

1. Let $g_1(x) = x + 1$ and let $g_2(x) = x^3 + x^2 + 1$. Consider the information bits (1,1,0,1,1,1).

a. Find the codeword corresponding to these information bits if $g_1(x)$ is used as the generating polynomial.

b. Find the codeword corresponding to these information bits if $g_2(x)$ is used as the generating polynomial.

a.

$G_1(x)$ is used as the generating polynomial

$$G_1(x) = x + 1 = 11 \rightarrow n=2$$

Step 1: Add $n-1$ bit 0 to the information bits \Rightarrow Information bits: 1101110

Step 2:

$$\begin{array}{r}
 100101 \\
 11 \overline{) 1101110} \\
 \underline{11} \\
 00011 \\
 \underline{11} \\
 0010 \\
 \underline{11} \\
 01
 \end{array}$$

$$R(x) = 01$$

$$Q(x) = 100101$$

Step 3:

Add $R(x)$ to the information bits

-> Codeword: 1101111

b.

$G_2(x)$ is used as the generating polynomial

$$G_2(x) = x^3 + x^2 + 1 = 1101 \rightarrow n=4$$

Step 1: Add $n-1$ bit 0 to the information bits

-> Information bits: 110111000

Step 2:

$$\begin{array}{r} 100011 \\ 1101 \overline{) 110111000} \\ \underline{1101} \\ 00001100 \\ \underline{1101} \\ 00010 \end{array}$$

$$R(x) = 010$$

$$Q(x) = 100011$$

Step 3:

Add $R(x)$ to the information bits

-> Codeword: 110111010

1. Let $g_1(x) = x + 1$ and let $g_2(x) = x^3 + x^2 + 1$. Consider the information bits (1,1,0,1,1,0).

a. Find the codeword corresponding to these information bits if $g_1(x)$ is used as the generating polynomial.

b. Find the codeword corresponding to these information bits if $g_2(x)$ is used as the generating polynomial.

a.

$G_1(x)$ is used as the generating polynomial

$$G_1(x) = x + 1 = 11 \rightarrow n=2$$

Step 1:

Add $n-1$ bit 0 to the information bits \Rightarrow Information bits: 1101100

Step 2:

$$\begin{array}{r} 100100 \\ 11 \overline{) 1101100} \\ \underline{11} \\ 00011 \\ \underline{11} \\ 0000 \\ \underline{00} \\ 0 \end{array}$$

$$R(x) = 0$$

$$Q(x) = 100100$$

Step 3:

Add $R(x)$ to the information bits

-> Codeword: 1101100

b.

$G_2(x)$ is used as the generating polynomial

$$G_2(x) = x^3 + x^2 + 1 = 1101 \rightarrow n=4$$

Step 1:

Add $n-1$ bit 0 to the information bits

-> Information bits: 110110000

Step 2:

$$\begin{array}{r}
 100011 \\
 1101 \overline{) 110110000} \\
 \underline{1101} \\
 00001000 \\
 \underline{1101} \\
 01010 \\
 \underline{1101} \\
 0000
 \end{array}$$

0111

$$R(x) = 111$$

$$Q(x) = 100011$$

Step 3:

Add $R(x)$ to the information bits

->Codeword: 110110111

2. Consider the 7-bit generator, $G=10111$, and suppose that D has the value 1010100001. What is the value of R ? Show your all steps to have result.

$$G = 10111$$

$$D = 1010100001$$

The polynomial expression of G :

$$= x^4 + x^2 + x^1 + x^0$$

Here, the degree of the expression is 4. So, $r = 4$.

Thus, $D + r$ becomes 10101000010000

Calculating the value of R :

$$\begin{array}{r} \overline{1001011001} \\ 10111 \overline{10101000010000} \\ \underline{10111} \\ 010000 \\ \underline{10111} \\ 0011101 \end{array}$$

$$\begin{array}{r}
 10111 \\
 \hline
 010100 \\
 10111 \\
 \hline
 00011000 \\
 10111 \\
 \hline
 01111
 \end{array}$$

Therefore, the value of R = 1111

2. Consider the 7-bit generator, G=10011, and suppose that D has the value 1010101010. What is the value of R? Show your all steps to have result.

G = 10011

D = 1010101010

The polynomial expression of G:

$$= x^4 + x^1 + x^0$$

Here, the degree of the expression is 4. So, r = 4.

Thus, D + r becomes 10101010100000.

Calculating the value of R:

$$\begin{array}{r}
 10111011100 \\
 10011 \overline{) 10101010100000} \\
 \underline{10011} \\
 11001 \\
 \underline{10011} \\
 010100 \\
 \underline{10011} \\
 001110
 \end{array}$$

$$\begin{array}{r}
 10011 \\
 \hline
 01101\mathbf{0} \\
 10011 \\
 \hline
 01001\mathbf{0} \\
 10011 \\
 01\mathbf{00}
 \end{array}$$

Therefore, the value of $R = 0100$

3. Consider the following network Figure 1. With the indicated link costs, use Dijkstra's shortest-path algorithm to compute the shortest path from u to all network nodes. Show how the algorithm works by computing a table.

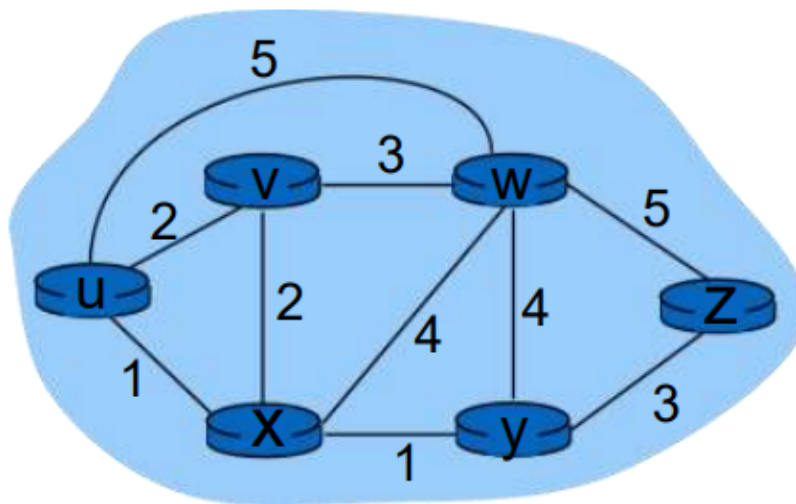


Figure 1

	u	v	w	x	y	z
u	-	2, u	5, u	1, u	~	~
u, x	-	2, u	5, u	-	2, x	~

u, x, v	-	-	5, u	-	2, x	~
u, x, v, y	-	-	5, u	-	-	5, y
u, x, v, y, w	-	-	-	-	-	5, y

u, v: 2

u, w: 5

u, x: 1

u, x, y: 2

u, x, y, z: 5

4. A router has the following CIDR entries in its routing table:

Address/mask Next hop

135.46.56.0/22 Interface 0

135.46.60.0/22 Interface 1

192.53.40.0 /23 Router 1

default Router 2

(a) What does the router do if a packet with an IP address 135.46.63.10 arrives?

(b) What does the router do if a packet with an IP address 135.46.57.14 arrives?

Ans:

(a) 135.46.63.10

Taking the first 22 bits of 135.46.63.10 as network address, we have 135.46.60.0.

The bit pattern of 135.46.63.10 is 10000111.00101110.00111111.00001010

When we perform the bit and operation with 22 leading bit 1s and 10 bit 0s, it is equivalent of making the last 10 bit zero.

We get the following network address bit pattern:

10000111.00101110.00111100.00000000.

The first two bytes are not changed.

The 3rd type changes from 63 to 60 while the 4th byte become zero.

Match with network address in the routing table.

The 2nd row matches. The router will forward the packet to Interface 1.

The bit pattern of 135.46.63.10 is: 10000111.00101110.00111111.00001010

Taking the first 22 bits of the above IP as network address (it is equivalent of making the last 10 bit zero), we have: 10000111.00101110.00111100.00000000 = 135.46.60.10

-> It matches the network address of the 2nd row. The packet will be forwarded to interface 1.

(b) 135.46.57.14

Taking the first 22 bits of the above IP address as network address, we have 135.45.56.0. It matches the network address of the first row.

The packet will be forwarded to Interface 0.

The bit pattern of 135.46.57.14 is 10000111.00101110.00111001.00001110

Taking the first 22 bits of the above IP as network address (it is equivalent making the last 10 bits zero), we have 10000111.00101110.00111000.00000000 = 135.46.56.0

It matches the network address of the first row. The packet will be forwarded to interface 0.

(c) 135.46.52.2

Taking the first 22 bits of the above IP address as network address, we have 135.45.52.0. It dose not matche the network addresses of the first three rows. The packet will be forwarded to default gateway which is Router 2.

(d) 192.53.40.7

Taking the first 23 bits of the above IP address as network address, we have 192.53.40.0. It matches the network address of the third row. The packet will be forwarded to Router 1.

(e) 192.53.56.7

Taking the first 23 bits of the above IP address as network address, we have 192.53.56.0. It does not match the network addresses of the first three rows. The packet will be forwarded to default gateway which is Router 2.

5. Suppose two hosts, A and B, are separated by 30,000 kilometers and are connected by a direct link of $R = 3$ Mbps. Suppose the propagation speed over the link is 2.5×10^8 meters/sec.

a. Calculate the bandwidth-delay product, $R \cdot d_{\text{prop}}$.

b. Consider sending a file of 900,000 bits from Host A to Host B. Suppose the file is sent continuously as one large message. What is the maximum number of bits that will be in the link at any given time?

a.

The distance between two hosts A and B = 30,000 km = 3×10^7 meters

Transmission rate(R) of the direct link between A and B = 3Mbps = 3×10^6 bps

Propagation Speed(S) of the link between A and B = 2.5×10^8 meters/sec

Calculate the propagation delay: $d_{\text{prog}} = \frac{\text{Distance}}{\text{Speed}} = \frac{3 \times 10^7}{2.5 \times 10^8} = 0.12\text{s}$

Calculate the band-width delay product: $R \cdot d_{\text{prog}} = 3 \times 10^6 \times 0.12 = 360000$ bits

Therefore, band-width delay product is: 360000 bits

b.

Size of the file = 900 000 bits = 9×10^5 bits

*Transmission rate(R) of the direct link between A and B = 3Mbps = $3 * 10^6$ bps*

*The band-width delay product: $R * d_{prog} = 3 * 10^6 * 0.12 = 360000$ bits*

Therefore, the maximum number of bits at a given time will be 360000 bits

6. Let $g(x)=x^3+x+1$. Consider the information sequence 1001. Find the codeword corresponding to the preceding information sequence. Using polynomial arithmetic we obtain

$$g(x) = x^3 + x + 1 \quad 1011$$

Using polynomial arithmetic we obtain:

$$\begin{array}{r} 101 \\ \hline 1011 \mid 1001000 \\ | 1011 \\ \hline 001000 \\ 1011 \\ \hline 00110 \end{array}$$

Codeword = 1 0 0 1 1 1 0

6. Let $g(x)=x^3+x+1$. Consider the information sequence 1011. Find the codeword corresponding to the preceding information sequence. Using polynomial arithmetic we obtain

$$g(x) = x^3 + x + 1 \quad 1011$$

Using polynomial arithmetic we obtain:

$$\begin{array}{r} 1000 \\ \hline 1011 \mid 1011000 \\ | 1011 \\ \hline 0000 \end{array}$$

Codeword = 1011000

7. A packet switch receives a packet and determines the outbound link to which the packet should be forwarded. When the packet arrives, one other packet is halfway done being transmitted on this outbound link and four other packets are waiting to be transmitted. Packets are transmitted in order of arrival.

Suppose all packets are 2,500 bytes and the link rate is 3 Mbps. What is the queuing delay for the packet? More generally, what is the queuing delay when all packets have length L , the transmission rate is R , x bits of the

currently-being-transmitted packet have been transmitted, and n packets are already in the queue?

Packet length = L

Transmission rate = R

Currently transmitted packet = x bits

Waiting queue = n packets

Formula for Queuing delay:

$$\text{Queuing delay} = \frac{[nL + (L-x)]}{R}$$

Given data:

$$L = 2500 \text{ bytes} = (8 * 2500) \text{ bits}$$

$$R = 3 \text{ Mbps.} = 3 * 10^6 \text{ bit/s}$$

$$X = \frac{2500}{2} = 1250 \text{ bytes} = (8 * 1250) \text{ bits}$$

$$n = 4$$

Calculation:

$$\frac{[nL + (L-x)]}{R} = \frac{(4*2500 + 1250) * 8}{3*10^6} = 0.03 \text{ s}$$

Thus, the queuing delay: 0.03s

8. Suppose a header consists of four 16-bit words: (11111111 11111110, 11111111 00000000, 11110000 11110000, 11000000 11000001). Find the Internet checksum for this code

$$A0 = 11111111 \ 11111110 = 2^{16} - 2 = 65534$$

$$A1 = 11111111 \ 00000000 = 2^{16} - 256 = 65280$$

$$A2 = 11110000 \ 11110000 = 2^{16} - 3856 = 61680$$

$$A3 = 11000000 \ 11000001 = 2^{15} + 2^{14} + 2^7 + 2^6 + 2^0 = 49345$$

$$x = (A0 + A1 + A2 + A3) \% 65535 = 241839 \% 65535 = 45234$$

$$A4 = -x \% 65535 = -45234 \% 65535 = 20301$$

The checksum would = 0100111101001101

8. Suppose a header consists of four 16-bit words: (11111111 11111111, 11111111 00000000, 11110000 11110000, 11000000 11000000). Find the Internet checksum for this code

$$A0 = 11111111 \ 11111111 = 2^{16} - 1 = 65535$$

$$A1 = 11111111 \ 00000000 = 2^{16} - 256 = 65280$$

$$A2 = 11110000 \ 11110000 = 2^{16} - 3856 = 61680$$

$$A3 = 11000000 \ 11000000 = 2^{16} - 16192 = 49344$$

$$x = (A0 + A1 + A2 + A3) \% 65535 = 241839 \% 65535 = 45234$$

$$A4 = -x \% 65535 = -45234 \% 65535 = 20301$$

The checksum would = 01001111 01001101

8. Suppose a header consists of four 16-bit words: (11111111 11111111, 11111111 00000000, 11110000 11110000, 11000000 **11000001**). Find the Internet checksum for this code

$$A0 = 11111111 \ 11111111 = 2^{16} - 1 = 65535$$

$$A1 = 11111111 \ 00000000 = 2^{16} - 256 = 65280$$

$$A2 = 11110000 \ 11110000 = 2^{16} - 3856 = 61680$$

$$A3 = 11000000 \ 11000001 = 2^{15} + 2^{14} + 2^7 + 2^6 + 2^0 = 49345$$

$$x = (A0 + A1 + A2 + A3) \% 65535 = 241840 \% 65535 = 45235$$

$$A4 = -x \% 65535 = -45234 \% 65535 = 20300$$

The checksum would = 100111101001100

9. Consider a packet of length 2,000 bytes that propagates over a link of distance 3,500 km with propagation speed of $2.5 \cdot 10^8$ m/s, and transmission rate 2 Mbps?

- a. How long does the packet propagation take?
- b. Does this propagation delay depend on the packet length?
- c. Does this propagation delay depend on the transmission rate?

*Length: 2000 bytes = 8 * 2000 bits*

*Distance: 3,500 km = 35 * 10⁵ m*

*Propagation Speed 2,5 * 10⁸ m/s*

*Rate: 2Mbps = 2 * 10⁶ bit/s*

a.

$$d/s = (35 * 10^5) / (2,5 * 10^8) = 0,014 s = 14ms$$

b. No, the delay depend on packet length is not true.

c. No, the delay depends on transmission rate is not true.

10. Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates $R1 = 250$ kbps, $R2 = 3$ Mbps, and $R3 = 2$ Mbps.

- a. Assuming no other traffic in the network, what is the throughput for the file transfer?
- b. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B?

a)

Consider given data $R1 = 250$ kbps, $R2 = 3$ Mbps, and $R3 = 2$ Mbps

The throughput for the file transfer = $\min \{R1, R2, R3\} = \min \{ 250 \text{ kbps}, 3 \text{ Mbps}, 2 \text{ Mbps}\} = 250 \text{ kbps}$

Thus, the throughput for the file transfer = 250 kbps

b)

Consider given data:

The file size = 4 million bytes

Convert to bits = 32 000 000 bits

-From (a), Throughput for the file transfer=250 Kbps = 250 000 bps

-File size/Throughput for the file transfer = 32 000 000 / 250 000 = 128 (second)

10. Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links of rates $R1 = 500$ kbps, $R2 = 2$ Mbps, and $R3 = 1$ Mbps.

a. Assuming no other traffic in the network, what is the throughput for the file transfer?

b. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B?

c. Repeat (a) and (b), but now with $R2$ reduced to 100 kbps.

a)

Consider given data: $R1 = 500$ kbps, $R2 = 2$ Mbps, and $R3 = 1$ Mbps

The throughput for the file transfer= $\min \{R1, R2, R3\}$

$$=\min \{500 \text{ kbps}, 2 \text{ Mbps}, 1 \text{ Mbps}\}$$

$$=\mathbf{500 \text{ kbps}}$$

So, the throughput for the file transfer=500 kbps

b)

Consider given data:

The file size= 4 million bytes

Convert million bytes to bits

$$=32000000 \text{ bits.}$$

From (a), Throughput for the file transfer=500 Kbps
=500000 bps

Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B:

=file size/throughput for the file transfer
=32000000 bits/500000 bps
=64 seconds

c)

Consider the given data:

Repeat (a) and (b), but now with R2 reduced to 100 kbps.

That means R2=100 kbps, R1 = 500 kbps, and R3 = 1 Mbps

The throughput for the file transfer=min {R1, R2, R3}
=min {500 kbps, 100 kbps, 1 Mbps}
=100 kbps

So, the throughput for the file transfer=100 kbps

Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B:

=file size/throughput for the file transfer
=32000000 bits/100000 bps
=320 seconds

11. Suppose an application layer entity wants to send an L-byte message to its peer process, using an existing TCP connection. The TCP segment consists of the message plus 20 bytes of header. The segment is encapsulated into an IP packet that has an additional 20 bytes of header. The IP packet in turn goes inside an Ethernet frame that has 18 bytes of header and trailer. What percentage of the

transmitted bits in the physical layer correspond to message information, if $L = 200$ bytes, 1000 bytes, 2000 bytes.

TCP/IP over Ethernet allows data frames with a payload size up to 1460 bytes. Therefore, $L = 200$, $L=1000$ and $L= 2000$ are within this limit

The message overhead includes:

- *TCP: 20 bytes of header*
- *IP: 20 bytes of header*
- *Ethernet: total 18 bytes of header and trailer.*

Therefore, the total message overhead is 58 bytes

*Therefore $L = 200$ bytes, $(200/(200+58)) * 100 = 77,5\%$*

*$L = 1000$ byte, $(1000/(1000+58)) * 100 = 94.5\%$*

*$L = 2000$ bytes, $(2000/(2000+58)) * 100 = 97,1\%$*

11. Suppose an application layer entity wants to send an L -byte message to its peer process, using an existing TCP connection. The TCP segment consists of the message plus 20 bytes of header. The segment is encapsulated into an IP packet that has an additional 20 bytes of header. The IP packet in turn goes inside an Ethernet frame that has 18 bytes of header and trailer. What percentage of the transmitted bits in the physical layer correspond to message information, if $L = 100$ bytes, 500 bytes, 1000 bytes

TCP/IP over Ethernet allows data frames with a payload size up to 1460 bytes. Therefore, $L = 100$, 500 and 1000 bytes are within this limit.

The message overhead includes:

- *TCP: 20 bytes of header*

- *IP: 20 bytes of header*
- *Ethernet: total 18 bytes of header and trailer.*

Therefore

$$L = 100 \text{ bytes, } (100/(100+58)) * 100 = 63,3\%$$

$$L = 500 \text{ byte, } (500/(500+58)) * 100 = 89,6\%$$

$$L = 1000 \text{ bytes, } (1000/(1000+58)) * 100 = 94,5\%$$

12. Suppose the size of an uncompressed text file is 1 megabyte

- How long does it take to download the file over a 35 kilobit/second modem?
- How long does it take to take to download the file over a 1 megabit/second modem?
- Suppose data compression is applied to the text file. How much do the transmission times in parts (a) and (b) change?

If we assume a maximum compression ratio of 1:6, then we have the following times for the 35 kilobit and 1 megabit lines respectively:

a.

$$\text{Size text file} = 1 \text{ (megabyte)} = 1 * 1024 * 1024 * 8 \text{ (bit)}$$

$$\text{Speed} = 35 \text{ (kilobit/second)} = 35 * 1000 \text{ (bit / second)}$$

$$T(35k) = (1 * 1024 * 1024 * 8) / 35000 = 239.67 \text{ (second)}$$

b.

$$\text{Size text file} = 1 \text{ (megabyte)} = 1 * 1024 * 1024 * 8 \text{ (bit)}$$

$$\text{Speed} = 1 \text{ (megabit/second)} = 1 * 10^6 \text{ (bit / second)}$$

$$T(1M) = (1 * 1024 * 1024 * 8) / (1 * 10^6) = 8.38 \text{ (second)}$$

c.

$$T(35k) = (1 * 1024 * 1024 * 8) / (35000 * 6) = 39.9 \text{ (second)}$$

$$T(1M) = (1 * 1024 * 1024 * 8) / (1 * 10^6 * 6) = 1.39 \text{ (second)}$$

12. Suppose the size of an uncompressed text file is 1 megabyte

- a. How long does it take to download the file over a 32 kilobit/second modem?
- b. How long does it take to take to download the file over a 1 megabit/second modem?
- c. Suppose data compression is applied to the text file. How much do the transmission times in parts (a) and (b) change?

If we assume a maximum compression ratio of 1:6, then we have the following times for the 32 kilobit and 1 megabit lines respectively:

a)

$$\begin{aligned}\text{Size text file} &= 1 \text{ (megabyte)} = 1 * 1024 * 1024 * 8 \text{ (bit)} \\ \text{Speed} &= 32 \text{ (kilobit/second)} = 32 * 1000 \text{ (bit / second)} \\ \Rightarrow T(32k) &= (1 * 1024 * 1024 * 8) / (32 * 1000) = 262.144 \text{ (seconds)}\end{aligned}$$

b)

$$\begin{aligned}\text{Size text file} &= 1 \text{ (megabyte)} = 1 * 1024 * 1024 * 8 \text{ (bit)} \\ \text{Speed} &= 1 \text{ (megabit/second)} = 1 * 10^6 \text{ (bit / second)} \\ \Rightarrow T(1M) &= (1 * 1024 * 1024 * 8) / (1 * 10^6) = 8.38 \text{ (seconds)}\end{aligned}$$

c)

$$\begin{aligned}\Rightarrow T(32k) &= (1 * 1024 * 1024 * 8) / (32 * 1000 * 6) = 43.69 \text{ (seconds)} \\ \Rightarrow T(1M) &= (1 * 1024 * 1024 * 8) / (1 * 10^6 * 6) = 1.4 \text{ (seconds)}\end{aligned}$$

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

- (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.
- (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?

a. 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. 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

14. Sender A wants to send 100111010011110 to receiver B. This transmission uses CRC algorithm for error detection with generator polynomial bits string is 10110. What is bits string will be transmitted on the medium. Show your all steps to have result

Data: 100111010011110

Generator polynomial: 10110-> n = 5

Step 1: Add n-1 bits 0 to the bit stream

1001110100111100000

Step 2:

	101000010100010
10110	1001110100111100000
	10110

$$\begin{array}{r}
 0010110 \\
 \underline{10110} \\
 010011 \\
 \underline{10110} \\
 10111 \\
 \underline{10110} \\
 10000 \\
 \underline{10110} \\
 1100 \Rightarrow \text{FCS}
 \end{array}$$

Step 3:

Add R(x) to the information bits

->Codeword: 1001110100111101100

Bits string will be transmitted on the medium is: 1001110100111101100.

15. A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is $x^3 + 1$. Show the actual bit string transmitted. Suppose the third bit from the left is inverted during transmission. Show that this error is detected at the receivers end.

Bit stream: 10011101

Generator polynomial: $X^3 + 1 \Leftrightarrow 1001 \rightarrow n = 4$

Step 1: Add n-1 bits 0 to the bit stream

10011101000

Step 2:

$$\begin{array}{r}
 10001100 \\
 1001 \overline{) 10011101000}
 \end{array}$$

$$\begin{array}{r}
 \underline{1001} \\
 00001101 \\
 \underline{1001} \\
 01000 \\
 \underline{1001} \\
 000100 \text{ (Remainder)}
 \end{array}$$

Step 3: Actual frame transmitted: $10011101000 - 100 = 10011101100$ (modulo 2 subtraction)

Now suppose the third bit from the left is garbled and frame is received as 10111101100. Hence on dividing this by the polynomial generator we get a remainder of 100 which shows that an error has occurred. Had the received frame been error free we would have got a remainder of zero. See below.

$$\begin{array}{r}
 10101000 \\
 1001 \overline{) 10111101100} \\
 \underline{1001} \\
 001011 \\
 \underline{1001} \\
 001001 \\
 \underline{1001} \\
 0000100 \\
 100 \text{ (Remainder indicating error)}
 \end{array}$$

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

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

(b) Design an appropriate CIDR addressing scheme.

[Answer]

a) Class B's address has 14 bits for the network ID and 16 bits for the host ID. We need to decide how many bits to allocate to the subnet id versus the host id for designing an appropriate subnet addressing scheme. We can choose either 7 bits or 8 bits to identify the hosts.

b) The CIDR addressing scheme involves generating a prefix length that indicates the length of the network mask.

A router has the following CIDR entries in its routing table:

Address/mask Next hop

135.46.56.0/22 Interface 0

135.46.60.0/22 Interface 1

192.53.40.0/23 Router 1

default Router 2

Câu a) What does the router do if a packet with an IP address 135.46.63.10 arrives?

Câu b) What does the router do if a packet with an IP address 135.46.57.14 arrives?

[Answer]

a) The router will forward the packet to Interface 1 if a packet with an IP address 135.46.63.10 arrives.

b) The packet will be forwarded to Interface 0 if a packet with an IP address 135.46.57.14 arrives.

Ans:

Ans: (a) 135.46.63.10

Taking the first 22 bits of 135.46.63.10 as network address, we have 135.46.60.0.

The bit pattern of 135.46.63.10 is 10000111.00101110.00111111.00001010

When we perform the bit and operation with 22 leading bit 1s and 10 bit 0s, it is equivalent of making the last 10 bit zero.

We get the following network address bit pattern:
10000111.00101110.00111100.00000000.

The first two bytes are not changed.

The 3rd byte changes from 63 to 60 while the 4th byte becomes zero.

Match with network address in the routing table. The 2nd row matches. The router will forward the packet to Interface 1.

Cách nhận biết:

Xét 135.46.63.10 có 135.46 giống Interface 0 và 1.

135.46.63 lớn hơn Interface 1 thì chọn Interface 1, ngược nếu ví dụ như đề câu b chỉ có 57 (lớn hơn 56 nhưng nhỏ hơn 60) -> chọn Interface 0.

Nếu đề hỏi khác nữa như cho 10.10.10.10 không giống cái nào ở Interface 0, 1 hay Router 1 thì mặc định chọn Default -> Router 2.

Consider the three-way handshake in TCP connection setup.

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.

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?

[Answer]

a) 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 acknowledge segment from B, A will find out that B received a wrong sequence number. A will discard the acknowledgement packet and reset the connection.

b) 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.

A bit stream 1101011011 is transmitted using the standard CRC method. The polynomial generator is $x^3 + x + 1$. What is the actual bit string transmitted? Show the major steps to your answer.

[Answer]

Bitstream is 1101011011

The polynomial generator is $x^3 + x + 1$

The Bit form = 111

Thus, $n=3$.

Suppose a IP header consists of four 16-bit words: (11111111 11111111, 11111111 00000000, 11110000 11110000, 11000000 11000000). Please find the Internet checksum for the code.

[Answer]

Đổi cái đăm nhị nhân 16 bit ra decimal, ra số âm xong lấy 65536 (16 bit thì là 2^{16}) rồi cộng cho số âm đó.

$$A0 = 11111111 \ 11111111 = 2^{16} - 1 = 65535$$

$$A1 = 11111111 \ 00000000 = 2^{16} - 256 = 65280$$

$$A2 = 11110000 \ 11110000 = 2^{16} - 3856 = 61680$$

$$A3 = 11000000 \ 11000000 = 2^{16} - 16192 = 49344$$

$$x = (A0 + A1 + A2 + A3) \% 65535 = 241839 \% 65535 = 45234$$

("%" có nghĩa là chia lấy phần dư, hay còn gọi là modulo)

$$A4 = -x \% 65535 = -45234 \% 65535 = 20301 \text{ (đổi } x \text{ sang số âm, sau đó chia } 65535 \text{ lấy phần dư)}$$

The checksum would = 01001111 01001101 (đổi A4 decimal sang nhị phân ra checksum).

Suppose that a group of computers is connected to an Ethernet LAN. If the computers communicate only with each other, does it make sense to use IP protocol in the computers? Should the computers run TCP directly over Ethernet? How is addressing handled?

[Answer]

IP convention can be utilized since the IP convention is a lot of necessities for tending to also as directing information on the Internet. Since the computer can't run principles-based TCP without IP, TCP uses IP addresses. There is a need to utilize an individual custom streaming convention. that depends on Ethernet or another link layer as a lower layer.

(1) The figures below show the TCP/UDP communication pattern diagrams. Which diagram works for TCP? Why?

(2) Fill the missing steps (blank boxes) in both diagrams for TCP/UDP correspondingly.

Link ảnh đi kèm của đề: <https://imgur.com/5A2tz94>

[Answer]

1) Diagram (b) will work for TCP since it is a network protocol that shows the details of how data is sent as well as received.

2) Missing steps : (a) bind, connect , (b) bind>listen>accept, connect.

Suppose that two peer-to-peer processes provide a service that involves the transfer of discrete messages. Suppose that the peer processes are allowed to exchange

PDUs that have a maximum size of M bytes including H bytes of header. Suppose that a PDU is not allowed to carry information from more than one message.

a. Develop an approach that allows the peer processes to exchange messages of arbitrary size.

b. What essential control information needs to be exchanged between the peer processes?

c. Now suppose that the message transfer service provided by the peer processes is shared by several message source-destination pairs. Is additional control information required, and if so, where should it be placed?

[Answer]

a) To convert arbitrary sizes, large contents must be split into bytes of each length that will be transmitted in multiple PDUs.

b) Peer-to-peer processes move information allowing messages to be aggregated.

c) A PDU must be attached to the sequence ID, to process each sequence independently when a message is selected. This stream ID can be dodged if the source and target work for them to process a content transfer at a certain time.

Consider a network link that has distance of 100 meters, and signal traverses at the speed of light in cable 2.5×10^8 meters per second. The link has transmission bandwidth of 100 megabits/second (100×10^6 bits per second). The packet size is 400 bits. What is the signal propagation delay?

[Answer]

$d = 100 \text{ megabits} / \text{second}$

(Cách đổi đơn vị: 100 megabits / second = 100×10^3 kilobits / seconds, đổi ngược lại thì chia cho 1000)

$$v = 2.5 \times 10^8 \text{ meters per / second}$$

Đề này cho đã cùng đơn vị sẵn nên không cần phải đổi, lấy chia luôn.

Công thức Signal propagation delay (seconds):

$$\Rightarrow t(\text{prop}) = d / v = 100 / (2.5 \times 10^8) = 4 \times 10^{-7} \text{ seconds}$$

Consider a network link that has distance of 100 meters, and signal traverses at the speed of light in cable 2.5×10^8 meters per second. The link has transmission bandwidth of 100 megabits/second (100×10^6 bits per second). The packet size is 400 bits. What is the packet transmission delay?

[Answer]

$$L = 400 \text{ bits (packet size)}$$

$$R = 100 \times 10^6 \text{ (bit / sec)}$$

Đề này cho đã cùng đơn vị sẵn nên không cần phải đổi, lấy chia luôn.

Công thức Packet transmission delay (seconds):

$$\Rightarrow t(\text{trans}) = L / R = 400 / (100 \times 10^6) = 4 \times 10^{-6} \text{ seconds}$$

Suppose an application layer entity wants to send an L-byte message to its peer process, using an existing TCP connection. The TCP segment consists of the message plus 20 bytes of header. The segment is encapsulated into an IP packet that has an additional 20 bytes of header. The IP packet in turn goes inside an Ethernet frame that has 18 bytes of header and trailer. What percentage of the transmitted bits in the physical layer correspond to message information, if L = 100 bytes, 500 bytes, 1000 bytes

Note: Explain your answer in details

[Answer]

Đây cũng là cách trình bày để lấy trọn vẹn 3 điểm của bài.

TCP / IP over Ethenet allows data frames with a payload size up to 1460 bytes. Therefore, L = 100, 500, 1000 are within this limit (chưa biết nếu vượt limit thì khác thế nào, nhưng mình nghĩ đề ra sẽ không vượt limit đâu).

TCP: 20 bytes of header (lấy trong đề)

IP: 20 bytes of header (lấy trong đề)

Ethernet: 18 bytes of header and trailer (lấy trong đề)

Therefore:

L = 100 bytes, $100 / (100 + 20 + 20 + 18) = 63\%$ efficiency

L = 500 bytes, $500 / (500 + 20 + 20 + 18) = 90\%$ efficiency

L = 1000 bytes, $1000 / (1000 + 20 + 20 + 18) = 95\%$ efficiency

Suppose the size of an uncompressed text file is 1 megabyte

- a. How long does it take to download the file over a 32 kilobit/second modem?
- b. How long does it take to take to download the file over a 1 megabit/second modem?
- c. Suppose data compression is applied to the text file. How much do the transmission times in parts (a) and (b) change?

If we assume a maximum compression ratio of 1:6, then we have the following times for the 32 kilobit and 1 megabit lines respectively:

Note: Explain your answer in details.

[Answer]

Câu a)

(Đổi hết sang đơn vị bit và bit / second)

Size text file = $1 \times 1024 \times 1024 \times 8$ (bit)

Speed = 32×1000 (bit / second)

$\Rightarrow T(32k) = (1 \times 1024 \times 1024 \times 8) / (32 \times 1000) = 262.144$ (seconds)

Câu b)

(Đổi hết sang đơn vị bit và bit / second)

Size text file = $1 \times 1024 \times 1024 \times 8$ (bit)

Speed = $1 \times 1000 \times 1000$ (bit / second)

$\Rightarrow T(1M) = (1 \times 1024 \times 1024 \times 8) / (1 \times 1000 \times 1000) = 8.38$ (seconds)

Câu c)

(Đề kêu 1:6 thì chỉ việc nhân thêm cho 6 ở chỗ tốc độ là xong, nếu trường ra đề 1:10 thì nhân 10)

$\Rightarrow T(32k) = (1 \times 1024 \times 1024 \times 8) / (32 \times 1000 \times 6) = 43.69$ (seconds)

$\Rightarrow T(1M) = (1 \times 1024 \times 1024 \times 8) / (1 \times 1000 \times 1000 \times 6) = 1.4$ (seconds)

Sender A wants to send 100111010011011 to receiver B. This transmission uses CRC algorithm for error detection with generator polynomial bits string is 10111. What is bits string will be transmitted on the medium. Show your all steps to have result.

Note: Explain your answer in details.

[Answer]

Step 1: Add 0000 to data bits string. It will be 1001110100110110000 (0.5 điểm)

Step 2: Divide 1001110100110110000 to 10111 by modulo (chia lấy phần dư)

(Cách chia:

Lấy 5 số đầu dùng phép xor với 10111

$10011 \text{ xor } 10111 \Rightarrow 00100$

Lấy tiếp 2 số cho đủ 5 số (so với số 1 đầu tiên)

$\Rightarrow 10010 \text{ xor } 10111 = 00101$

Lấy tiếp 2 số cho đủ 5 số (so với số 1 đầu tiên)

$\Rightarrow 10110 \text{ xor } 10111 = 00001$

Lấy tiếp 4 số cho đủ 5 số (so với số 1 đầu tiên)

$\Rightarrow 10110 \text{ xor } 10111 = 00001$

Lấy tiếp 4 số cho đủ 5 số (so với số 1 đầu tiên)

$\Rightarrow 11100 \text{ xor } 10111 = 01011$

Lấy tiếp 1 số cho đủ 5 số (so với số 1 đầu tiên)

$\Rightarrow 10110 \text{ xor } 10111 = 0001$

Do còn mỗi 1 số 0, không còn đủ để bù đắp nữa, thêm vào chuyển thành kết quả
 $\Rightarrow 00010$

(Xong được phần này các bạn được 1.5 điểm, nhưng làm giấy phải làm đúng như phép chia thực thụ nha, các bạn tập tành làm)

Step 3: Lấy 4 số cuối của step 2, thay số 4 số 0000 ở step 1, ta được kết quả cuối cùng.

The transmitted bits string is 1001110100110110010. (0.5 điểm)

Suppose two hosts, A and B, are separated by 30,000 kilometers and are connected by a direct link of $R = 3$ Mbps. Suppose the propagation speed over the link is 2.5×10^8 meters/sec.

- Calculate the bandwidth-delay product, $R \cdot d_{\text{prop}}$.
- Consider sending a file of 900,000 bits from Host A to Host B. Suppose the file is sent continuously as one large message. What is the maximum number of bits that will be in the link at any given time?

A)ta có

A và B cách nhau 30000km--> $3 \cdot 10^7$ mét(1km bằng 10^3 m)

Tốc độ truyền tải (R) của liên kết trực tiếp giữa A và B = 3 Mbps--> $3 \cdot 10^6$ bps(1Mbps = 10^6 bps)

Tốc độ lan truyền (S) của liên kết trực tiếp giữa A và B = $2.5 \cdot 10^8$ mét/s

giải

-Calculate the propagation delay(độ delay tốc độ lan truyền)

$$d(\text{prog}) = (\text{khoảng cách})/(\text{tốc độ}) = 3 \cdot 10^7 / 2.5 \cdot 10^8 = 0.12 \text{ sec(giây)}$$

- Calculate the band-width delay product(Tính tích số trữ độ rộng băng tần)\

$$R \cdot d(\text{prog}) = 3 \cdot 10^6 \cdot 0.12 = 36 \cdot 10^4 \text{ bits}$$

band-width delay product là 360000 bits

B) Size of a file is 900000 bits= $9 \cdot 10^5$ bits

Vẫn y như đáp án câu A, số bits lớn nhất là 360000 bits

7. A packet switch receives a packet and determines the outbound link to which the packet should be forwarded. When the packet arrives, one other packet is halfway done being transmitted on this outbound link and four other packets are waiting to be transmitted. Packets are transmitted in order of arrival.

Suppose all packets are 2,500 bytes and the link rate is 3 Mbps. What is the queuing delay for the packet? More generally, what is the queuing delay when all packets have length L , the transmission rate is R , x bits of the currently-being-transmitted packet have been transmitted, and n packets are already in the queue?

package length = L

transmission rate = R

currently transmitted packet = x bits

Waiting queue = n package

-Công thức queuing delay

Queuing delay = $[(nL) + (L-x)] * R(\text{Mbps}) * n / R(\text{bps})$

ta có

$L = 2500$ bytes

$R = 3 \text{ Mbps} = 3 * 10^6 \text{ bps}$

$x = 2500/2 = 1250$ (transmit halfway)

$n = 4$ (do có 4 package đang chờ chuyển)

giải

$$-[nL + (L-x)] = [(4 \cdot 2500) + (2500 - 1250)]$$

$$10000 + 1250 = 11250 \text{ bytes}$$

-Package transmit rate at 3Mbps

$$11250 \cdot 3 \cdot 4 = 135000$$

$$\text{-Queuing delay} = 135000 / 3 \cdot 10^6 = 0.045 \text{ sec (giây)}$$

Consider a packet of length 2,000 bytes that propagates over a link of distance 3,500 km with propagation speed of $2,5 \cdot 10^8$ m/s, and transmission rate

2 Mbps?

a. How long does the packet propagation take? b. Does this propagation delay depend on the packet length?

c. Does this propagation delay depend on the transmission rate?

a.

Length: $8 \cdot 2000$ bit

Distance: $35 \cdot 10^5$ m

Propagation Speed $2,5 \cdot 10^8$ m/s

Rate: $2 \cdot 10^6$ bit/s

$$(35 \cdot 10^5) / (2,5 \cdot 10^8) = 0,014 \text{ s} = 14 \text{ ms}$$

b. No

c. No

Suppose Host A wants to send a large file to Host B. The path from Host A to Host B has three links, of rates $R_1 = 250$ kbps, $R_2 = 3$ Mbps, and $R_3 = 2$

Mbps.

- a. Assuming no other traffic in the network, what is the throughput for the file transfer?
- b. Suppose the file is 4 million bytes. Dividing the file size by the throughput, roughly how long will it take to transfer the file to Host B?

$R_1 = 250 \text{ kbps}$

$R_2 = 3 \text{ Mbps}$

$R_3 = 2 \text{ Mbps}$

a. Throughput

- Bandwidth (Băng thông): Độ rộng của đường truyền tin. (Khoảng chênh lệch giữa thành phần tần số thấp nhất tới thành phần tần số cao nhất trong 1 tín hiệu = Hz) \Rightarrow Khả năng truyền đc nhiều dữ liệu tại 1 thời điểm là càng lớn. Đo bằng bit/s
- Tốc độ truyền (Transmission Rate): Là tốc độ truyền được bao nhiêu bit /s. Đo cũng bằng bit/s
- Thông lượng (Throughput): Là số lượng bit hữu ích truyền đi trong 1 s. bit/s. Luôn nhớ: thông lượng \leq tốc độ truyền

A) the throughput for the file

$\text{transfer} = \min\{R_1, R_2, R_3\} = \min\{250 \text{ kbps}, 3 \text{ Mbps}, 2 \text{ Mbps}\} = 250 \text{ kbps}$

B) the file size = 4 million bytes

-convert to bits = 32000000 bits

-From (a), Throughput for the file transfer = 250 Kbps = 250000 bps

-File size/Throughput for the file transfer = $32000000 / 250000 = 128 \text{ giây(second)}$

12. Suppose the size of an uncompressed text file is 1 megabyte

- a. How long does it take to download the file over a 35 kilobit/second modem?
- b. How long does it take to take to download the file over a 1 megabit/second modem?
- c. Suppose data compression is applied to the text file. How much do the transmission times in parts (a) and (b) change?

If we assume a maximum compression ratio of 1:6, then we have the following times for the 35 kilobit and 1 megabit lines respectively:

Ta có 1 file dung lượng là 1 megabytes(1 MB)

a) cần tốn thời gian bao lâu để tải file đó nếu có mạng 35 kilobit/second (35kb/s)

Ta cần đổi hết các đơn vị sang bit

ta có $1\text{MB} = 8\text{Mb}(\text{megabits}) = 8000000(\text{bits})$

NHưng do phần này đang ở dạng nhị phân nên ta không dùng cách đổi

$8\text{Mb} = 8 \cdot 10^6$ bit được

Ta dùng $8 \cdot 1024 \cdot 1024$ là ra được đáp án đầu = 8388608 bits

Ta có 1 kilobit = 1000 bit nên $35\text{Kb/s} = 35000 \text{ bits/s}$

Lấy dung lượng file được cho chia cho dung lượng mạng

$8388608/35000 = 239.67$ giây(second)

FULL câu(yêu cầu làm theo thể này) : $T(32k) = 8 \cdot 1024 \cdot 1024 / 35000 = 239.67$ giây(second)

b)Như trên, ta có 1 megabit = $1 \cdot 10^6$ bits

$$8388608/1*10^6 = 8.38 \text{ giây(second)}$$

FULL câu(yêu cầu làm theo thể này) : $T(1M) = 8*1024*1024/1*10^6 = 8.38$ giây(second)

c) If we assume a maximum compression ratio of 1:6, then we have the following times

for the 32 kilobit and 1 megabit lines respectively:

$$T(32k) = 8*1024*1024/(35000*6) = 39.9 \text{ giây(second)}$$

$$T(1M) = 8*1024*1024/(1*10^6*6) = 1.39 \text{ giây(second)}$$

Three users are sharing a common link of 1 Mb/s. User A is

downloading large files and is connected to the common link via a slow access link at x

Mb/s, user B is connected via a 100 Mb/s link, but the application she is using requires at

most x Mb/s, and finally user C is connected via a 1 Gb/s link and is downloading a

movie that can take up any amount of bandwidth available to it. What is the max-min fair

allocation for these three flows at the common link?

Two Cases: 1) If $x < 1/3 \text{ Mb/s}$ then A and B get x Mb/s, C gets $(1-2x) \text{ Mb/s}$, otherwise A, B,

and C get $1/3 \text{ Mb/s}$

If we want to “break-in” to an ongoing TCP connection, and pretend to be one of the end-hosts, we need to send an sequence number that will convince the other end that we are legitimate. For example, imagine that a source host, A, tries to create a TCP connection with host B. When host A sends the SYN message, host C (masquerading as B) sends a SYN+ACK message to host A with the correct sequence number. Even though C didn’t see the original SYN message, it gets lucky and guesses the right sequence number to send back so as to fool host A.

(a.) If Host C were to guess the value from among all possible 32-bit sequence numbers, on average how many guesses would it take to get it right?

2^{31}

If Host C already knew 12 of the bits (e.g. because it knew something about the pseudo-random ISN generator), on average how many guesses would it take to get it right?

2^{19}

If Host C wanted to send a spoofed email to A, it would take just one “guessed” packet. Assuming the e-mail body and packet headers were 1500bytes, given a 1Gb/s

link (and assuming that C already knew 12 bits of the sequence number), on average

how long would it take to send one e-mail with a spoofed source?

$$2^{19} * 1500\text{bytes} * 8\text{bits/byte} * 1 / 10^9 = 6.3\text{seconds}$$

(b.) What simple change could you add to the SMTP protocol to defend against this

attack?

Send an application level challenge and wait for a response before sending the message.

(Multiple answers were accepted for this. Solutions requiring secrets or keys were docked points because this is a public mail system.)

(c.) (See Figure on next page). A variant of spoofing is used to protect a highbandwidth (and thus valuable) machine by forging the source of a lower bandwidth (less valuable) machine also under the attacker's control. The victim's return packets are then forwarded by the low-bandwidth host back to the high-bandwidth host which then can retrieve the correct sequence number.

Assuming the low bandwidth host is connected to a 56Kb/s link and the high bandwidth host has a 1Gb/s connection, how many messages per second can the high bandwidth server send (again use single packet e-mail messages of 1500bytes)?

Acks on the 56k link are the bottleneck in this attack. Since an ACK is generally 64 bytes

$$56\text{k}/(64*8\text{bits}) = 109\text{messages/sec}$$

If Host C already knew 12 of the bits (e.g. because it knew something about the

pseudo-random ISN generator), on average how many guesses would it take to get it

right?

2^{19}

If Host C wanted to send a spoofed email to A, it would take just one “guessed”

packet. Assuming the e-mail body and packet headers were 1500bytes, given a 1Gb/s

link (and assuming that C already knew 12 bits of the sequence number), on average

how long would it take to send one e-mail with a spoofed source?

$$2^{19} * 1500\text{bytes} * 8\text{bits/byte} * 1 / 10^9 = 6.3\text{seconds}$$

(d.) Does your fix for part (d) above also apply to this form of spoofing?

In general, any solution that relied on the attacker not receiving the initial SYN|ACK does not work in this setup.

Congestion Control. Physicists at Stanford Linear Accelerator Center

(SLAC) generate a few terabytes of data per day, which they would like to share with

their colleagues in CERN, Geneva. SLAC and CERN are connected with 10 Gb/s links

end-to-end and have a round-trip time of 200 milliseconds. They use standard TCP in

transferring the bulk file, but quickly discover they are not able to achieve a sustained 10

Gb/s from end to end. They consult the local networking expert, Alice, who conjectures

that the problem is the way in which TCP responds to losses in the network. Let's first

find out if Alice's conjecture is true.

(a.) Assuming a simplified model of TCP (ignore Slow-Start and assume TCP is in the AIMD mode) derive an approximate expression for TCP's average window size as a function of the loss-probability, p .

The area under the sawtooth is $(1/2 * W/2 * W/2) + (W/2 * W/2) = 3/8 * W^2$. The probability that a packet will be dropped is $1/(3/8 * W^2)$ (1 drop per sawtooth).

Solving for W , we see $W = \sqrt{8/3 * p}$. The average window is equal to $3/4$ of W , which is $\sqrt{3/2p}$.

(b.) Using your answer to part (a), what is the average number of round-trip times between loss events that TCP can tolerate so as to sustain an average window size of w packets. Observe that the time interval between losses needs to increase as the "pipe" size increases.

Since the average window, $w[a]$, is equal to $3/4$ of W , then we can see that W is equal to $4/3$ of w . $(W * RTT)/2$ is the time interval between loss events.

Therefore, we need $(2/3) * w[a] * RTTs$ between loss events

(c.) Let's see what it takes to fill the pipe (i.e. achieve 10Gb/s). To achieve 10Gb/s, we need an average window size of

!

10Gb/s " 200ms ÷ 1500bytes

(we are

assuming the packets are 1500bytes long). How low a loss-rate do we need for this to happen? Comment on your answer.

The average window size is

$$(10 \times 10^9 \times 200 \times 10^{-3}) / 1500 \times 8 = (5 \times 10^5) / 3 = 166666.67$$

packets. Using the equation from part (a), we can see that $\sqrt{3/2P} = 166666.67$. Solving for p , we see that we need a 5.4×10^{-11} loss rate.

(d.) Alice recommends Alice-TCP: an alternative to AIMD – and proposes that TCP

be modified to use MIMD instead (Multiplicative Increase Multiplicative Decrease). She believes it will increase the throughput for bulk transfers. In particular, she proposes that when an acknowledgement is correctly received, Alice-TCP will increase the window size by additive constant a (i.e. $w \rightarrow w+a$, so that it increases by a multiplicative amount when the whole window has been acknowledged); when there is a loss, the window size is multiplied by factor $(1-b)$, where $b < 1$. Assuming that exactly one packet is lost each time the window size reaches W , show that the average window size is given by: $w = ((2-b)/2) \times W$

The average window per cycle is equal to the sum of one half of the smallest window size and largest window size. The largest window equals W and the

smallest window is equal the drop factor multiplied by W . Therefore,

$$w = 1/2 ((1-b)*W + W) = (((1-b)*W + W)/2) = ((2-b)/2)*W$$

(e.) Alice argues that Alice-TCP is better. She says that if we want to sustain a particular average window size, the number of round-trip times between losses is independent of the pipe size. Is she correct? Explain your answer.

Show that for MIMD TCP, the following holds true:

$$pw[a] = a/b (1 - b/2) =: p$$

So, the number of round-trips times between losses for a sustainable average window size, w_a , does not increase with the pipe size!

One way to do this would be to write a differential equation for rate change and equate it to 0 for the equilibrium

$$w(i) = ((aw[i](t))/T[i]) - ((2b)/(2 - b))*x[i](t)*p(i)(t)w(i)(t)$$

where $x[i](t) := w[i](t)/T[i]$. Now equate equilibrium rate change $x[i](t)$ to zero.

A router buffers arriving packets in separate queues – one per flow. Assume that the queues are all empty at time 0, and that the packets of each flow have a fixed size. Assume that flow A's traffic is confined to a

leaky-bucket constraint with $\sigma = 5$ packets, and $\rho = 10$ packets per second. The queues

are all served using a WFQ scheduler, so as to meet an end-to-end delay guarantee.

(a.) What is the maximum backlog (in packets) at the queue corresponding to flow A?

5 packets

Packets of flow B follow a modified leaky-bucket model: The number of arrivals until time t is given by $a(0,t) \leq 5 + t$

(b.) What is the maximum backlog at the queue corresponding to flow B?

5 packets

Packets of user C enter the router periodically. All we know is that in each 10 seconds, no more than 1000 packets arrive.

c.) Can this traffic be modeled as leaky-bucket traffic? If so, what are the corresponding σ and ρ parameters? If not, explain why.

Yes. $\sigma=1000$ and $\rho=100$

(d.) At a given time, assume packet p of flow A is scheduled to leave the router before packet q of flow B. Is it possible that a future arriving packet changes the departure order of these packets? Why?

When any two packets have been scheduled, they have been placed in order of increasing Finishing Time (as determined by the WFQ scheduler). They then leave in order of Finishing Time. Therefore, once scheduled, their departure

order is fixed

Leaky-bucket constrained traffic. In this question we'll assume that flows consist of a stream of bits rather than packets. In class, we saw how a leaky bucket regulator can constrain a flow so that the number of bits sent in an interval of duration τ is given by: $A(t, t + \tau) \leq \sigma + \rho\tau$. If you think about it, this leaky bucket would allow a burst of up to σ bits to depart from the source in zero time (i.e. at an infinite rate)! This is clearly not possible; so in practice, the source is constrained by the data rate of its outgoing link. We'll assume that the flow is constrained by a leaky bucket at the source, and that a burst can depart no faster than R bits/second (the data rate of the link to which the source is connected). The figure below shows how the traffic is constrained. The flow passes through ten routers, all of which use bit-by-bit weighted fair queuing to serve

(a.) If the first router allocates the flow a service rate greater than or equal to c (where $\rho \leq c$), write down expressions for the maximum delay of a bit in the router, and the size of the buffer needed to prevent overflows. (Write your answers in terms of σ , ρ and R , but not c).

Maximum buffer size= $\sigma - (\sigma\rho)/R$, Delay $(\sigma - (\sigma\rho)/R)/\rho = \sigma/\rho - \sigma/R$

(b.) What happens to your answer in (a) when $R = \rho$? Explain your answer.

No buffer is needed

(c.) Why do you think it is customary to assume that a burst can leave a leaky bucket

regulator at infinite rate?

R is usually much greater than ρ so the answer changes very little when using R
