Solutions to Chapter 8

8.1. The IP header checksum only verifies the integrity of IP header. Discuss the pros and cons of doing the checksum on the header part versus on the entire packet.

Solution:

Error checking in the header is more important because the packet is routed according to the header information. In addition, the delivery of the data at the destination to the higher layers also requires the header information. Thus error checking of the header protects against misdelivery of the information. Restricting the error checking to the header also simplifies the implementation in the nodes, requires less checksum bits, and prevents unnecessary packet discard. Some higher layers can tolerate some data errors, and higher layers also have the option of performing retransmission.

8.2. Identify the address class of the following IP addresses: 200.58.20.165; 128.167.23.20; 16.196.128.50; 50.156.10.10; 250.10.24.96.

Solution:

An IP address has a fixed length of 32 bits, where the most significant bits identify the particular class. Therefore, to identify the address class we need to convert the dotted-decimal notation back into its binary counterpart, and compare the binary notation to the class prefixes shown in Figure 8.5 in the text. (Recall that the dotted-decimal notation was devised to communicate addresses more readily to other people. In this notation, the 32 bits are divided into four groups of 8 bits – separated by periods – and then converted to their decimal counterpart.) The first few bits (shown in red) of the address can be used to determine the class.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

200.58.20.165
11001000.00111010.00010100.10100101
Class C

128.167.23.20
10000000.10100111.00010111.00010100
Class B

16.196.128.50
00010000.11000100.10000000.00110010
Class A

150.156.10.10
10010110.10011100.00001010.00001010
Class B

250.10.24.96
11111010.00001010.00011000.01100000
Class E

8.3. Convert the IP addresses in Problem 8.2 to their binary representation.

Solution:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

200.58.20.165
11001000.00111010.00010100.10100101

128.167.23.20
10000000.10100111.00010111.00010100

16.196.128.50
00010000.11000100.10000000.00110010

150.156.10.10
10010110.10011100.00001010.00001010

250.10.24.96
11111010.00001010.00011000.01100000

8.4. Identify the range of IPv4 addresses spanned by Class A, Class B, and Class C.

Solution:

The range of IPv4 addresses spanned by each class is:
Class A: 1.0.0.0 to 127.255.255.255
Class B: 128.0.0.0 to 191.255.255.255
Class C: 192.0.0.0 to 223.255.255.255

8.5. What are all the possible subnet masks for the Class C address space? List all the subnet masks in dotted-decimal notation, and determine the number of hosts per subnet supported for each subnet mask.

Solution:

255.255.255.128 supports 126 hosts (not including the broadcast address)

255.255.255.192 supports 62 hosts

255.255.255.224 supports 30 hosts

255.255.255.240 supports 14 hosts

255.255.255.248 supports 7 hosts

255.255.255.252 supports 3 hosts

255.255.255.254 and 255.255.255.255 are not practically usable.

8.6. A host in an organization has an IP address 150.32.64.34 and a subnet mask 255.255.240.0. What is the address of this subnet? What is the range of IP addresses that a host can have on this subnet?

Solution:

Address: 10010110 00100000 01000000 00100010

Mask: 11111111 11111111 11110000 00000000

Subnet: **10010110 00100000 0100**0000 00000000

Host:

From: 10010110 00100000 0100**0000 00000001**

To: 10010110 00100000 0100**1111 11111110**

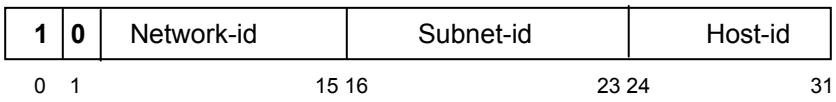
8.7. A university has 150 LANs with 100 hosts in each LAN.

Solutions follow questions:

(a) Suppose the university has one Class B address. Design an appropriate subnet addressing scheme.

A Class B address has 14 bits for the network ID and 16 bits for the host ID. To design an appropriate subnet addressing scheme we need to decide how many bits to allocate to the host ID versus the subnet ID. We can choose either 7 bits or 8 bits to identify the hosts.

If we allocate 8 bits for to identify the host, as shown below, then there are sufficient subnet-id bits to cover up to 2^8=256 LANs and enough host-id bits to cover up to 256 hosts for each LAN. The subnet mask in this case is 255.255.255.0



Subnet mask: 255.255.255.0

If we allocate 7 bits for to identify the host, as shown below, then there are sufficient subnet-id bits to cover up to 2^9=512 LANs and enough host-id bits to cover up to 128 hosts for each LAN. The subnet mask in this case is 255.255.255.128.

The choice between 7 or 8 bits to represent the hosts depends on which is likely to grow more, the number of subnets or the number of hosts in a LAN. Alternatively a variable-length prefix scheme using 7-bit host addresses, and grouping these form larger subnets provides greater flexibility in accommodating future changes.

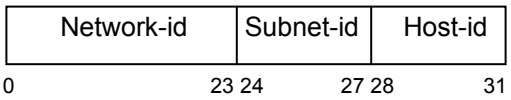
(b) Design an appropriate CIDR addressing scheme.

CIDR addressing scheme involves devising a prefix length that indicates the length of the network mask. In this case, 8 bits are required to identify each LAN (since 127 < 150 < 255) and 7 bits are required to identify each host in each LAN (since 63 < 100 < 127). Therefore a CIDR address would use a 17-bit prefix, and thus have an address of the form address/17.

8.8. A small organization has a Class C address for seven networks each with 24 hosts. What is an appropriate subnet mask?

Solution:

A Class C address requires 21 bits for its network ID, leaving 8 bits for the host ID and subnet ID to share. One possible scheme would assign 4 bits to the host and 4 to the subnet ID, as shown below. The number of bits assigned to the host can be increased to 5 as well.



Subnet mask: 255.255.255.224

8.9. A packet with IP address 150.100.12.55 arrives at router R1 in Figure 8.8. Explain how the packet is delivered to the appropriate host.

Solution:

The packet with IP address 150.100.12.55 arrives from the outside network. R1 has to know the next-hop router or host to send the packet to. The address corresponds to the binary string 10010110.01100100.00001100.00110111. R1 knows that a 9 bit subnet field is in use so it applies the following mask to extract the subnetwork address from the IP address.
11111111.11111111.11111111.10000000

The resulting IP address is 10010110.01100100.00001100.00000000 and corresponds 150.100.12.0. This indicates that the host is in subnet 150.100.12.0, so the router transmits the IP packet on this (attached) LAN.

8.10. In Figure 8.8 assign a physical layer address 1, 2, … to each physical interface starting from the top row, moving right to left, and then moving down. Suppose H4 sends an IP packet to H1. Show the sequence of IP packets and Ethernet frames exchanged to accomplish this transfer.

Solution:

1. Send IP packet from H4 to R1:
Source address 150.100.12.55 to destination IP address 150.100.12.176
Source Ethernet 4 to Receive Ethernet 6
2. Forward IP packet from R1 to H1
Source address 150.100.12.55 to destination IP address 150.100.12.176
Source Ethernet 3 to Receive Ethernet 2

8.11. ARP is used to find the MAC address that corresponds to an IP address; RARP is used to find the IP address that corresponds to a MAC address. True or false?

Solution:

True, ARP is used to find the MAC address for a given IP address.

True, Reverse ARP is used by a device to find its IP address given its MAC address.

8.12. Perform CIDR aggregation on the following /24 IP addresses: 128.56.24.0/24; 128.56.25.0./24; 128.56.26.0/24; 128.56.27.0/24.

Solution:

128.56.24.0/$_{22}$ = 10000000.00111000.00011000.00000000
128.56.25.0/$_{22}$ = 10000000.00111000.00011001.00000000
128.56.26.0/$_{22}$ = 10000000.00111000.00011010.00000000
128.56.27.0/$_{22}$ = 10000000.00111000.00011011.00000000
mask = 11111111.11111111.11111100.00000000
The resulting prefix is 128.56.24.0/22.

8.13. Perform CIDR aggregation on the following /24 IP addresses: 200.96.86.0/24; 200.96.87.0/24; 200.96.88.0/24; 200.96.89.0/24.

Solution:

200.96.86.0/$_{20}$ = 11001000.01100000.01010110.00000000
200.96.87.0/$_{20}$ = 11001000.01100000.01010111.00000000
200.96.88.0/$_{20}$ = 11001000.01100000.01011000.00000000
200.96.89.0/$_{20}$ = 11001000.01100000.01011001.00000000
mask = 11111111.11111111.11110000.00000000
The resulting prefix is 200.96.80.0/20.

8.14. The following are estimates of the population of major regions of the world: Africa 900 million; South America 500 million; North America 400 million; East Asia 1500 million; South and Central Asia 2200 million; Russia 200 million; Europe 500 million.

Solutions follow questions:

(a) Suppose each region is to be assigned 100 IP addresses per person. Is this possible? If not, how many addresses can be assigned per person? Repeat for IPv6.

The total number of IPv4 addresses is: 2^{32} which is approximately 4.29 billion. The above world population estimate totals 6.2 billion, so it is not possible to assign an individual address to each person. IPv6 is required to provide 100 addresses per person.

(b) Design an appropriate CIDR scheme to provide the addressing in part (a).

	(millions)	(millions)	IPV6 128 bits	
Regions	**Population**	**IP/Person**	**Net-id**	**Host-id**
Africa	900	90000	91	37
South America	500	50000	92	36
North America	400	40000	92	36
East Asia	1500	150000	90	38
South C. Asia	2200	220000	90	38
Russia	200	20000	93	35
Europe	500	50000	92	36

8.15. Suppose four major ISPs were to emerge with points of presence in every major region of the world. How should a CIDR scheme treat these ISPs in relation to addressing for each major region?

Solution:

The networks of these ISPs will span across national and geographical boundaries and will connected in a non-hierarchical manner. These large global ISPs constitute major transit routing domains, so it makes sense to assign them blocks of unique IP addresses and to require that domains attached to them should begin with the transit domain's prefix. The blocks of addresses should be managed so that fragmentation of the CIDR block does not take place. For example, when a customer of the ISP changes to another ISP, the customer is required to return the block of addresses. This policy enables CIDR to be effective in controlling the size of routing tables.

8.16. Discuss the difficulties with using actual time in the TTL field.

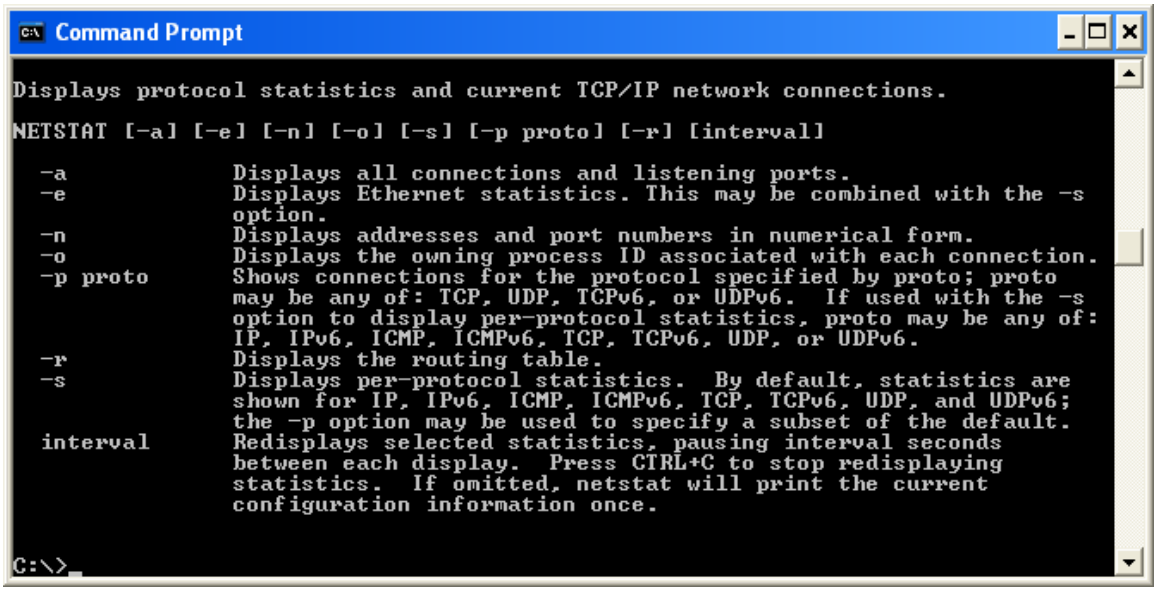
Solution:

Unlike the number of hops, which is predictable if a packet is routed correctly, the actual time that it takes to go through the route is not predictable. Therefore the amount of time that a packet stays in the network is not necessarily an indication of misrouting. To allow an upper limit for delay across the network TTL field would become a very large number. It is also more complex to track and update the TTL according to actual time as the packet traverses the network.

8.17. Lookup the `netstat` command in the manual for your system. Use the command to display the routing table in your host. Try the different command options.

Solution:

The exact command depends on your computing environment. In a DOS (Windows) environment, the command may be `netstat -r`. In a SUN/UNIX environment, the command may be `netstat -a`. See the answer to problem 45 in Chapter 1 for an example routing table. The figure below shows a screen capture of the parameters for the command in Windows XP.



8.18. Suppose a router receives an IP packet containing 600 data bytes and has to forward the packet to a network with maximum transmission unit of 200 bytes. Assume that the IP header is 20 bytes long. Show the fragments that the router creates and specify the relevant values in each fragment header (i.e., total length, fragment offset, and more bit).

Solution:

Given:
IP packet = 600 data bytes
MTU = 200 bytes
IP header = 20 header bytes
Maximum possible data length per fragment = MTU – IP header = 200 – 20 = 180 bytes.
The data length of each fragment must be a multiple of eight bytes; therefore the maximum number of data bytes that can be carried per fragment is 22*8=176.

The data packet must be divided into 4 frames, as shown by the following calculations:
176 + 176 + 176 + 72 = 600
20 + 20 + 20 + 20
196 196 196 92

The sequence of frames and packet headers is shown below:

	Total length	Id	Mf	Fragment Offset
Original Packet	620	x	0	0
Fragment 1	196	x	1	0
Fragment 2	196	x	1	22
Fragment 3	196	x	1	44
Fragment 4	92	x	0	66

8.19. Design an algorithm for reassembling fragments of an IP packet at the destination IP.

Solution:

I. Set data = Null
II. Verify that all fragments for id = x have arrived
III. Sort fragments in ascending order based on fragment offset
IV. For each fragment starting with fragment offset = 0, move data = data + data-in-fragment$_I$
V. Data contains the reassembled information

8.20. Does it make sense to do reassembly at intermediate routers? Explain.

Solution:

No, because the packet may be de-fragmented again, and all the time required to wait for all fragments and to reassemble the packet will be wasted. Also it is not guaranteed that all fragments go through the same path and arrive at the same node in a datagram network such as IP.

8.21. Describe the implementation issues of IPv4 packet processing.

Solution:

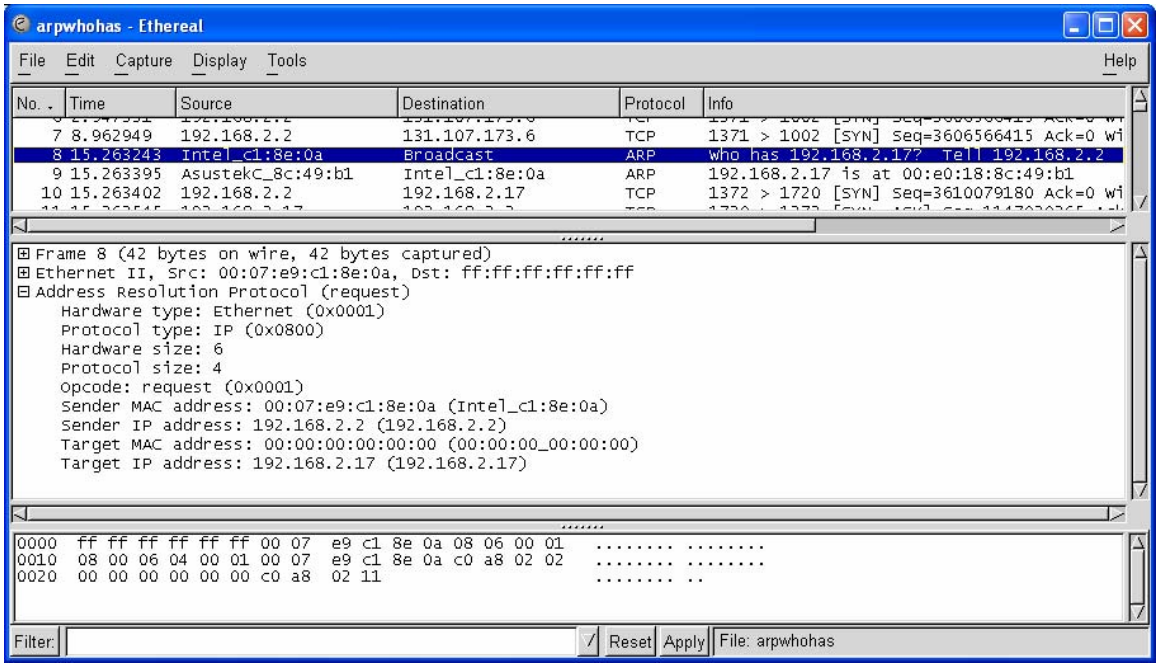
The router performs error checking: the header checksum is computed; the version and total length fields are checked for valid values. The router finds the next-hop by consulting its routing table. The

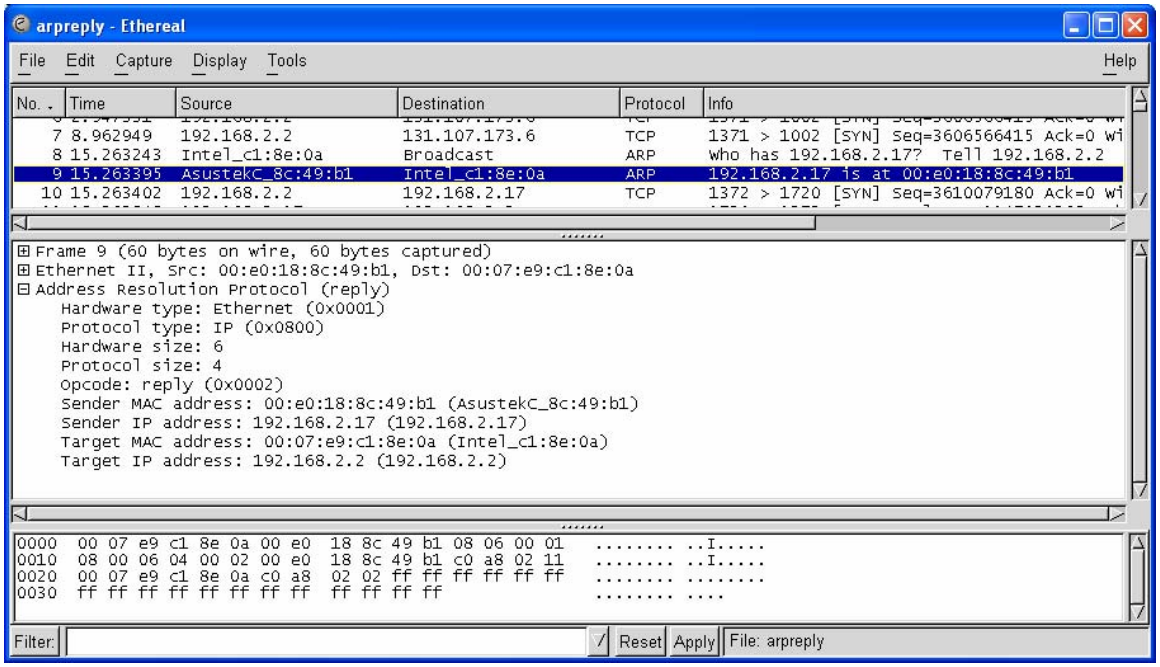
router then updates various fields, including the TTL and checksum fields. The packet is then forwarded.

8.22. Use Ethereal to capture ARP packets to find the MAC addresses in a LAN.

Solution:

The following screen captures show and ARP request and the corresponding ARP reply.





8.23. Abbreviate the following IPv6 addresses:

Solutions follow questions:

(a) 0000:0000:0F53:6382:AB00:67DB:BB27:7332

::F53:6382:AB00:67DB:BB27:7332

(b) 0000:0000:0000:0000:0000:0000:004D:ABCD

::4D:ABCD

(c) 0000:0000:0000:AF36:7328:0000:87AA:0398

::AF36:7328:0:87AA:398

(d) 2819:00AF:0000:0000:0000:0035:0CB2:B271

2819:AF::35:CB2:B271

8.24. What is the efficiency of IPv6 packets that carry 10 ms of 64 kbps voice? Repeat if an IPv6 packet carries 1 frame of 4 Mbps MPEG2 video, assuming a frame rate of 30 frames/second.

Solution:

10 ms of 64 kbps voice = $10 \times 10^{-3} \times 64 \times 10^{3} = 640$ bits = 80 bytes
Header = 40 bytes; Efficiency = 80/(80 +40) = 2/3 = 0.6666 = 66.7%.

1 frame of video is: $4 \times 10^{6}/30 = 133{,}333$ bits = 16666 bytes
Efficiency = 16666/(16666+40) = 99.76%.

8.25. Why does IPv6 allow fragmentation at the source only?

Solution:

The task of fragmenting a packet uses processing resources in a router. By requiring that all fragmentation be done at the source, routers are relieved of the fragmentation processing load, and hence they can operate faster on the basic routing task.

8.26. Assuming the population estimates in problem 8.14, how many IP addresses does IPv6 provide per capita?

Solution:

Based on the estimates in problem 10, there are $6{,}200 \times 10^{6}$ humans which in turn means 5.4×10^{28} IPv6 addresses per capita.

8.27. Suppose that IPv6 is used over a noisy wireless link. What is the effect of not having header error checking?

Solution:

The transmission over the noisy wireless link will introduce errors in the transmitted frames. If the frames do not contain error-checking, then assuming the frame is recognizable, erroneous packets may be passed to the router and unpredictable behavior may ensue. However, error checking (and retransmission) is included in most noisy wireless links; thus the effect of transmission errors is to trigger retransmission of frames in the link layer. Only packets that arrive in frames that pass error-checking are transferred to the IP layer.

8.28. Explain how the use of hierarchy enhances scalability in the following aspects of Internet:

Solutions follow questions:

(a) Domain name system

The use of hierarchy helps to speed up the translation of a domain name into an internet address. The search starts from the highest level (for example, .com, .org, .net) and eventually down to the specific hostname information. The hierarchy also helps to organize the database architecture of the DNS. Moreover, with a hierarchical system, it is not necessary for each DNS server to contain every single domain name in the network. There are different levels of DNS servers each containing the essential information for its own domain.

(b) IP addressing

Classful IP addressing uses hierarchy to arrange the address space in several discrete classes of addresses that correspond to networks of different sizes. CIDR IP addressing uses a variable-length prefix and a subnet mask to represent networks at a finer granularity of network size. In doing so, CIDR addressing increases the utilization of the address space. When combined with address allocation policies that aggregate routes, CIDR makes it possible to reduce the size of the routing tables required in router.

(c) OSPF routing

OSPF uses a two-level hierarchy that allows an AS to be partitioned into several groups called areas each interconnected by a central backbone area. This localization reduces the amount of routing information that needs to be maintained by individual routers. It also reduces the number of routing messages that need to be exchanged within the network.

(d) Interdomain routing

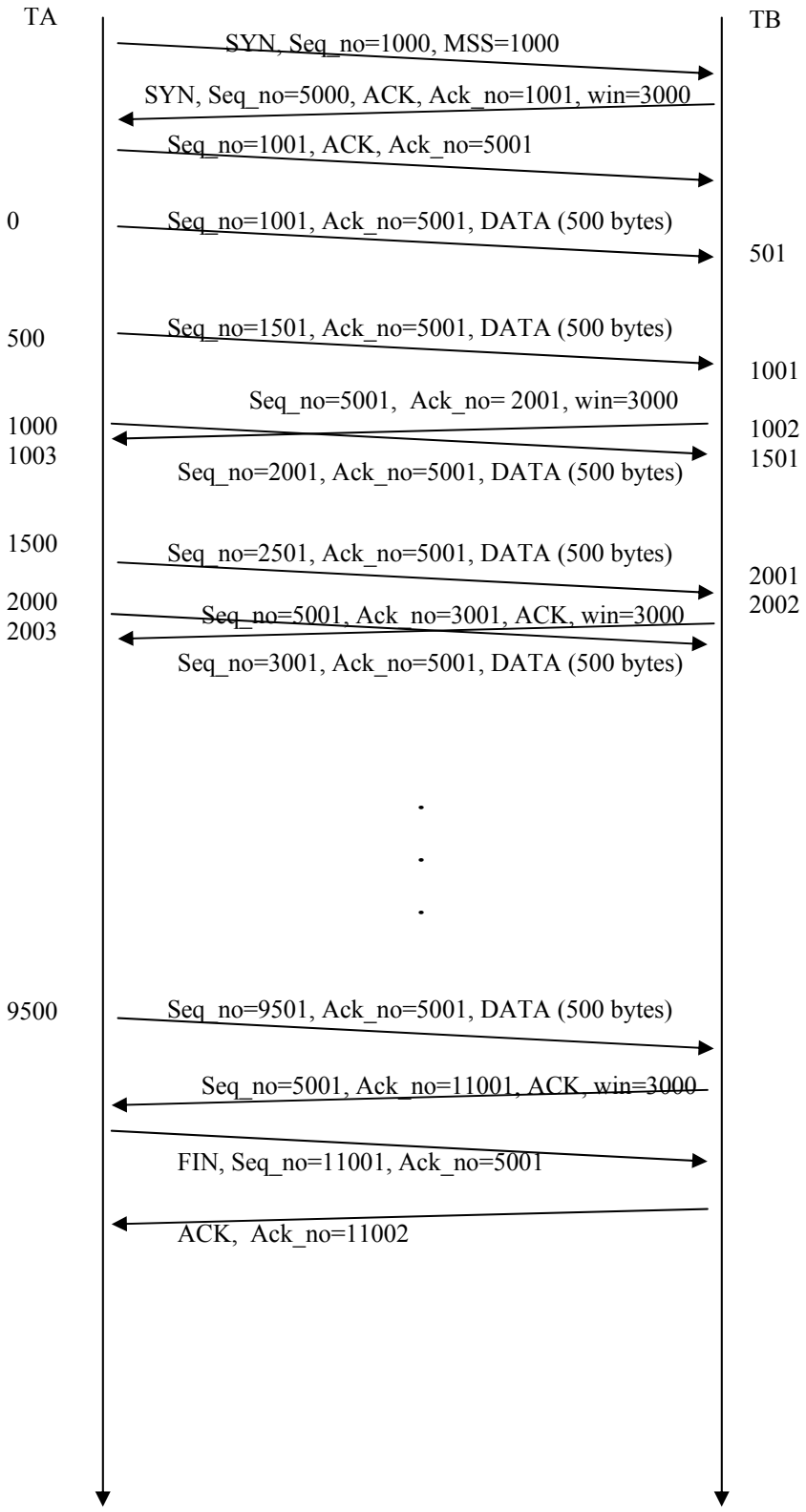
Interdomain routing uses the Border Gateway Protocol (BGP) to exchange routing information between AS's that in turn allows IP packets to flow across the AS border. Thus interdomain routing enables the scalability of the overall Internet by enabling various AS's to become interconnected.

8.29. The TCP in station A sends a SYN segment with ISN = 1000 and MSS = 1000 to station B. Station B replies with a SYN segment with ISN = 5000 and MSS = 500. Suppose station A has 10,000 bytes to transfer to B. Assume the link between stations A and B is 8 Mbps and the distance between them is 200 m. Neglect the header overheads to keep the arithmetic simple. Station B has 3000 bytes of buffer available to receive data from A. Sketch the sequence of segment exchanges, including the parameter values in the segment headers, and the state as a function of time at the two stations under the following situations:

Solutions follow questions:

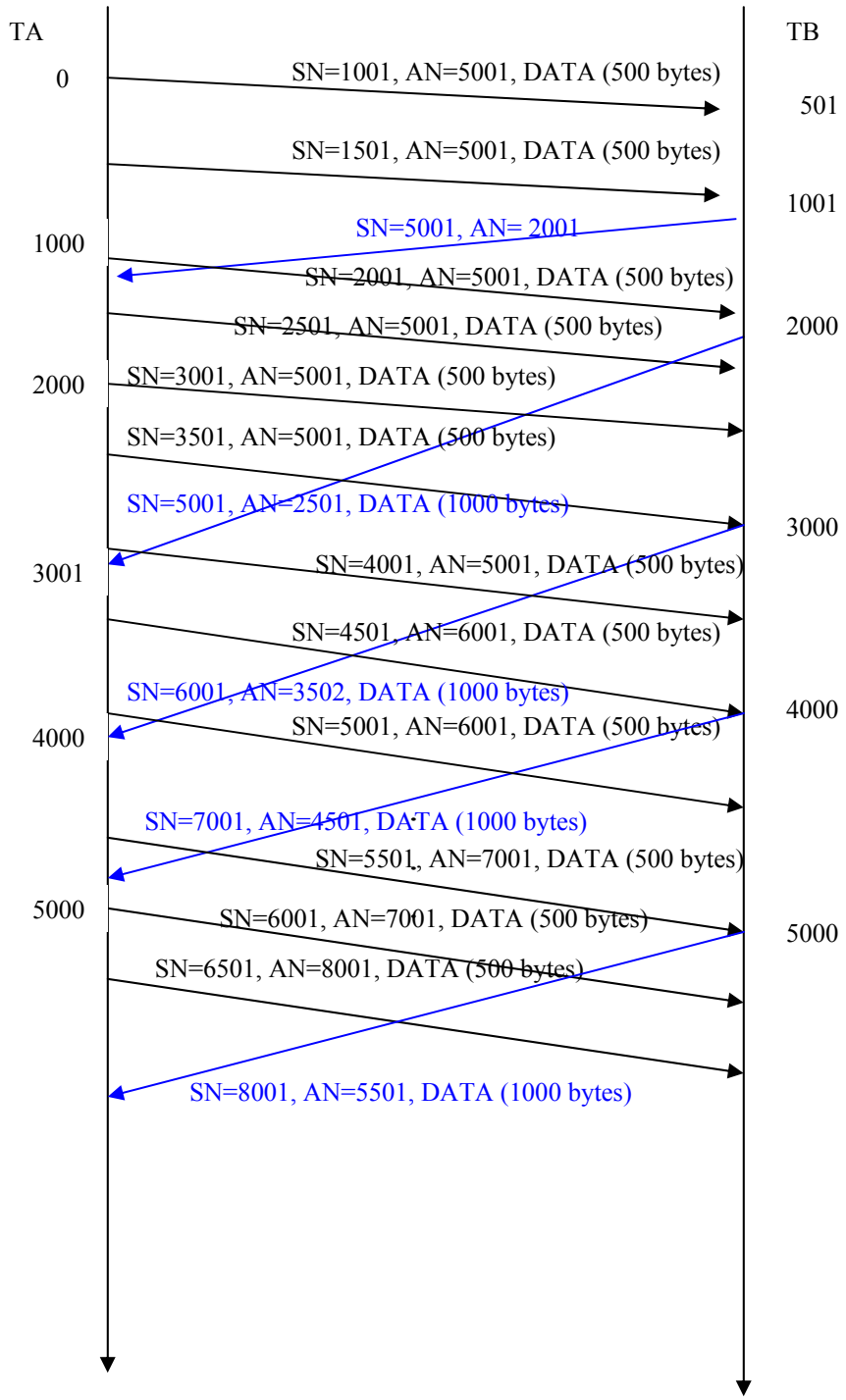
(a) Station A sends its first data segment at $t = 0$. Station B has no data to send and sends an ACK segment every other frame.

At a transmission rate of 8 megabits per second, a single byte has a transmission time of 8 bits / 8×10^6 bits/second = 1 microsecond. A distance of 200 meters in optical fiber has a propagation time of 200 meters / 2×10^8 meters/second = 1 microsecond. Therefore a segment of 500 bytes requires 501 microseconds to arrive completely at the receiver. In the following we also assume that the send window is replenished by the receiver as soon as it receives a segment. The time scale below is in microseconds.


TA
TB
SYN, Seq_no=1000, MSS=1000
SYN, Seq_no=5000, ACK, Ack_no=1001, win=3000
Seq_no=1001, ACK, Ack_no=5001
0
Seq_no=1001, Ack_no=5001, DATA (500 bytes)
501
500
Seq_no=1501, Ack_no=5001, DATA (500 bytes)
1001
Seq_no=5001, Ack_no= 2001, win=3000
1000
1002
1003
1501
Seq_no=2001, Ack_no=5001, DATA (500 bytes)
1500
Seq_no=2501, Ack_no=5001, DATA (500 bytes)
2001
2000
2002
Seq_no=5001, Ack_no=3001, ACK, win=3000
2003
Seq_no=3001, Ack_no=5001, DATA (500 bytes)
9500
Seq_no=9501, Ack_no=5001, DATA (500 bytes)
Seq_no=5001, Ack_no=11001, ACK, win=3000
FIN, Seq_no=11001, Ack_no=5001
ACK, Ack_no=11002

(b) Station A sends its first data segment at t=0. Station B has 6000 bytes to send, and it sends its first data segment at t = 2 ms.

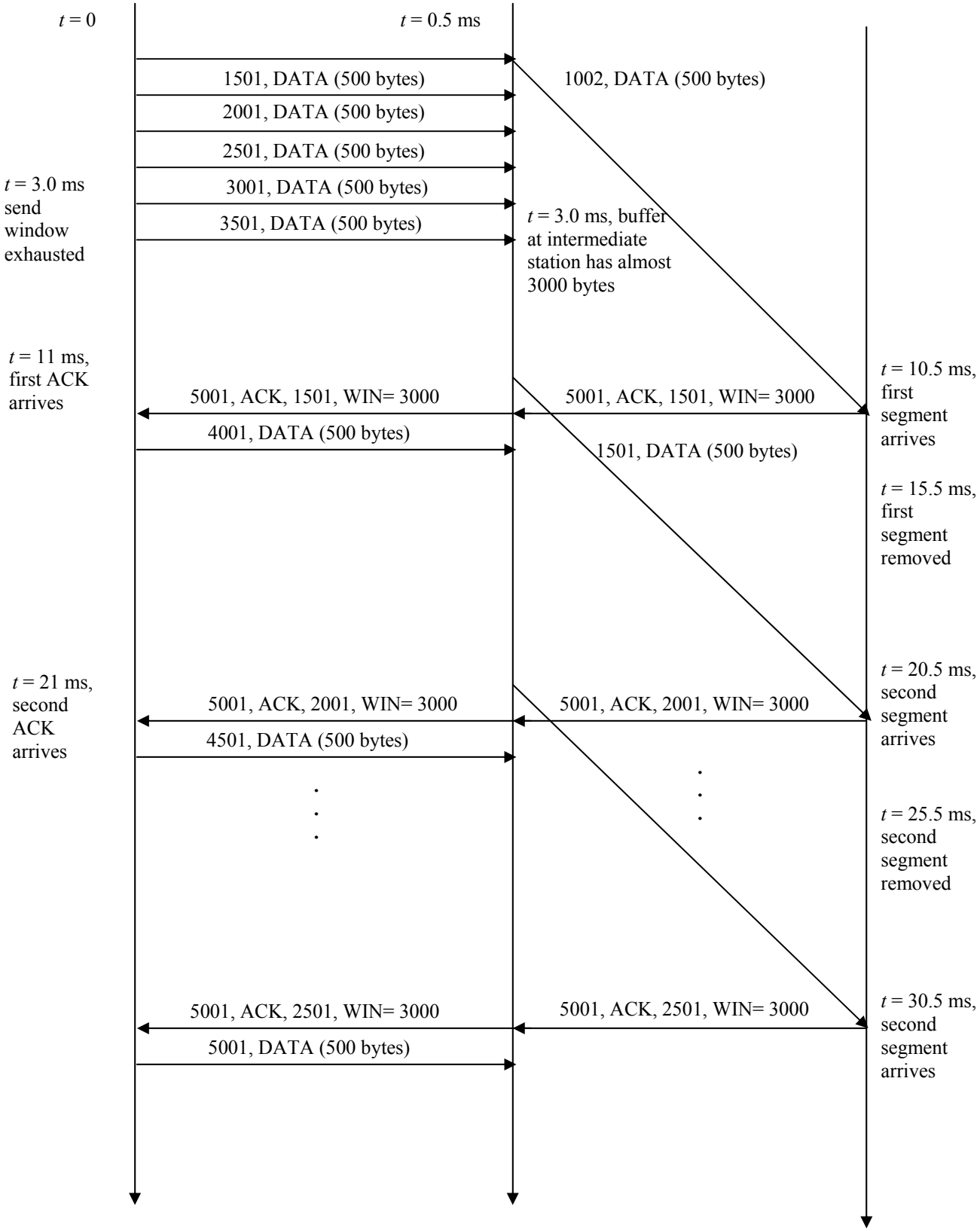
The main feature of this problem is that the acknowledgments are delayed longer because of the long segments that are transmitted from B to A.



8.30. Suppose that the TCP in station A sends information to the TCP in station B over a two-hop path. The data link in the first hop operates at a speed of 8 Mbps, and the data link in the second hop operates at a speed of 400 kbps. Station B has a 3 kilobyte buffer to receive information from A, and the application at station B reads information from the receive buffer at a rate of 800 kbps. The TCP in station A sends a SYN segment with ISN = 1000 and MSS = 1000 to station B. Station B replies with a SYN segment with ISN = 5000 and MSS = 500. Suppose station A has 10,000 bytes to transfer to B. Neglect the header overheads to keep the arithmetic simple. Sketch the sequence of segment exchanges, including the parameter values in the segment headers, and the state as a function of time at the two stations. Show the contents of the buffers in the intermediate switch as well as at the source and destination stations.

Solution:

It takes 500 microseconds to transmit 500 bytes from station A to the intermediate station, but it takes 10 milliseconds to send the same 500 bytes from the intermediate station to station B. Hence, segments will accumulate at the intermediate station until station A exhausts its send window of 3000 bytes. Eventually station A will receive acknowledgments that allow it to resume transmission. Note that the rate at which acknowledgments are returned to station A is controlled by the rate at which segments arrive at station B from the bottleneck at the intermediate node.


t = 0
t = 0.5 ms
1501, DATA (500 bytes)
1002, DATA (500 bytes)
2001, DATA (500 bytes)
2501, DATA (500 bytes)
t = 3.0 ms send window exhausted
3001, DATA (500 bytes)
3501, DATA (500 bytes)
t = 3.0 ms, buffer at intermediate station has almost 3000 bytes
t = 11 ms, first ACK arrives
5001, ACK, 1501, WIN= 3000
5001, ACK, 1501, WIN= 3000
t = 10.5 ms, first segment arrives
4001, DATA (500 bytes)
1501, DATA (500 bytes)
t = 15.5 ms, first segment removed
t = 21 ms, second ACK arrives
5001, ACK, 2001, WIN= 3000
5001, ACK, 2001, WIN= 3000
t = 20.5 ms, second segment arrives
4501, DATA (500 bytes)
t = 25.5 ms, second segment removed
5001, ACK, 2501, WIN= 3000
5001, ACK, 2501, WIN= 3000
t = 30.5 ms, second segment arrives
5001, DATA (500 bytes)

8.31. Suppose that the delays experienced by TCP segments traversing the network is equally likely to be any value in the interval [50 ms, 75 ms]. (See Equations 5.17 to 5.20.)

Solutions follow questions:

(a) Find the mean and standard deviation of the delay.

The delay lies between interval [50ms, 75ms] and is a uniform random variable. The mean is:

$E[X] = (50 + 75) / 2 = 62.5$ ms.

The standard deviation of delay is:

$$STD[x] = VAR[x]^{1/2} = [(75 - 50)^2 / 12]^{1/2} = 7.217$$

(b) Most computer languages have a function for generating uniformly distributed random variables. Use this function in a short program to generate random times in the above interval. Also, calculate t_{RTT} and d_{RTT} and compare to part (a).

See below for sample program written in C.

```
/* Communication Networks - Chapter 8    */
/* Question 25 (b)                       */
/* Description - Generate a random value */
/* between 50 to 75 ms.  Calculate t_RTT */
/* and d_RTT                             */
/* The min, max and avg value of t_RTT   */
/* and d_RTT are also recorded           */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main (void)
{
  int i;
  float temp, t_n, t_rtt_new, t_rtt_old;
  float d_rtt_new, d_rtt_old;
  float t_rtt_min, t_rtt_sum, t_rtt_max;
  float d_rtt_min, d_rtt_sum, d_rtt_max;

  const float alpha = 0.875;
  const float beta = 0.25;

  srand (time(NULL));
  t_rtt_old = 0;
  d_rtt_old = 0;

  t_rtt_sum = t_rtt_max = 0;
  d_rtt_sum = d_rtt_max = 0;

  t_rtt_min = d_rtt_min = 500;

  for (i = 0; i < 500; i++)
  {

    /* Generate a random value between 0 to 1 */
    temp = (float) rand() / RAND_MAX;

    /* Scale the random value to fit between 50 to 75 */
    t_n = temp * 25 + 50;
```

```
    /* Calculate t_RTT and d_RTT */
    t_rtt_new = (alpha * t_rtt_old) + ((1 - alpha) * t_n);
    d_rtt_new = (beta * d_rtt_old) + ((1 - beta) * fabs (t_n - t_rtt_old));

    if (t_rtt_new < t_rtt_min)
      t_rtt_min = t_rtt_new;
    if (t_rtt_new > t_rtt_max)
      t_rtt_max = t_rtt_new;

    if (d_rtt_new < d_rtt_min)
      d_rtt_min = d_rtt_new;
    if (d_rtt_new > d_rtt_max)
      d_rtt_max = d_rtt_new;

    t_rtt_sum += t_rtt_new;
    d_rtt_sum += d_rtt_new;

    printf ("t_RTT: %f d_RTT: %f\n", t_rtt_new, d_rtt_new);
    t_rtt_old = t_rtt_new;
    d_rtt_old = d_rtt_new;
  }
  printf ("t_RTT min: %f t_RTT max: %f t_RTT avg: %f\n",
           t_rtt_min, t_rtt_max, (t_rtt_sum / 500.0));
  printf ("d_RTT min: %f d_RTT max: %f d_RTT avg: %f\n",
           d_rtt_min, d_rtt_max, (d_rtt_sum / 500.0));
}
```

We ran the preceding program and obtained the average values of t_{RTT} = 61.6924 and d_{RTT} = 7.1139. These values are averaged from a sample of 500 values.

8.32. Suppose that the advertised window is 1 Mbyte long. If a sequence number is selected at random from the entire sequence number space, what is the probability that the sequence number falls inside the advertised window?

Solution:

If the sequence number field is 32 bits in length and the advertised window is 1Mbyte long, the probability that the sequence number falls inside the advertised window is:

$$P = (1 \times 10^6) / 2^{32} = 2.33 \times 10^{-4}$$

8.33. Explain the relationship between advertised window size, RTT, delay-bandwidth product, and the maximum achievable throughput in TCP.

Solutions follow questions:

(a) Plot the maximum achievable throughput versus delay-bandwidth product for an advertised window size of 65,535 bytes.

First consider delay-bandwidth product, $DBP = R*2t_p$. Here delay $2t_p$ is the propagation time that elapses from when a bit is sent by a source to the destination to when the bit can be returned back to the source. This is the minimum time that elapses from when a packet leaves a source to when the acknowledgment is received. The delay-bandwidth product *DBP* is then the number of bits (or bytes) that are in the network when the source transmits continuously at the maximum rate and when the bits return immediately back to the source.

The round-trip time *RTT* is the time that actually elapses from when a packet is sent to when its acknowledgment is received. *RTT* includes not only the propagation delay, but also queueing and

processing delays. The advertised window, *W,* places a limit on the amount of information that a source can have outstanding in the network.

Consider the time from when a byte leaves the source to when its acknowledgment is received (that is, consider a *RTT*). In that time, the source will have transmitted at most a window-full of bytes into the network. Therefore the window size divided by the *RTT* places a limit on the throughput *r*, that is, the rate at which information can be transmitted into the network: $r < W/RTT$.

The throughput cannot exceed the maximum bit rate $R = DBP/2t_p$ that is available for the source to transmit into the network. Therefore, the throughput increases as the window size is increased, but cannot exceed the bit rate *R*:

$$\text{Throughput} = r = \min\{R, W/RTT\} = \min\{DBP/2*t_p, W/RTT\}$$

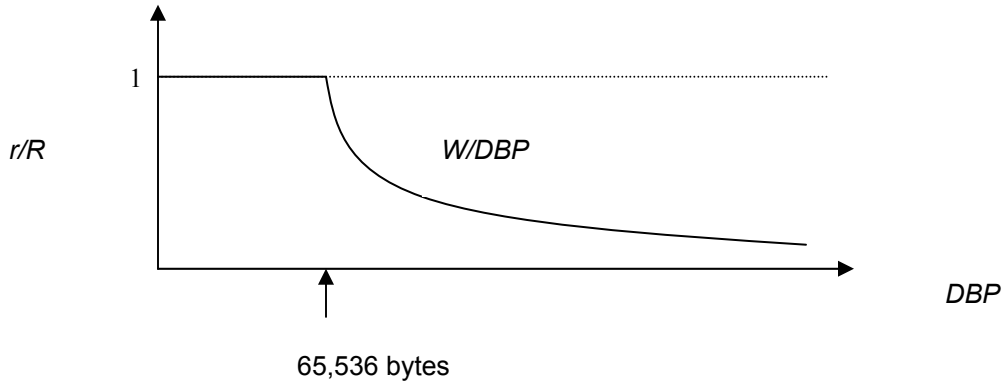
Suppose that the window size is less than the delay-bandwidth product. We then expect that the source cannot transmit at the maximum bit rate *R*. Indeed, we have that:

$$r < W/RTT < W/2t_p.$$

Therefore we have that:

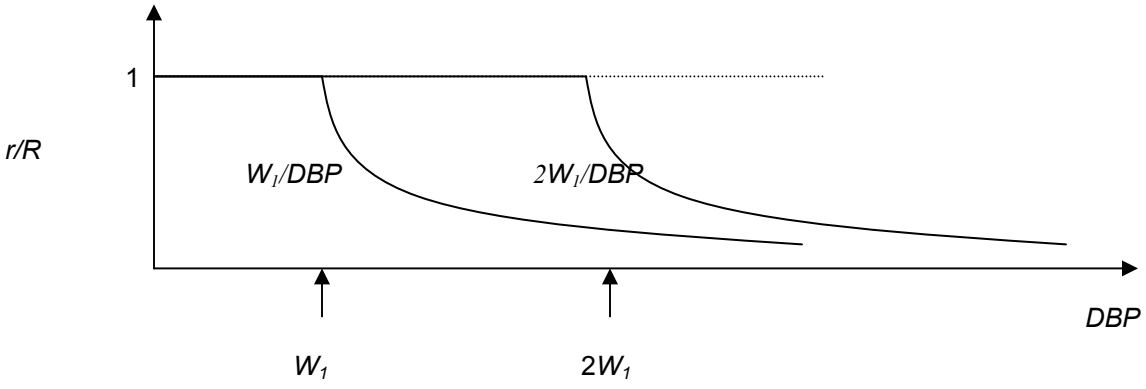
$$r/R < W/(R*2t_p) = W/DBP.$$

We conclude that the ratio of the maximum achievable throughput to *R* is less than the ratio of the window size to the *DBP*, as shown in the figure below.



(b) In the above plot include the maximum achievable throughput when the above window size is scaled up by a factor of $2K$, where $K = 4, 8, 12$.

The following figure shows the case where the window size is doubled.



(c) Place the following scenarios in the plot obtained in part (b): Ethernet with 1 Gbps and distance 100 meters; 2.4 Gbps and distance of 6000 km; satellite link with speed of 45 Mbps and RTT of 500 ms; 40 Gbps link with distance of 6000 km.

Case	***DBP* implied**
Ethernet R = 1 Gbps D = 100 m	$T_p = 100 / 2.5 \times 10^8$ $T_p = 4 \times 10^{-7}$ $DBP = 2 * T_p * R$ $DBP = 8 \times 10^{-7} * 1 \times 10^9$ $DBP = 8 \times 10^2$ ***DBP* = 800 bits**
Link R = 2.4 Gbps D = 6000 km	$T_p = 6 \times 10^3 / 2.5 \times 10^5$ $T_p = 2.4 \times 10^{-2}$ $DBP = 2 * T_p * R$ $DBP = 2 * 2.4 \times 10^{-2} * 2.4 \times 10^9$ $DBP = 11.52 \times 10^7$ ***DBP* = 115.2 Mbits (14.4 Mbytes)**
Satellite link R = 45 Mbps RTT = 500 ms (5×10^{-1} sec)	$DBP = RTT * R$ $DBP = 5 \times 10^{-1} * 45 \times 10^6$ $DBP = 225 \times 10^5$ bits ***DBP* = 22.5 Mbits (2.85Mbytes)**
link R = 40 Gbps D = 6000 km (6×10^6 m)	$T_p = 6 \times 10^3 / 2.5 \times 10^5$ $T_p = 2.4 \times 10^{-2}$ $DBP = 2 * T_p * R$ $DBP = 2 * 2.4 \times 10^{-2} * 40 \times 10^9$ $DBP = 192 \times 10^7$ ***DBP* = 1.92 Gbits (240 Mbytes)**

8.34. Consider the three-way handshake in TCP connection setup.

Solutions follow questions:

(a) Suppose that an old SYN segment from station A arrives at station B, requesting a TCP connection. Explain how the three-way handshake procedure ensures that the connection is rejected.

In a three-way handshake procedure, one must ensure the selection of the initial sequence number is always unique. If station B receives an old SYN segment from A, B will acknowledge the request based on the old sequence number. When A receives the acknowledgment segment from B, A will find out that B received a wrong sequence number. A will discard the acknowledgment packet and reset the connection.

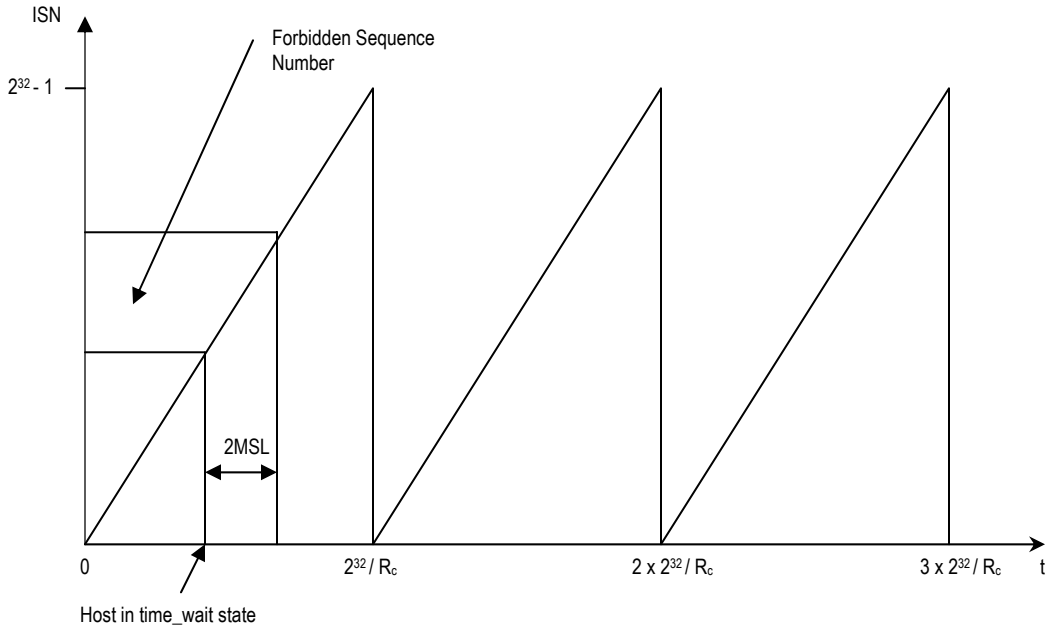
(b) Now suppose that an old SYN segment from station A arrives at station B, followed a bit later by an old ACK segment from A to a SYN segment from B. Is this connection request also rejected?

If an old SYN segment from A arrives at B, followed by an old ACK segment from A to a SYN segment from B, the connection will also be rejected. Initially, when B receives an old SYN segment, B will send a SYN segment with its own distinct sequence number set by itself. If B receives the old ACK from A, B will notify A that the connection is invalid since the old ACK sequence number does not match the sequence number previously defined by B. Therefore, the connection is rejected.

8.35. Suppose that the Initial Sequence Number (ISN) for a TCP connection is selected by taking the 32 low-order bits from a local clock.

Solutions follow questions:

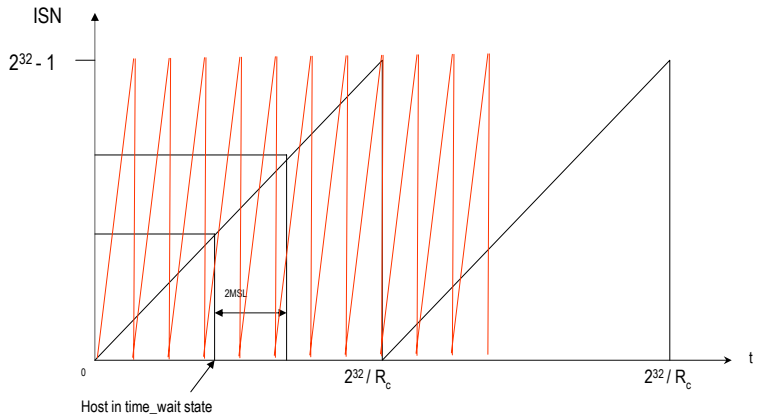
(a) Plot the ISN versus time assuming that the clock ticks forward once every $1/R_c$ seconds. Extend the plot so that the sequence numbers wrap around.



(b) To prevent old segments from disrupting a new connection, we forbid sequence numbers that fall in the range corresponding to 2MSL seconds prior to their use as an ISN. Show the range of forbidden sequence numbers versus time in the plot from part (a).

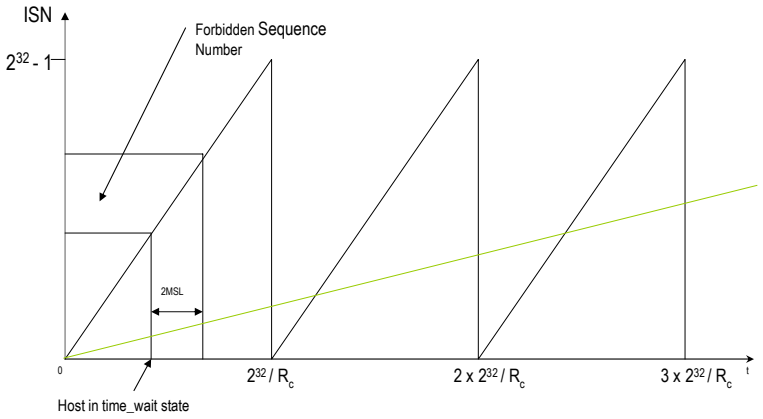
See above graph.

(c) Suppose that the transmitter sends bytes at an average rate $R > R_c$. Use the plot from part (b) to show what goes wrong.



If the transmitter sends data at an average rate $R > R_c$, the ISN will lag behind the transmitter's sequence number. In particular, if the source uses all the sequence numbers in less than 2 MSL, then all the sequence numbers would be forbidden as ISNs for the next connection.

(d) Now suppose that the connection is long-lived and that bytes are transmitted at a rate R that is much lower than R_c. Use the plot from part (b) to show what goes wrong. What can the transmitter do when it sees that this problem is about to happen?

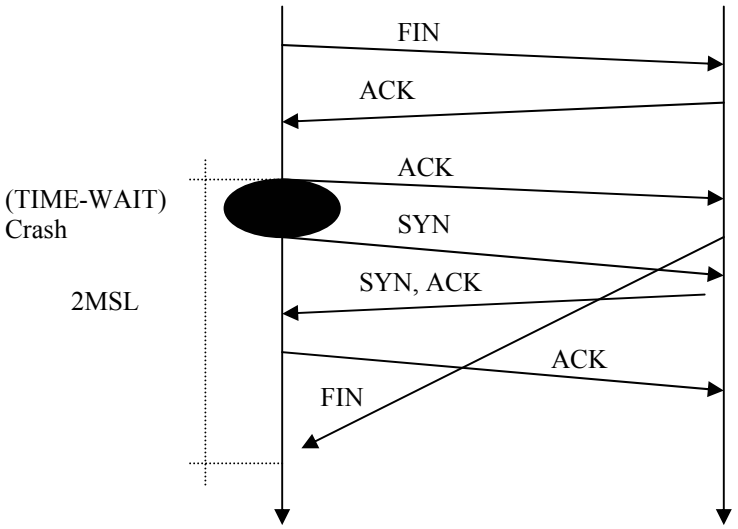


When the connection is long-lived and $R < R_c$, the ISN sequence number will wrap around at a much faster rate than the transmitter sequence number. When a new connection is established, it is possible for the ISN to be selected within the range of sequence numbers used by the slow connection.

8.36. Suppose that during the TCP connection closing procedure, a machine that is in the TIME_WAIT state crashes, reboots within MSL seconds, and immediately attempts to reestablish the connection using the same port numbers. Give an example that shows that delayed segments from the previous connections can cause problems. For this reason RFC 793 requires that for MSL seconds after rebooting TCP is not allowed to establish new connections.

Solution:

A delayed FIN from the earlier connection causes the new connection to be closed prematurely.



8.37. Are there any problems if the server in a TCP connection initiates an active close?

Solution:

As TCP is defined, no problems should arise if the server initiates an active close. Recall from Figure 8.36 that the side that does the active close (by issuing the first FIN segment) will enter the TIME_WAIT state. The side that does the passive close does not. Therefore, when the server does the active close, it will go into the TIME_WAIT state and hence will not be able to be restarted with its same (well-known) port number, because this is part of the parameters that are set aside during the 2MSL wait.

8.38. Use a network analyzer to capture the sequence of packets in a TCP connection. Analyze the contents of the segments that open and close the TCP connection. Estimate the rate at which information is transferred by examining the frame times and the TCP sequence numbers. Do the advertised windows change during the course of the connection?

Solution:

The following sequence of packet captures was obtained by connecting to www.yahoo.com using telnet. The TCP open and close can be observed at the beginning and end of the packet sequence. Advertised window sizes, acknowledgments, and sequence numbers are also shown. (The data was obtained from Ethereal using the print-to-file option.)

```
No. Time      Source         Destination  Protocol Info
 1 0.000000    Intel_c1:8e:0a  Broadcast       ARP    Who has 192.168.2.3?  Tell 192.168.2.2
 2 12.992506   192.168.2.18    192.168.2.1     DNS    Standard query A www.yahoo.com
 3 13.001008   192.168.2.1     192.168.2.18    DNS    Standard query response CNAME www.yahoo.akadns.net A 216.109.125.79 A
216.109.125.72 A 216.109.125.69 A 216.109.117.205 A 216.109.125.78 A 216.109.125.71 A 216.109.125.64 A 216.109.118.64
 4 13.001678   192.168.2.18    216.109.125.79  TCP    2498 > http [SYN] Seq=147142992 Ack=0 Win=8192 Len=0
 5 13.039151   216.109.125.79  192.168.2.18    TCP    http > 2498 [SYN, ACK] Seq=2183346772 Ack=147142993 Win=65535 Len=0
 6 13.039221   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142993 Ack=2183346773 Win=8760 Len=0
 7 18.472270   192.168.2.18    216.109.125.79  TCP    2498 > http [PSH, ACK] Seq=147142993 Ack=2183346773 Win=8760 Len=1
 8 18.600842   00000000.0001031dccf7 00000000.Broadcast    IPX SAP  General Query
 9 18.622879   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183346773 Ack=147142994 Win=65535 Len=0
10 18.734094   192.168.2.18    216.109.125.79  TCP    2498 > http [PSH, ACK] Seq=147142994 Ack=2183346773 Win=8760 Len=1
11 18.905241   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183346773 Ack=147142995 Win=65535 Len=0
12 18.924462   192.168.2.18    216.109.125.79  TCP    2498 > http [PSH, ACK] Seq=147142995 Ack=2183346773 Win=8760 Len=1
13 19.078556   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183346773 Ack=147142996 Win=65535 Len=0
14 19.244070   192.168.2.18    216.109.125.79  TCP    2498 > http [PSH, ACK] Seq=147142996 Ack=2183346773 Win=8760 Len=1
15 19.369356   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183346773 Ack=147142997 Win=65535 Len=0
16 19.692438   192.168.2.18    216.109.125.79  TCP    2498 > http [PSH, ACK] Seq=147142997 Ack=2183346773 Win=8760 Len=2
17 19.755278   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183346773 Ack=147142999 Win=65535 Len=1460
18 19.756467   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183348233 Ack=147142999 Win=65535 Len=1460
19 19.756515   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183349693 Win=8760 Len=0
20 19.758513   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183349693 Ack=147142999 Win=65535 Len=1460
21 19.843349   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183351153 Ack=147142999 Win=65535 Len=1460
22 19.843467   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183352613 Win=8760 Len=0
23 19.845612   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183352613 Ack=147142999 Win=65535 Len=1460
24 19.846719   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183354073 Ack=147142999 Win=65535 Len=1460
25 19.846764   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183355533 Win=8760 Len=0
26 19.899221   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183355533 Ack=147142999 Win=65535 Len=1460
27 19.900426   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183356993 Ack=147142999 Win=65535 Len=1460
28 19.900508   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183358453 Win=8760 Len=0
29 19.902280   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183358453 Ack=147142999 Win=65535 Len=1460
30 19.911624   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183359913 Ack=147142999 Win=65535 Len=1460
31 19.911745   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183361373 Win=8760 Len=0
32 19.916041   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183361373 Ack=147142999 Win=65535 Len=1460
33 19.917275   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183362833 Ack=147142999 Win=65535 Len=1460
34 19.917335   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183364293 Win=8760 Len=0
35 19.942667   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183364293 Ack=147142999 Win=65535 Len=1460
36 19.943834   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183365753 Ack=147142999 Win=65535 Len=1460
37 19.943914   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183367213 Win=8760 Len=0
38 19.959414   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183367213 Ack=147142999 Win=65535 Len=1460
39 19.962971   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183368673 Ack=147142999 Win=65535 Len=1460
40 19.963078   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183370133 Win=8760 Len=0
41 19.964934   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183370133 Ack=147142999 Win=65535 Len=1460
42 19.965690   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183371593 Ack=147142999 Win=65535 Len=1460
43 19.965733   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183373053 Win=8760 Len=0
44 19.987210   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183373053 Ack=147142999 Win=65535 Len=1460
45 19.994098   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183374513 Ack=147142999 Win=65535 Len=1460
46 19.994170   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183375973 Win=8760 Len=0
47 20.013318   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183375973 Ack=147142999 Win=65535 Len=1460
```

```
48 20.016425   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183378893 Ack=147142999 Win=65535 Len=1460
49 20.016538   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183377433 Win=8760 Len=0
50 20.018301   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183380353 Ack=147142999 Win=65535 Len=1460
51 20.018342   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183377433 Win=8760 Len=0
52 20.034934   216.109.125.79  192.168.2.18  TCP  http > 2498 [FIN, PSH, ACK] Seq=2183381813 Ack=147142999 Win=65535 Len=697
53 20.035032   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183377433 Win=8760 Len=0
54 21.275248   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183377433 Ack=147142999 Win=65535 Len=1460
55 21.275415   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183382510 Win=8760 Len=0
56 21.346468   216.109.125.79  192.168.2.18    TCP    http > 2498 [FIN, ACK] Seq=2183382510 Ack=147142999 Win=65535 Len=0
57 21.346585   192.168.2.18    216.109.125.79  TCP    2498 > http [ACK] Seq=147142999 Ack=2183382511 Win=8760 Len=0
58 23.239449   192.168.2.18    216.109.125.79  TCP    2498 > http [FIN, ACK] Seq=147142999 Ack=2183382511 Win=8760 Len=0
59 23.273437   216.109.125.79  192.168.2.18    TCP    http > 2498 [ACK] Seq=2183382511 Ack=147143000 Win=65535 Len=0
```

8.39. Devise an experiment to use a network analyzer to observe the congestion control behavior of TCP. How would you obtain Figure 8.37 empirically? Run the experiment and plot the results.

Solution:

Congestion involves the buildup of packets in a buffer and can be triggered by the sustained arrivals of packets from a high speed network, e.g. a LAN, to a router feeding a slow-speed network, e.g. a dialup modem to an ISP. Thus one way of observing congestion is to send a stream of packets from a home LAN onto the Internet via a dialup modem. Congestion can also occur when multiple users send packets on multiple inputs to the same output port on a router. Thus a second way of generating congestion is to have several machines simultaneously send a stream of packets to the Internet via a home router that connects to a DSL or cable modem. A packet capture tool such as Ethereal can be used to track the evolution of sequence numbers and segment retransmissions over time. However, the congestion window is controlled by TCP which operates in the kernel of the OS. To obtain traces of `cwnd` versus time a tool such as `tcpdump` with debug option, or TCP instrumentation tools (such as provided by the Web100 project for Linux) need to be used.

8.40. A fast typist can do 100 words a minute, and each word has an average of 6 characters. Demonstrate Nagle's algorithm by showing the sequence of TCP segment exchanges between a client, with input from our fast typist, and a server. Indicate how many characters are contained in each segment sent from the client. Consider the following two cases:

Solutions follow questions:

The typist types 100 words per minutes, averaging 6 characters per word. This is equivalent to 600 characters per minute or 10 characters per second. Therefore, the typist can type a character every 100 ms.

(a) The client and server are in the same LAN and the RTT is 20 ms.

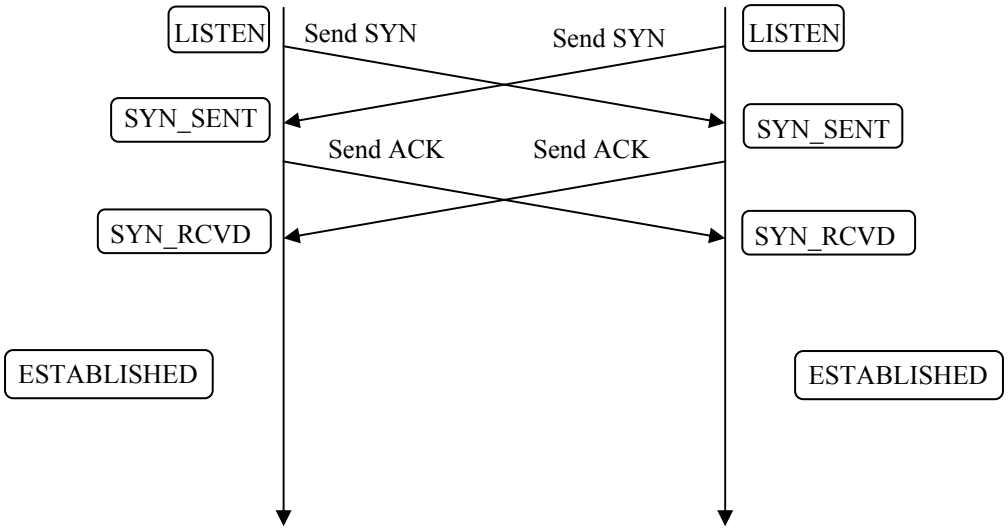
In this case Nagle's Algorithm is not activated since acknowledgments arrive before the next character is typed. Each client segment is 41 bytes long assuming IP and TCP headers are 20 bytes each.

(b) The client and server are connected across a WAN and the RTT is 100 ms.

In this case one or two characters are typed before an acknowledgment is received. Therefore, segments are either 41 or 42 bytes long.

8.41. *Simultaneous Open*. The TCP state transition diagram allows for the case where the two stations issue a SYN segment at nearly the same time. Draw the sequence of segment exchanges and use Figure 8.36 to show the sequence of states that are followed by the two stations in this case.

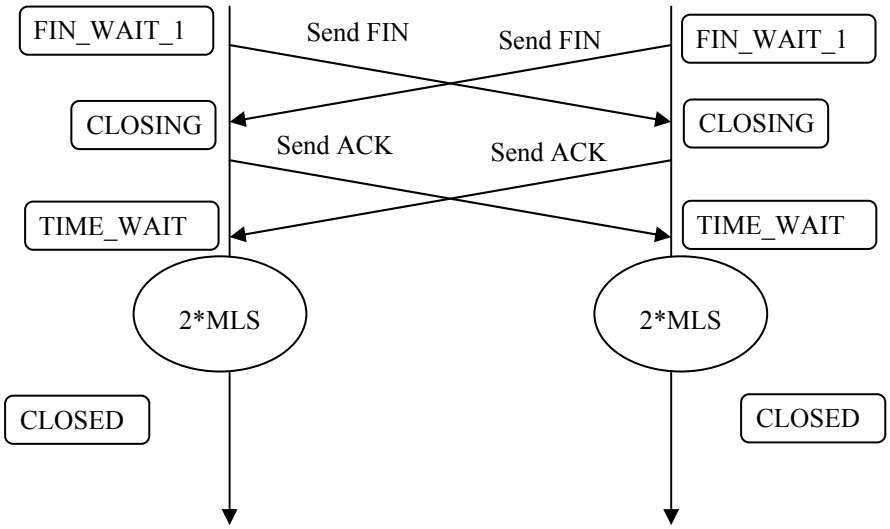
Solution:



8.42. *Simultaneous Close*. The TCP state transition diagram allows for the case where the two stations issue a FIN segment at nearly the same time. Draw the sequence of segment exchanges and use Figure 8.36 to show the sequence of states that are followed by the two stations in this case.

Solution:

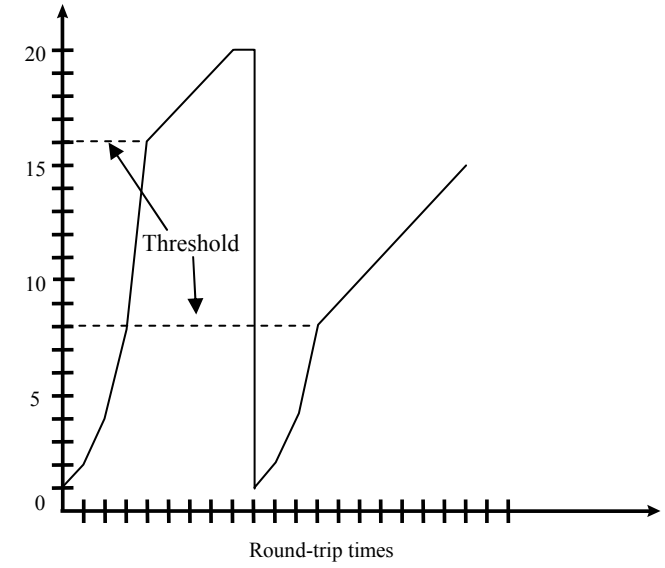
The sequence of the state transition will be the same for two hosts. After FIN sent, it moves from ESTABLISHED to FIN_WAIT_1. Once the host receives the FIN from other host, it sends an ACK and move from FIN_WAIT_1 to CLOSING. Finally, when the host received ACK from each other, it moves from CLOSING to TIME_WAIT. After 2MSL, both hosts transit back to CLOSED state.

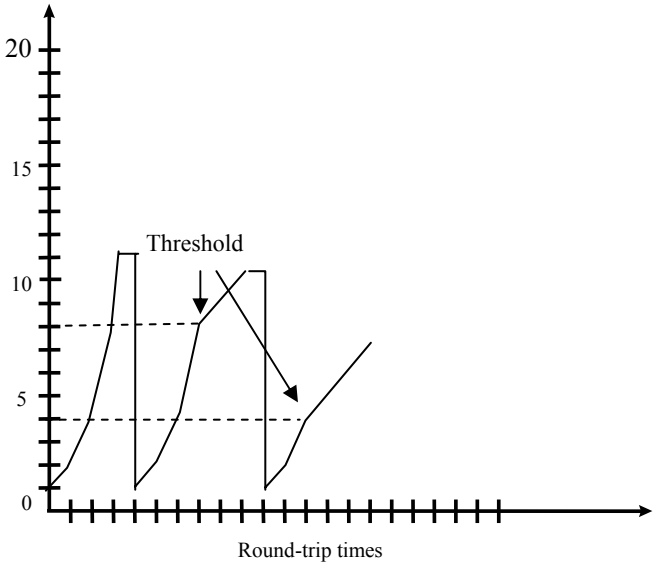


8.43. Suppose that a TCP source (with unlimited amount of information to transmit) begins transmitting onto a link that has 1 Mbps in available bandwidth. Sketch congestion window versus time trajectory. Now suppose that another TCP source (also with unlimited amount of information to transmit) begins transmitting over the same link. Sketch the congestion window versus the time for the initial source.

Solution:

Initially the TCP source has about 1 Mbps of available bandwidth, so its congestion avoidance behavior will begin as the rate approaches 1 Mbps. When the available bandwidth drops to 500 kbps, the congestion window behavior will begin sooner. The following two figures show the corresponding congestion window trajectories.





8.44. What is the maximum width of a RIP network?

Solution:

The maximum width of an RIP network is 15 nodes. Since the maximum number of hops in RIP network is 15, node 16 represents infinity.

8.45. Let's consider the bandwidth consumption of the RIP protocol.

Solutions follow questions:

(a) Estimate the number of messages exchanged per unit time by RIP.

A router implementing RIP sends an update message every 30 seconds. Assume that the typical node has D neighbors and that the number of nodes in the network is N. The number of message exchanges is then $ND/30$ messages per second.

(b) Estimate the size of the messages exchanged as a function of the size of the RIP network.

A RIP message consists of a four-byte header plus 20-byte per entry and up to 25 entries per message. Therefore, each node will send out a message of at most $20(N - 1) + 4$ bytes every 30 seconds.

(c) Estimate the bandwidth consumption of a RIP network.

The bandwidth consumption is at most $[ND [20 (N - 1) + 4]] / 30$ bytes per second.

8.46. RIP runs over UDP, OSPF runs over IP, and BGP runs over TCP. Compare the merits of operating a routing protocol over TCP, UDP, IP.

Solution:

RIP is a protocol in which the routers operate in a highly distributed fashion following a distance vector algorithm. Message exchanges occur only between neighbors and at periodic intervals or triggered by specific events. UDP is suitable for the exchange of individual messages but without delivery guarantees. However the operation of RIP makes allowances for the lack of such guarantees.

OSPF relies on the use of a reliable flooding procedure to distribute link-state information to all the routers. This reliable flooding procedure requires close coordination with the operation of the routers, and hence direct operation over IP instead of over a transport layer protocol is preferred.

BGP peers exchange the entire BGP routing table initially and incremental updates are sent instead of periodic updates to reduce the bandwidth consumption. A small periodic KEEPALIVE message is used to determine that the BGP peers are alive. Reliable delivery of the routing information is required to minimize the bandwidth consumption. For this reason, TCP is chosen to provide the reliable delivery required by BGP.

8.47. Compare RIP and OSPF with respect to convergence time and the number of messages exchanged under several trigger conditions, that is, link failure, node failure, link coming up.

Solution:

Link failure – OSPF has a faster convergence time than RIP. When a link fails, the corresponding OSPF routers send a link-state update to all peer routers. The peer routers then update their databases. This process allows the link failure state information to be propagated to other routers

quickly. In contrast, when a link fails in RIP, the corresponding RIP router updates its own distance vector and sends the link update message to its neighbor. The neighboring nodes update their own routing tables, calculate the new distance vector values and send the updated distance vector to their neighbors. The processing and distribution overhead in RIP slows down the convergence time and routing loops may be created while the algorithm is converging. The faster convergence time of OSPF is at the cost of flooding the network with update messages. RIP is based on the exchange of messages between adjacent nodes only.

Node failure – OSPF converges faster than RIP. OSPF sends a HELLO packet every 10 seconds, compared to RIP's update message every 30 seconds. In case of node failure, OSPF can detect failure of a node within the range of 10 seconds. On the other hand, RIP requires a period of 180 seconds (worst case) to detect a node failure. The grace period of 180 seconds in RIP is due to the fact that RIP is running over UDP which cannot be relied upon to deliver messages consistently. Again, OSPF requires more message exchanges (HELLO packet every 10 seconds) than RIP (update message every 30 seconds).

Link coming up – OSPF converges faster than RIP. When a link is coming up, the OSPF routers attached to this link start sending HELLO packets. Next, these router pairs exchange link-state database description packets and send link-state request packets for those LSA headers that are not in their respective link-state databases. After the databases are updated and synchronized, these OSPF routers send the updated database description packet to their own neighbors and execute the routing algorithms to find out the shortest path. In contrast, a RIP router must first perform a distance vector routing calculation. Next, the router sends the update distance vector its own neighbor. The time it takes for RIP to distribute the new link information is slower than OSPF due to the processing overhead imposed on each RIP router.

8.48. Consider the OSPF protocol.

Solutions follow questions:

(a) Explain how OSPF operates in an autonomous system that has not defined areas.

When OSPF operates in an AS that has no defined area, the broadcastpackets (link-state update and HELLO) must flow all over the AS. If the network (AS) is too large, this approach consumes too many network resources and it does not scale well. Also, each OSPF router link-state database and routing table size increase dramatically when the number of nodes within the AS increases.

(b) Explain how the notion of area reduces the amount of routing traffic exchanged.

When area is used in OSPF, the number of routers within an area decreases as compared to the previous case. The broadcast packets only need to flow within an area. Therefore, the network traffic is reduced and utilization of the network resources is increased. Also, each router has a smaller link-state database and routing table.

(c) Is the notion of area related to subnetting? Explain. What happens if all addresses in an area have the same prefix?

The idea behind subnetting is to add another hierarchical level within a particular class of IP address. The idea behind an area is similar in that it involves the use of hierarchy to simplify routing, but there is no direct relationship between area and subnetting. In an area, multiple classes of IP addresses can exist. Within each class of IP addresses, different ways of subnetting can be done. Therefore, subnetting is just another level of hierarchical level to allow a network administrator to better manage a particular class of IP address.

If all addresses in an area have the same prefix, the area border router (ABR) will advertise or exchange a simple summary to indicate all the addresses with this prefix belong to this particular

area. The ABR contains the whole network topology and performs appropriate routing within its area upon the transmission between different areas.

8.49. Assume that there are N routers in the network and that every router has m neighbors.

Solutions follow questions:

(a) Estimate the amount of memory required to store the information used by the distance-vector routing.

Each node needs the distance to each neighbor and the distance from each neighbor to all destinations which is $m(N-1)$ entries. Assuming E bytes for each entry the amount of memory is $m(N-1)E$.

(b) Estimate the amount of memory required to store the information by the link-state algorithm.

Each node needs the information for all links across the network. Each node is connected to m links and there are N nodes in the network. Therefore the total number of links in the network is $(1/2)Nm$ entries. Assuming E bytes per entry the amount of memory is $(1/2)mNE$.

8.50. Suppose a network uses distance-vector routing. What happens if the router sends a distance vector with all 0s?

Solution:

A distance vector with all zeros means that the node has distance 0 to all other nodes. This will prompt all neighbors to route all their packets through the given router. Eventually all packets in the network will be routed to this router, resulting in what can be characterized as a "black hole."

8.51. Suppose a network uses link-state routing. Explain what happens if:

Solutions follow questions:

(a) The router fails to claim a link that is attached to it.

The link will be eventually omitted from the routing tables and as a result will not be utilized. This will result in some routing decisions that are not the shortest (optimal) paths. Loss of connectivity is also possible. Moreover, a routing loop may occur due to confusion among routers as to which links they are attached to.

(b) The router claims to have a link that does not exist.

If a router claims to have a link that does not exist, it reports false topology information to other routers within the network. If this non-existing link belongs to a shortest path, all the packets that are sent via this path will be lost. This will affect the network performance severely.

8.52. Consider a broadcast network that has n OSPF routers.

Solutions follow questions:

(a) Estimate the number of database exchanges required to synchronize routing databases.

In the worst case, for an n-router OSPF network, n^2 database exchanges are required to synchronize all of the routers databases.

(b) What is the number of database exchanges after a designated router is introduced into the network?

If a designated router is introduced, the number of database exchanges will reduce to n. This is because each router must only communicate with the designated router to obtain link-state information.

(c) Why is the backup designated router introduced? What is the resulting number of database exchanges?

If the primary designated router fails, the information exchange will function properly when the backup designated router is introduced. This mechanism is designed to protect against failure and to provide fast recovery. The number of database exchanges is $2n$ since every router must update both the primary and backup designated router with link-state information.

8.53. Suppose n OSPF routers are connected to a non-broadcast multi-access network, for example, ATM.

Solutions follow questions:

(a) How many virtual circuits are required to provide the required full connectivity?

For a full connectivity, ATM requires an $n(n-1)$ virtual circuit connections.

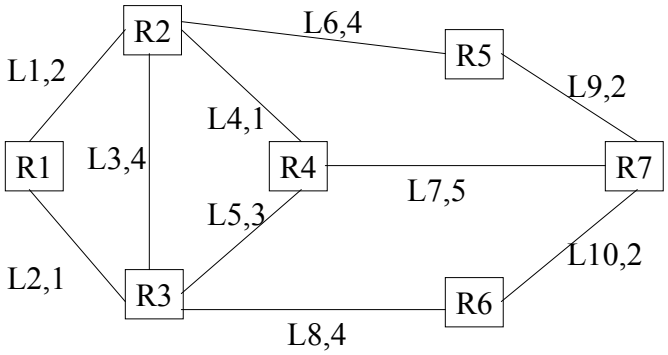
(b) Does OSPF function correctly if a virtual circuit fails?

Yes, OSPF will function correctly if one of the VC fails. This is because the failed VC becomes a failed link in the IP layer and the packet can re-route via a different link (VC) to reach the destination.

(c) Is the number of required virtual circuits reduced if point-to-multipoint virtual circuits are available?

If point-to-multipoint VCs are used, the number of VCs required will be n.

8.54. The figure below shows seven routers connected with links that have the indicated costs. Use the Hello protocol to show how the routers develop the same topology database for the network.



Solution:

In the first stage, every router sends a HELLO packet to all the links that it is attached to.

- R1 sends HELLO packets to R2 and R3 via L1 and L2 respectively.
- R2 sends HELLO packets to R1, R3, R4 and R5 via L1, L3, L4 and L6 respectively.
- R3 sends HELLO packets to R1, R2, R4 and R6 via L2, L3, L5 and L8 respectively.
- R4 sends HELLO packets to R2, R3 and R7 via L4, L5 and L7 respectively.

- R5 sends HELLO packets to R2 and R7 via L6 and L9 respectively.
- R6 sends HELLO packets to R3 and R7 via L8 and L10 respectively.
- R7 sends HELLO packets to R4, R5 and R6 via L7, L9 and L10 respectively.

In the next stage, when a router receives a HELLO packet, it replies with a HELLO packet containing the router ID of each neighbor it has currently seen. For example, when R1 receives HELLO from R2, it will send a HELLO packet to R2. The HELLO packet will inform R2 that the neighbors R1 currently sees are R2 and R3, given that R3's HELLO packet has been received by R1 before R2's HELLO packet.

Once all the routers know their neighbor routers, routers exchange Database Description packets to check if their link-state databases are in agreement. If the databases are not in agreement, the routers exchange link-state request and update packets to synchronize their databases.

8.55. Consider the exchange of Hello messages in OSPF.

Solutions follow questions:

(a) Estimate the number of Hello messages exchanged per unit time.

A HELLO packet is sent periodically every 10 seconds. Assume the number of links within a network is L. The total number of HELLO packet exchanged per 10-second unit time is $2L$.

(b) Estimate the size of the Hello messages.

The size of the HELLO packet is 20 bytes for the header plus an entry for each neighbor the router sees. Therefore the size depends on the number of neighbors a router is connected to. Assume the average degree for a router is m. Each router ID is 4 bytes. The size of a HELLO packet = $20 + 4m$ bytes.

(c) Estimate the bandwidth consumed by Hello messages.

The bandwidth consumed by the HELLO packets is $2L$ [(20 x $4m$) x 8] / 10 bps.

8.56. Consider the notion of adjacency in OSPF.

Solution:

(a) Explain why it is essential that all adjacent routers be synchronized.

Adjacent routers must be synchronized to ensure that routers use the same topology when running the routing algorithm, and ultimately to avoid loops and unnecessary packet dropping in the network.

(b) Explain why it is sufficient that all adjacent routers be synchronized, that is, it is not necessary that all pairs of routers be synchronized.

For every adjacent router pair, the routers that are connected to this particular adjacent router pair must also be synchronized (according to the rule of adjacency). As a result, all the routers within the network will eventually synchronize.

8.57. Consider the robustness of OSPF.

Solutions follow questions:

(a) Explain how the LSA checksum provides robustness in the OSPF protocol.

The LSA checksum provides error detection for the entire content of the LSA except the link-state age. It gives a second level of error detection to ensure each individual LSA entry carries correct information. If an error is detected in an LSA entry, the router can discard the specific entry while continuing to use other entries in the LSA update packet.

(b) An OSPF router increments the LS Age each time it inserts the LSA into a link-state update packet. Explain how this protects against an LSA that is caught in a loop.

The router increments the LS age each time it inserts the LSA into a link-state update packet. Therefore, whenever a router receives an LSA that is caught in a loop, the router can verify the validity of the LSA with its current LS age. If the LSA is too old, the router will ignore the LSA packet.

(c) OSPF defines a minimum LS update interval of 5 seconds. Explain why.

OSPF uses flooding to distribute the LSA packets. If the LS update interval is too short, the router may not be able to distinguish the order of the LSA packets it receives. A too-short update interval may consume a lot of processing power just to handle the link-state database and routing algorithm. Therefore, choosing the interval of 5 seconds will allow router to exchange LSA twice with every HELLO packet interval. This would allow the network to respond faster in case of any link-state change while keeping the processing overhead low.

8.58. Assume that for OSPF updates occur every 30 minutes, an update packet can carry three LSAs, and each LSA is 36 bytes long. Estimate the bandwidth used in advertising one LSA.

Solution:

An OSPF update occurs every 30 minutes = 1800 seconds. An update can carry three LSAs. Each LSA is 36 bytes long. The OSPF common header is 24 bytes. The size of the LSA field in link-state update consumes 4 bytes. Therefore, if OSPF updates occur every 1800 seconds containing 3 LSAs, the bandwidth for advertising one LSA is equal to ([(24 + 4 + 3 x 36) x 8] / 1800) x (1/3) = 0.201 bits/second.

8.59. Identify elements where OSPF and BGP are similar and elements where they differ. Explain the reasons for similarity and difference.

Solution:

Similarities - OSPF and BGP are used to exchange routing information including active routes, inactive routes and error conditions within the network in general. OSPF handles the information exchange within an AS while the BGP handles the information exchange between different ASs. Both OSPF and BGP allow a router to construct the network topology based on the link-state and path vector information respectively. OSPF and BGP use HELLO and KEEPALIVE messages respectively to determine the presence of peers.

Differences - OSPF runs over IP while BGP runs over a TCP connection. OSPF is a link-state protocol and BGP uses a path-vector protocol. BGP can enforce policy by affecting the selection of different paths to a destination and by controlling the redistribution of routing information. BGP only requires an incremental update of the database. However, OSPF requires a periodic update (refresh every 30 minutes).

8.60. Discuss the OSPF alternate routing capability for the following cases:

Solutions follow questions:

(a) Traffic engineering, that is, the control of traffic flows in the network.

In OSPF, the link-state metric exchange between routers can represent the volume of flow of traffic for each individual link. Therefore, each router can use these link-state metrics (traffic flow) to determine the best route for a particular IP packet given its type-of-service (TOS) field. However, the processing required in each router increases dramatically. One possible solution is to make use of explicit routing by pre-establishing a path that is suitable for each class of service.

(b) QoS routing, that is, the identification of paths that meet certain QoS requirements.

OSPF can use a QoS parameter or metric for each link to determine the optimal path for QoS routing. All routers within the network can use a set of routing algorithms to send packets along a particular route that satisfies a certain QoS requirement, e.g. delay, bandwidth, or bit-error rate requirement.

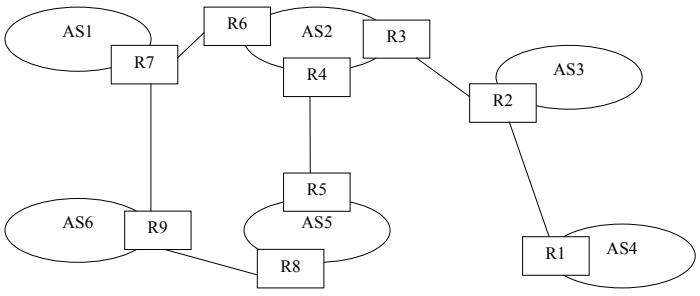
(c) Cost-sensitive routing, that is, the identification of paths that meet certain price constraints.

In cost-sensitive routing, the link-state metric can represent the cost of each link. The router determines the best route that meets a certain price constraint for a particular packet. An ingress router needs to have the information to determine the minimum cost paths to all destinations.

(d) Differential security routing, the identification of paths that provide different levels of security.

In differential security routing, the link-state metric involves identification of the security level of each link. All routers need to come up with a standard definition of a security metric for links. The router identifies the best route to satisfy each individual packet's security requirement.

8.61. Consider the autonomous systems and BGP routers in the following figure.



Solutions follow questions:

(a) Suppose that a certain network prefix belongs to AS4. Over which router pairs will the route to the given network be advertised?

If a route in AS4 requires advertising, R1 will run an eBGP protocol and establish a TCP connection to R2 to exchange the routing information. R2 will relay the message to R3 through a TCP connection based on eBGP. R3 forwards this information to R4 and R6 via iBGP. R6 advertises the information to R7 and R4 advertises to R5 via eBGP. R7advertises the information to R9 via eBGP. Finally, R5 sends the information to R8 via iBGP and R9 sends the information also to R8 via eBGP.

(b) Now suppose the link between R1 and R2 fails. Explain how a loop among AS1, AS2, AS6, and AS5 is avoided.

With the use of path-vector routing, a loop occurs whenever a BGP router receives an update message with an AS path attribute that contains its own AS number. The BGP router ignores the route and discards the information immediately.

(c) Suppose that R9 is configured to prefer AS1 as transit and R6 is configured to prefer AS1 as transit. Explain how BGP handles this situation.

The use of path-vector routing will again allow routers R9 and R6 to avoid the potential loop that can result from these preferences.

8.62. Why does BGP not exchange routing information periodically like RIP?

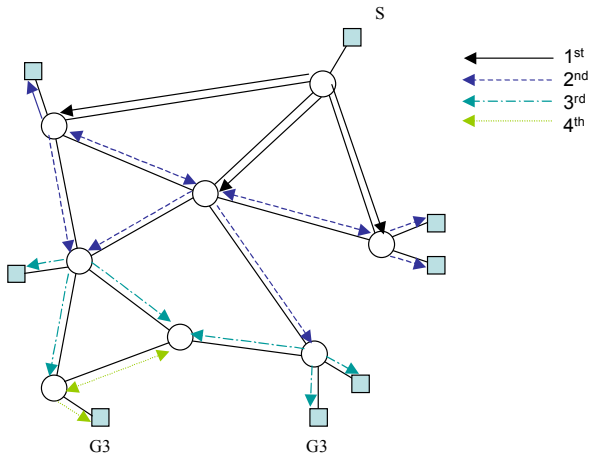
Solution:

BGP runs over TCP, which provides reliable service and simplifies BGP significantly by allowing the protocol to assume the availability of reliable information. In contrast, RIP runs over UDP, which is an unreliable protocol that may experience packet loss. RIP therefore requires periodic exchange of routing information to ensure routing information is correct and up-to-date.

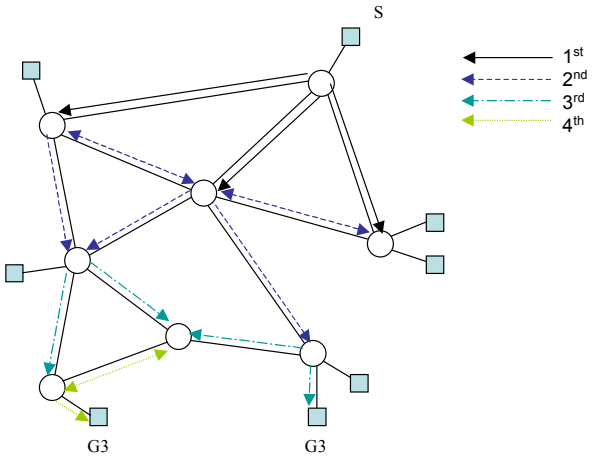
8.63. Consider the network shown in Figure 8.60. Suppose that a source connected to router 7 wishes to send information to multicast group G3.

Solutions follow questions:

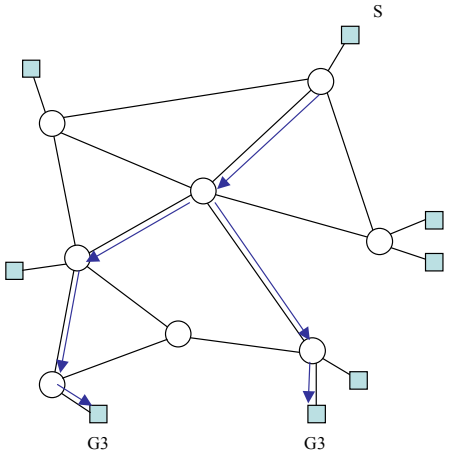
(a) Find the set of paths that are obtained from reverse-path broadcasting.



(b) Repeat for truncated reverse-path broadcasting.



(c) Repeat for reverse-path multicasting.



8.64. Discuss the operation of the reverse-path multicasting in the following two cases:

Solutions follow questions:

(a) The membership in the multicast group in the network is dense.

If the membership in the multicast group in the network is dense, then there will not be extensive use of prune messages and operation will be similar to reverse-path broadcasting. The utilization of bandwidth for multicasting packets will be relatively efficient.

(b) The membership in the multicast group in the network is sparse.

If the membership in the multicast group in the network is sparse, there will be extensive pruning of paths. The multicast tree will be “thin” relative to the overall network, and the utilization of bandwidth for multicasting packets will be inefficient.

8.65. Suppose an ISP has 1000 customers and that at any time during the busiest hour of the day, the probability that a particular user requires service is .20. The ISP uses DHCP. Is a class C address enough so that the probability is less than 1 percent that there is no IP address available when a customer places a request?

Solution:

A class C address space provides 254 IP addresses. The distribution of the number of active customers is given by a binomial distribution with parameters n = 1000 customers and p =.20. Let X be the number of active customers. The probability that $X = k$ customers are active at a given time is given by:

$$P[X = k] = \binom{1000}{k}(0.2)^k (0.8)^{1000-k}$$

The random variable X has mean equal to $m = np$ =1000*0.2 = 200 and variance given by $\sigma^2 = np(1–p) = 1000(.2)(.8) = 160$.

For large values of n, the binomial distribution can be approximated by a Gaussian distribution with the same mean and variance. In particular we have that:

$$P[X > k] \approx P[X_{Gaus\sin} > k] = \frac{1}{\sqrt{2\pi\sigma^2}} \int_k^\infty e^{-(x-m)^2/2\sigma^2} dx = \frac{1}{\sqrt{2\pi}} \int_{\frac{k-m}{\sigma}}^\infty e^{-x^2/2} dx$$

The value of the above integral at the point $(k – m)/\sigma$ = (254-200)/40 = 1.35 is 0.088=8.8%. This is the probability that the ISP does not have enough IP addresses to serve customer connection requests.

8.66. Compare mobile IP with the procedures used by cellular telephone networks (Chapter 4) to handle roaming users.

Solutions follow questions:

(a) Which cellular network components provide the functions of the home and foreign agent?

The home location register in cellular network provides the equivalent functions of the home agent in mobile IP network. The visitor location register in cellular network provides the functions of the foreign agent in mobile IP network.

(b) Is the handling of mobility affected by whether the transfer service is connectionless or connection-oriented?

The basic handling of the mobile users is the same in terms of the operation of the home location and visitor location registers. The manner in which information is transferred is of course different given that one is connectionless and the other connection-oriented.

8.67. Consider a user that can be in several places (home networks) at different times. Suppose that the home networks of a user contain registration servers where users send updates of their location at a given time.

Solutions follows questions:

(a) Explain how a client process in a given end system can find out the location of a given user in order to establish a connection, for example, Internet telephone, at a given point in time.

A client process can locate a given user by sending a request to a designated registration server. This registration server contains the most up-to-date location of a given user. Once the client process finds the location, it can establish a connection based on the information. The way to locate the registration server can be predefined by the network administrator or based on the request IP address prefix. In the latter approach, the client can use the address prefix of the given user to locate the corresponding registration server.

(b) Suppose that proxy servers are available, whose function is to redirect location requests to another server that has more precise location information about the callee. For example, a university might have such a server, which redirects requests for prof@university.edu to departmental servers. Explain how a location request for engineer@home.com might be redirected to a.prof@ece.university.edu.

When a location request for engineer@home.com is sent to the proxy server, the server will forward the request to the home.com server. The home.com server can send a redirect message informing the proxy server that the engineer@home.com can be redirected to a.prof@ece.university.edu. Therefore, whenever the client sends messages to the engineer@home.com via the proxy server, the server will automatically redirect the message to a.prof@ece.university.edu.