COMPUTER NETWORK

ASSIGNMENT-1

Submitted by-ABHISHEK GUPTA(IIT2018187)

Q1. *Let suppose an entity generates 4454 bytes of data. Suppose also that by the time this data arrives at the data link layer, 106 bytes of header information has been added. At the data link layer, the maximum frame size is 1568 bytes, of which 48 bytes are its header.*

(a) How many frames will be used?

(b) How many total bytes must be transmitted?

(c) What percentage of the transmitted bits are from the application layer?

solution:

a) How many frames will be used?

Bytes arrives at the data link layer

=4454+106

=4560 Bytes.

New, within each frame, only 1568 bytes out of 4560 can be sent.

A data link layers payload contains

1568 - 48 =1520 bytes.

So, no of frames =4560/1520=3.

Here , 3 frames must be sent.

b) We know that ,

Each data link layer frames still needs header, so when sending 3 frames,

headers also need to be added to each frame.

So,total bytes sent = 3*1568

=4704 bytes.

c) Percentage of transmitted bits frame application layer

=(4454/4704)*100

=0.9468*100

=94.68%

Q2. *Suppose two hosts, A and B, are separated by 10,000 kilometers and are connected by a direct link of R = 2 Mbps. Suppose the propagation speed over the link is 2.5 * 10^8 meters/sec.*

*(a) Calculate the bandwidth-delay product, R * d_prop.*

(b) Consider sending a file of 1000,000 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?

(c) Provide an interpretation of the bandwidth-delay product.

Solution:

The distance (Distance) between two hosts A and B = 10,000 km

a)

The distance (Distance) between two hosts A and B = 10,000 km

$$= 1 \times 10^7 \ metres$$

Transmission rate(R) of the direct link between A and B =2Mbps

$$= 2 \times 10^6 bps \ (1Mbps = 10^6 bps)$$

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

Calculate the propagation delay:

$$d_{prog} = \frac{distance}{speed} = \frac{1\times10^7}{2.5\times10^8} = 0.04 \text{ sec}$$

Calculate the band-width delay product:

$$R \times d_{prog} = 2 \times 0.04 \times 10^6 = 8 \times 10^4 \; bits$$

Therefore, bandwidth delay product is 80000bits

b)

Size of the file =1000,000 bits

Transmission rate(R) of the direct link between A and B =2Mbps

$$= 2 \times 10^6 bps \; (1Mbps = 10^6 bps)$$

The band-width delay product:

$$R \times d_{prog} = 2 \times 0.04 \times 10^6 = 8 \times 10^4 \; bits$$

Therefore, the maximum number of bits at a given time will be 80000bits.

c)

The product of band-windth delay is equal to the maximum number of bits on the transmission line.

Q3: *Assume the two hosts A and B are connected Via a Router R1 which is having a queuing delay of 7 μ sec. is as shown below: The Router R has two links Link1 and Link2, connected to A and B respectively. Each link is 250 meters long and has a bandwidth of 10^9 bps. The propagation speed is 500,000 km/sec. The total transfer delay for 2 KB packet from A to B is ...in μ sec. (ignore the processing delay).*

Solution:

$T_X = (250/500000) \times 10^3 = 0.5\ \mu s.$

Now it reaches to the router after= (16.384+0.5) μs =**16.884** μs

Queuing delay= (16.384+7) $\mu s = 23.884\ \mu s$.

From router A to B = $T_p + T_x = 16.384 + 0.5$ =16.884 μs

Total transfer delay =16.884 μs + $23.884\ \mu s = 40.768\ \mu s$.

Q4: *What are the various types of error correcting techniques? An 8-bit byte with binary value 10101111 is to be encoded using an even-parity Hamming code. What is the binary value after encoding?*

Solution :

Error Correction can be handled in two ways:

- **Backward error correction:** Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.
- **Forward error correction:** In this case, the receiver uses the error-correcting code which automatically corrects the errors.

Check bits are inserted at positions that are powers of 2 i.e. 1,2,4,8,16,32,e.t.c. Data bits are at positions 3,5,6,7,9,10,11,12 e.t.c. So after inserting check bits our data should look like this:

? ? 1 ? 0 1 0 ? 1 1 1 1

positions 1 2 3 4 5 6 7 8 9 10 11 12

3 = 1+2 ,

5 = 1+4 ,

6 = 2+4 ,

7 = 1+2+4 ,

9 = 1+8 ,

10 = 2+8 ,

11 = 1+2+8 ,

12 = 4+8

Hence for the check bit 1 we look at bits 3,5,7,9,11 and get value 1. For check bit 2 we look at bits 3,6,7,10,11 and get value 0. For check bit at position 4 we look at bits 5,6,7,12 and get value 0. For the check bit at position 8 we look at bits 9,10,11,12 and get value 0. Hence the binary value after encoding is

1 0 1 0 **0** 1 0 **0** 1 1 1 1.

Q5: *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 receiver's end.*

Solution:

Answer

Our generator $G(x) = x^3 + 1$ encoded as 1001. Because the generator polynomial is of the degree three we append three zeros to the lower end of the frame to be transmitted. Hence after appending the 3 zeros the bit stream is **10011101000**. On dividing the message by generator after appending three zeros to the frame we get a remainder of 100. We do modulo 2 subtraction thereafter of the remainder from the bit stream with the three zeros appended. **The actual frame transmitted is 10011101100**. See below.

```
              1 0 0 0 1 1 0 0
          ---------------------
1001 |    1 0 0 1 1 1 0 1 0 0 0
     |    1 0 0 1
     -----------
       0 0 0 1
       0 0 0 0
       ---------
         0 0 1 1
         0 0 0 0
         ---------
           0 1 1 0
           0 0 0 0
           ---------
             1 1 0 1
             1 0 0 1
             ---------
               1 0 0 0
               1 0 0 1
               ---------
                 0 0 1 0
                 0 0 0 0
                 ---------
                   0 1 0 0
                   0 0 0 0
                   ---------
                     1 0 0 (remainder)
```

Actual frame transmitted : 10011101000 – 100 = 10011101100 (modulo 2 subtraction)

Now suppose the third bit from the left is garbled and the 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.

```
                1 0 1 0 1 0 0 0
          ---------------------------
1001 |    1 0 1 1 1 1 0 1 0 0 0
     |    1 0 0 1
     ------------
       0 1 0 1
       0 0 0 0
       ---------
         1 0 1 1
         1 0 0 1
         -------
           0 1 0 0
           0 0 0 0
           -------
             1 0 0 1
             1 0 0 1
             -------
               0 0 0 1
               0 0 0 0
               -------
                 0 0 1 0
                 0 0 0 0
                 -------
                   0 1 0 0
                   0 0 0 0
                   -------
                     1 0 0 (remainder indicating error)
```

Q6: *"Pipelined" protocols, also known as "sliding window" protocols, are a type of ARQ*

(Automatic Repeat Request) protocols that use a "window" to control the amount of data they inject into the network.

(a) *Stop-and-Wait is one type of ARQ protocol. What is the window size of Stop-and-Wait? Explain.*

Solution:

In Stop-and-Wait, the window size is 1. By design, in Stop-and-Wait ARQ, only one segment is sent "at a time", i.e., before an acknowledgment is received.

(b) *In the Stop and Wait protocol every 4th packet is lost and we need to send a total of 10 packets so how many transmissions it took to send all the packets?*

Solution:

1 2 3 4 5 6 7 8 9 10 (Initially)
^
1 2 3 4 4 5 6 7 8 9 10 (Packet no. 4 retransmitted)
^
1 2 3 4 4 5 6 7 7 8 9 10 (Packet no. 10 retransmitted)
^
1 2 3 4 4 5 6 7 7 8 9 10 10 (Result)

So, we retransmitted packet number 4, 7, 10

Total count = 13

(c) I*n S&W protocol if Error probability is p and no. of packets to send is 'n'. How many packets do we have to send?*
Total retransmissions

Solution:

= $n*p_0 + n*p_1 + n*p_2 + n*p_3 + n*p_4 + \ldots$

= $n(1 + p + p^2 + p^3 + p^4 + \ldots)$

= $\frac{n}{1-p}$ using infinite GP sum formula

(d) *In GBN sender Window size = 10 and Tp = 49.5ms & Tt = 1ms. What is the Efficiency of the protocol and Throughput given Bandwidth = 1000 bps?*

Solution:

Efficiency = N/(1+2a), N = 10 (given), a = T_p/T_t = 49.5

Efficiency = 10/(1 + 2 * 49.5) = 10/100 = 0.1 or 10%

Throughput = Efficiency × Bandwidth

= 0.1 * 1000 = 100

(e) If there is a K bits sequence no. define require sender window size and receiver window size for S&W, GBN & SR?

Solution:

Given, K bits, For S&W Ws = 1 and Wr = 1

For GBN, Ws = 2^k -1 and Wr = 1

For SR, Ws 2^{k-1} = and Wr = 2^{k-1}

Q7: *For a 100Mbits/sec channel with 100ms propagation delay, what is the channel utilization when sending 2KByte segments if Stop-and-Wait is used? Show your work*

Solution:

$T_p = 100 \times 10^6 bps$

$T_t = L/R = 16000 / 1000000000 = 0.16$ *milliseconds*

utilisation= $\frac{1}{1+2\times(\frac{Tp}{Tt})} \times 100$

$$= \frac{1}{1+2\times(\frac{100}{0.16})} \times 100$$

$$= 1/1251 = 0.08\%$$

Q8: *Consider a token ring with latency 500 μsec and packet size of 1500 bytes. What is the effective throughput rate for both a single active host and for many active hosts that can be achieved if the ring has 3 Mbps bandwidth? Assume the strategy used is delayed token reinsertion.*

Solution :

Given-

- Ring latency = 500 μsec
- Packet Size = 1500 bytes
- Bandwidth = 3 Mbps
- Strategy used is Delayed Token Reinsertion (DTR)

We know,

Transmission delay (T_t)

= Packet size / Bandwidth

= 1500 bytes / 3 Mbps

= (1500 x 8 bits) / (3 x 10^6 bits per sec)

= 4000 μsec

We know,

$a = T_p / T_t$

a = Latency / T_t

a = 500 μsec / 4000 μsec

a = 0.125

Efficiency of Delayed Token Reinsertion (DTR) strategy is-

$$Efficiency(\eta) = \frac{1}{1+(1+\frac{1}{n})^{a}}$$

For a single active host, N = 1.

Substituting N = 1 in efficiency formula, we get-

Efficiency (η)

= 1 / (1 + 2a)

= 1 / (1 + 2 x 0.125)

= 1 / 1.2

= 0.8

Throughput = $efficiency(\eta) \times bandwidth$

= 0.8 x 3 Mbps

= 2.4 Mbps

For many active hosts, N = ∞.

Substituting N = ∞ in efficiency formula, we get-

Efficiency (η)

= 1 / (1 + a)

= 1 / (1 + 0.125)

= 1 / 1.125

= 0.89

Now,

Throughput

= Efficiency (η) x Bandwidth

= 0.89 x 3 Mbps

= 2.67 Mbps

Q9: *What is bit and byte stuffing explain each mechanism? Suppose the following bit string is received by the data link layer from the network layer: 11010111110101111101011111110. What is the resulting string after bit stuffing?*

Solution:

Bit stuffing:

Bit stuffing is the insertion of one or more bits into a transmission unit as a way to provide signaling information to a receiver. The receiver knows how to detect and remove or disregard the stuffed bits.

Mechanism:

In a data link frame, the delimiting flag sequence generally contains six or more consecutive 1s. In order to differentiate the message from the flag in case of the same sequence, a single bit is stuffed in the message. Whenever a 0 bit is followed by five consecutive 1bits in the message, an extra 0 bit is stuffed at the end of the five 1s.

When the receiver receives the message, it removes the stuffed 0s after each sequence of five 1s. The un-stuffed message is then sent to the upper layers.

Byte stuffing:

Byte stuffing is a mechanism to convert a message formed of a sequence of bytes that may contain reserved values such as frame delimiter, into another byte sequence that

does not contain the reserved values.

Mechanism:

If the pattern of the flag byte is present in the message byte sequence, there should be a strategy so that the receiver does not consider the pattern as the end of the frame. Here, a special byte called the escape character (ESC) is stuffed before every byte in the message with the same pattern as the flag byte. If the ESC sequence is found in the message byte, then another ESC byte is stuffed before it.

Bit sequence: 1101011111010111111010111111110 (without bit stuffing)

Bit sequence: 1101011111001011111010101111110110 (with bit stuffing)