8** bytes at the transport layer and **28** bytes at the IP layer. This size datagram would contain no data–only an IP header with no options and a UDP header. The implementation may require padding.

-

---

6. What is the maximum size of a UDP datagram?

6. Since the length of a datagram must be contained in a 2 byte field, the maximum size of a UDP datagram is **65,535** bytes (header plus data). However, given that the IP layer must also store the total length of the packet in a 2 byte field, the maximum length would be 20 bytes less than this, or **65,515** bytes, to leave room for the IP header. The implementation may impose a smaller limit than this.

---

7. What is the minimum size of the process data that can be encapsulated in a UDP datagram?

7. The smallest amount of process data that can be encapsulated in a UDP datagram is **0** bytes.

8. What is the maximum size of the process data that can be encapsulated in a UDP datagram?

8. The largest amount of process data that can be encapsulated in a UDP datagram is **65,507** bytes. (65,535 minus 8 bytes for the UDP header minus 20 bytes for the IP header). The implementation may impose a smaller limit than this.

9. Compare the TCP header and the UDP header. List the fields in the TCP header that are missing from UDP header. Give the reason for their absence.

9. See Table 23.1.

Table 23.1 *Answer to the Question 9.*

Fields in UDP	*Fields in TCP*	*Explanation*
Source Port Address	Source Port Address	
Destination Port Address	Destination Port Address	
Total Length		There is no need for total length.
Checksum	Checksum	
	Sequence Number	UDP has no flow and error control.
	Acknowledge Number	UDP has no flow and error control.
	Header Length	UDP has no flow and error control.
	Reserved	UDP has no flow and error control.
	Control	UDP has no flow and error control.
	Window Size	UDP has no flow and error control.
	Urgent Pointer	UDP cannot handle urgent data.
	Options and Padding	UDP uses no options.

10. UDP is a message-oriented protocol. TCP is a byte-oriented protocol. If an application needs to protect the boundaries of its message, which protocol should be used, UDP or TCP?

10. ***UDP*** is preferred because each user datagram can be used for each chunk of data. However, a better solution is ***SCTP***.

---

11. What can you say about the TCP segment in which the value of the control field is one of the following?
    a. 000000
    b. 000001
    c. 010001

11.
    a. None of the control bits are set. The segment is part of a data transmission without piggybacked acknowledgment.
    b. The ***FIN*** bit is set. This is a FIN segment request to terminate the connection.
    c. The ***ACK*** and the ***FIN*** bits are set. This is a ***FIN+ACK*** in response to a received ***FIN*** segment.

---

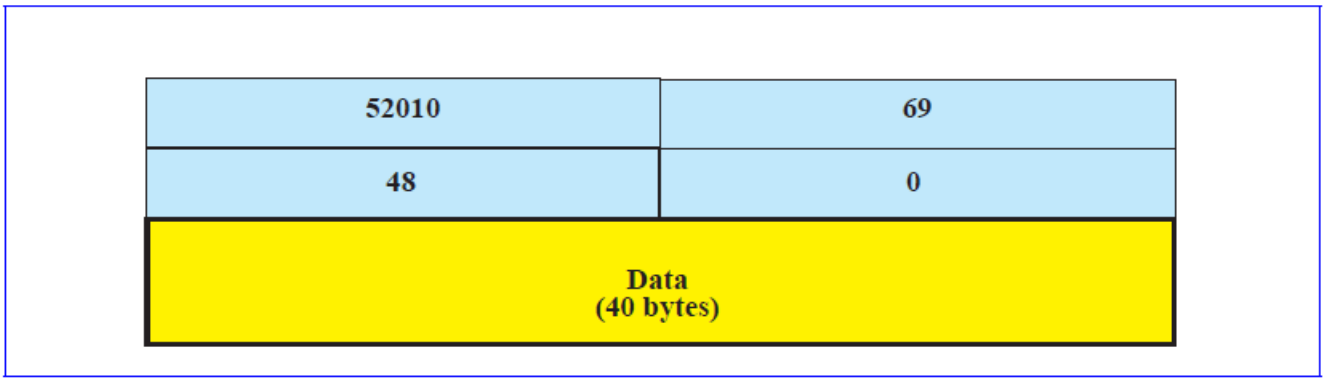
12. What is the maximum size of the TCP header? What is the minimum size of the TCP header?

12. The ***maximum size*** of the TCP header is **60** bytes. The ***minimum size*** of the TCP header is **20** bytes.

Exercises

13. Show the entries for the header of a UDP user datagram that carries a message from a TFTP client to a TFTP server. Fill the checksum field with Os. Choose an appropriate ephemeral port number and the correct well-known port number. The length of data is 40 bytes. Show the UDP packet, using the format in Figure 23.9.

Figure 23.1 *Solution to Exercise 13*



---

14. An SNMP client residing on a host with IP address 122.45.12.7 sends a message to an SNMP server residing on a host with IP address 200.112.45.90. What is the pair of sockets used in this communication?

14. The client would use the IP address **122.45.12.7**, combined with an ephemeral port number, for its source socket address and the IP address **200.112.45.90**, combined with the well- known port number **161**, as the destination socket address.

---

15. A TFTP server residing on a host with IP address 130.45.12.7 sends a message to a TFTP client residing on a host with IP address 14.90.90.33. What is the pair of sockets used in this communication?

15. The server would use the IP address **130.45.12.7**, combined with the well-known port number **69** for its source socket address and the IP address **14.90.90.33**, combined with an ephemeral port number as the destination socket address.

---

16. A client has a packet of 68,000 bytes. Show how this packet can be transferred by using only one UDP user datagram.

16. This datagram ***cannot be transferred*** using a single user datagram.

---

17. A client uses UDP to send data to a server. The data are 16 bytes. Calculate the efficiency of this transmission at the UDP level (ratio of useful bytes to total bytes).

17. 16 bytes of data / 24 bytes of total length = **0.666**

---

18. Redo Exercise 17, calculating the efficiency of transmission at the IP leveL Assume no options for the IP header.

18. 16 bytes of data / 44 bytes of total length = **0.364**

---

19. Redo Exercise 18, calculating the efficiency of transmission at the data link layer. Assume no options for the IP header and use Ethernet at the data link layer.

19. 16 bytes of data / 72 byte minimum frame size = **0.222**

---

20. The following is a dump of a UDP header in hexadecimal format.

0632000DOO lCE217

a. What is the source port number?
b. What is the destination port number?
c. What is the total length of the user datagram?
d. What is the length of the data?
e. Is the packet directed from a client to a server or vice versa?
f. What is the client process?

20.

a. Port number **1586**
b. Port number **13**
c. **28** bytes
d. **20** bytes (28 – 8 byte header)
e. ***From a client to a server***
f. ***Daytime***

---

21. An IP datagram is carrying a TCP segment destined for address 130.14.16.17/16. The destination port address is corrupted, and it arrives at destination 130.14.16.19/16. How does the receiving TCP react to this error?

21. It looks as if both the destination IP address and the destination port number are corrupted. ***TCP calculates the checksum and drops the segment***.

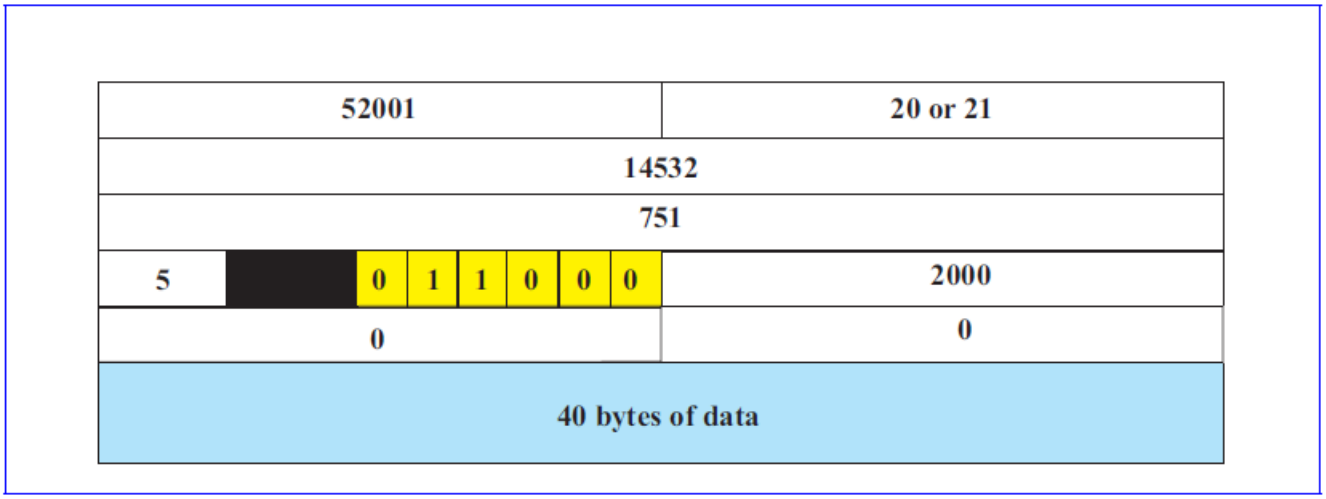
---

22. In TCP, if the value of HLEN is 0111, how many bytes of option are included in the segment?

22. 0111 in decimal is 7. The total length of the header is 7 × 4 or **28**. The base header is **20** bytes. The segment has **8** bytes of options.

23. Show the entries for the header of a TCP segment that carries a message from an FTP client to an FTP server. Fill the checksum field with 0s. Choose an appropriate ephemeral port number and the correct well-known port number. The length of the data is 40 bytes.

Figure 23.2 *Solution to Exercise 23*



24. The following is a dump of a TCP header in hexadecimal format.

05320017 00000001 00000000 5OO207FF 00000000

a. What is the source port number?
b. What is the destination port number?
c. What the sequence number?
d. What is the acknowledgment number?
e. What is the length of the header?
f. What is the type of the segment?
g. What is the window size?

24.

a. The source port number is **0x0532** (**1330** in decimal).
b. The destination port number is **0x0017** (**23** in decimal).
c. The sequence number is **0x00000001** (**1** in decimal).

d. The acknowledgment number is **0x00000000** (**0** in decimal).
e. The header length is **0x5** (**5** in decimal). There are 5 × 4 or **20** bytes of header.
f. The control field is **0x002**. This indicates a SYN segment used for connection establishment.
g. The window size field is **0x07FF** (**2047** in decimal).

---

25. To make the initial sequence number a random number, most systems start the counter at 1 during bootstrap and increment the counter by 64,000 every 0.5 s. How long does it take for the counter to wrap around?

25. Every second the counter is incremented by 64,000 × 2 = **128,000**. The sequence number field is 32 bits long and can hold only $2^{32}-1$. So it takes $(2^{32}-1)/(128{,}000)$ seconds or **33,554** seconds.

---

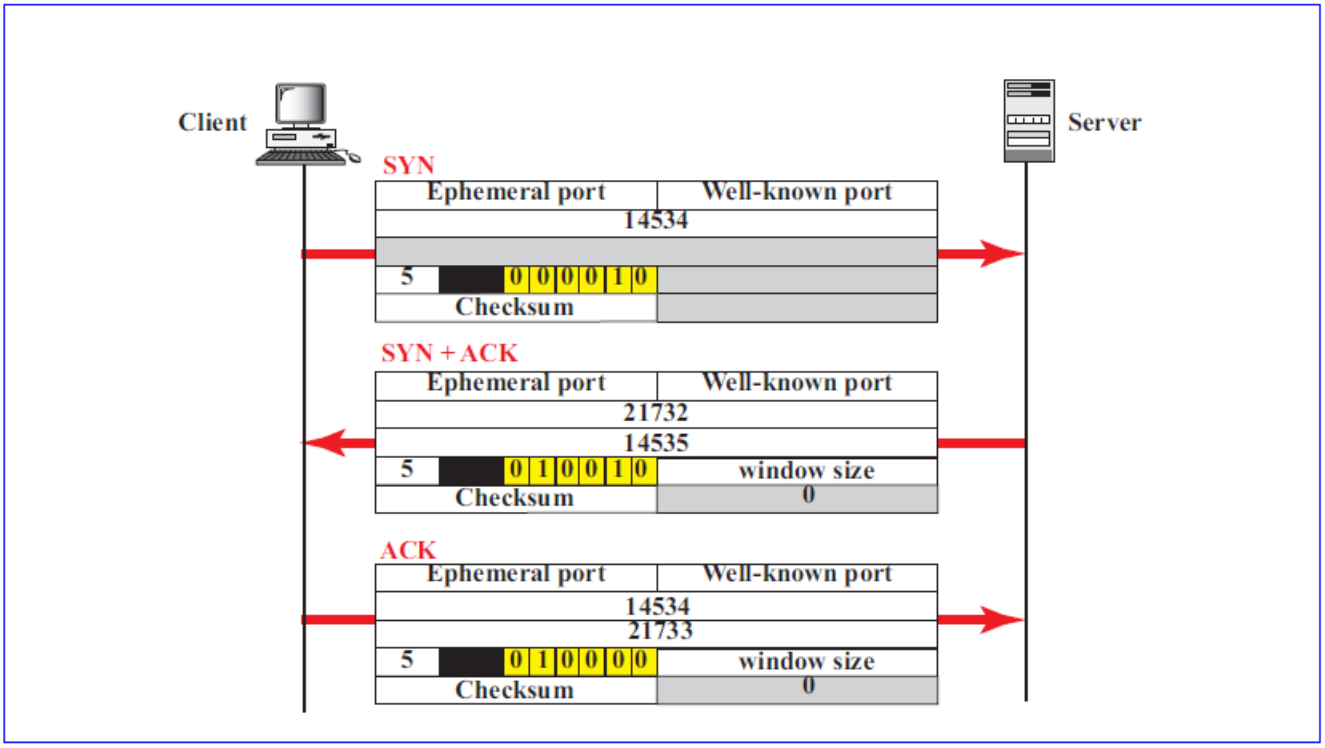
26. In a connection, the value of *cwnd* is 3000 and the value of *rwnd* is 5000. The host has sent 2000 bytes which has not been acknowledged. How many more bytes can be sent?

26. 3000 – 2000 = **1000** bytes

---

27. TCP opens a connection using an initial sequence number (ISN) of 14,534. The other party opens the connection with an ISN of 21,732. Show the three TCP segments during the connection establishment.

Figure 23.3 *Solution to Exercise 27*



28. A client uses TCP to send data to a server. The data are 16 bytes. Calculate the efficiency of this transmission at the TCP level (ratio of useful bytes to total bytes). Calculate the efficiency of transmission at the IP level. Assume no options for the IP header. Calculate the efficiency of transmission at the data link layer. Assume no options for the IF header and use Ethernet at the data link layer.

28.

a. **At TCP level:**

16 bytes of data / (16 bytes of data + 20 bytes of TCP header) ≈ **0.44** or **44** percent

b. **At IP level**

16 bytes of data / (16 bytes of data + 20 bytes of TCP header
+ 20 bytes of IP header) ≈ **0.29** or **29** percent

c. **At data link level** (assuming no preamble or flag)**:**

16 bytes of data / (16 bytes of data + 20 bytes of TCP header + 20 bytes of IP header
+ 18 bytes of Ethernet header and trailer ≈ **0.22** or **22** percent,

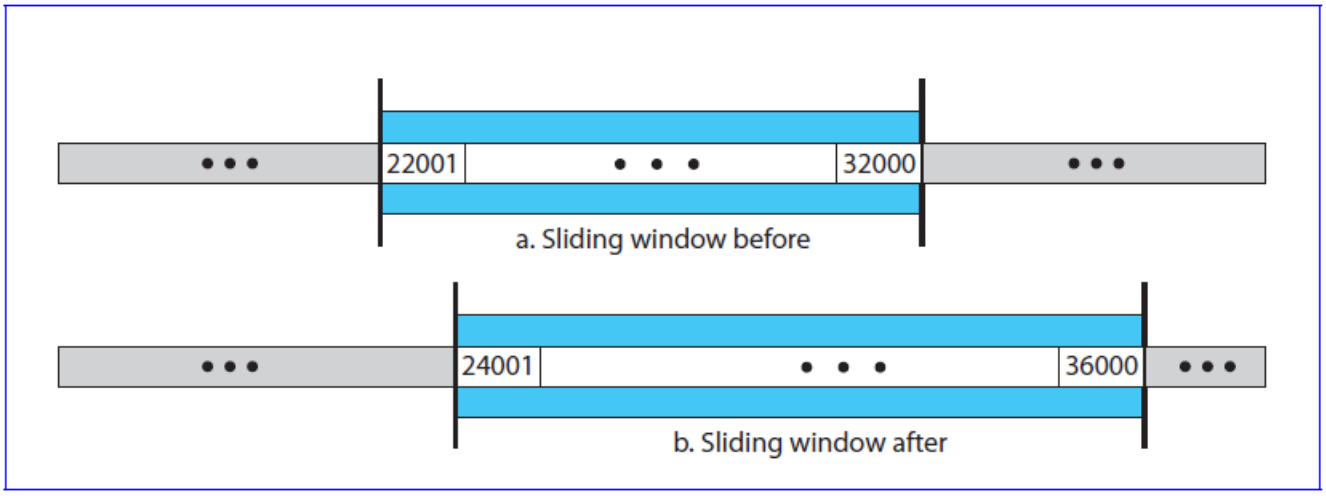
29. TCP is sending data at 1 Mbyte/s. If the sequence number starts with 7000, how long does it take before the sequence number goes back to zero?

29. The largest number in the sequence number field is $2^{32} - 1$. If we start at 7000, it takes $[(2^{32} - 1) - 7000] / 1{,}000{,}000 = \mathbf{4295}$ seconds.

---

30. A TCP connection is using a window size of 10,000 bytes, and the previous acknowledgment number was 22,001. It receives a segment with acknowledgment number 24,001 and window size advertisement of 12,000. Draw a diagram to show the situation of the window before and after.

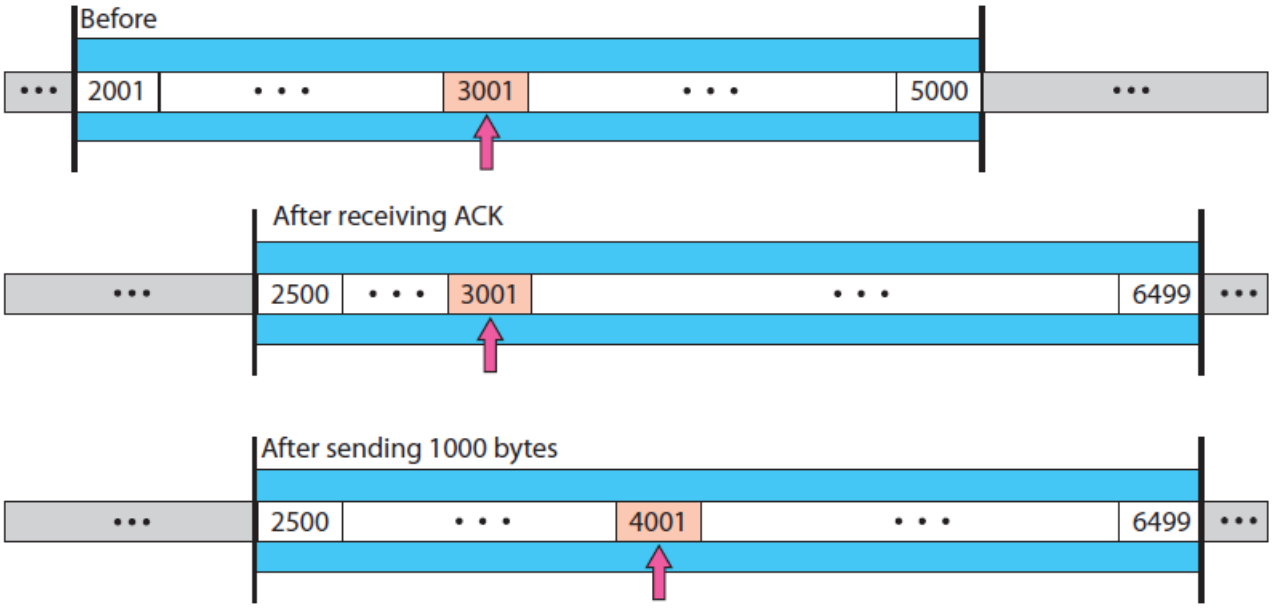
Figure 23.4 *Solution to Exercise 30*



---

31. A window holds bytes 2001 to 5000. The next byte to be sent is 3001. Draw a figure to show the situation of the window after the following two events.
    a. An ACK segment with the acknowledgment number 2500 and window size advertisement 4000 is received.
    b. A segment carrying 1000 bytes is sent.

Figure 23.5 *Solution to Exercise 31*



---

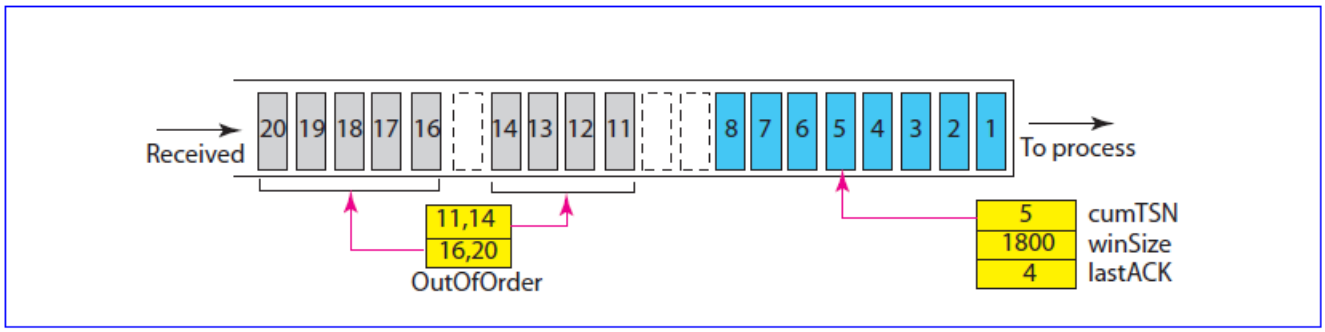
32. In SCTP, the value of the cumulative TSN in a SACK is 23. The value of the previous cumulative TSN in the SACK was 29. What is the problem?

32. The SACK chunk with a cumTSN of **23** was delayed.

---

33. In SCTP, the state of a receiver is as follows:
    a. The receiving queue has chunks 1 to 8, 11 to 14, and 16 to 20.
    b. There are 1800 bytes of space in the queue.
    c. The value of *lastAck* is 4.
    d. No duplicate chunk has been received.
    e. The value of *cumTSN* is 5.

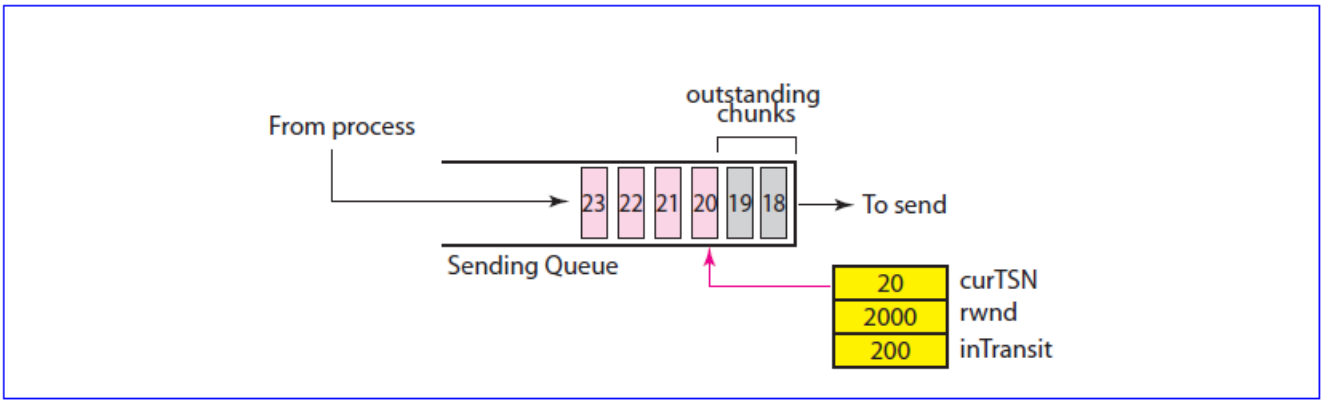
Figure 23.6 *Solution to Exercise 33*



34. In SCTP, the state of a sender is as follows:
    a. The sending queue has chunks 18 to 23.
    b. The value of *cumTSN* is 20.
    c. The value of the window size is 2000 bytes.
    d. The value of *inTransit* is 200.

    If each data chunk contains 100 bytes of data, how many DATA chunks can be sent now? What is the next DATA chunk to be sent?

Figure 23.7 *Solution to Exercise 34*



24.12 SUMMARY

- The average data rate, peak data rate, maximum burst size, and effective band width are qualitative values that describe a data flow.
- A data flow can have a constant bit rate, a variable bit rate, or traffic that is bursty.
- Congestion control refers to the mechanisms and techniques to control congestion and keep the load below capacity.
- Delay and throughput measure the performance of a network.
- Open-loop congestion control prevents congestion; closed-loop congestion control removes congestion.

- TCP avoids congestion through the use of two strategies: the combination of slow start and additive increase, and multiplicative decrease.
- Frame Relay avoids congestion through the use of two strategies: backward explicit congestion notification (BECN) and forward explicit congestion notification (FECN).
- A flow can be characterized by its reliability, delay, jitter, and bandwidth.
- Scheduling, traffic shaping, resource reservation, and admission control are techniques to improve quality of service (QoS).
- FIFO queuing, priority queuing, and weighted fair queuing are scheduling techniques.
- Leaky bucket and token bucket are traffic shaping techniques.
- Integrated Services is a flow-based QoS model designed for IP.
- The Resource Reservation Protocol (RSVP) is a signaling protocol that helps IP create a flow and makes a resource reservation.
- Differential Services is a class-based QoS model designed for IP.
- Access rate, committed burst size, committed information rate, and excess burst size are attributes to control traffic in Frame Relay.
- Quality of service in ATM is based on service classes, user-related attributes, and network-related attributes.