Chapter5

12. In Stop-and-Wait ARQ why should the receiver always send an acknowledgment message each time it receives a frame with the wrong sequence number?

Solution:

The sender cannot send the next frame until it has received the ACK for the last frame so, if the receiver gets a frame with the wrong sequence it has to be a retransmission of the previous frame received. This means that the ACK was lost so the receiver has to ACK again to indicate the sender that it has received the frame.

15. A 1 Mbyte file is to be transmitted over a 1 Mbps communication line that has a bit error rate of $p = 10^{-6}$.

Solutions follow questions:

The file length $n = 8 \times 10^6$ bits, the transmission rate $R = 1$ Mbps, and $p = 10^{-6}$.

a. What is the probability that the entire file is transmitted without errors? Note for n large and p very small, $(1-p)^n \approx e^{-np}$.

P[no error in the entire file] $= (1 - p)^n \approx e^{-np}$, for $n >> 1$, $p << 1$

$$= e^{-8} = 3.35 \times 10^{-4}$$

We conclude that it is extremely unlikely that the file will arrive error free.

b. The file is broken up into N equal-sized blocks that are transmitted separately. What is the probability that all the blocks arrive correctly without error? Does dividing the file into blocks help?

A subblock of length n/N is received without error with probability:

P[no error in subblock] $= (1 - p)^{n/N}$

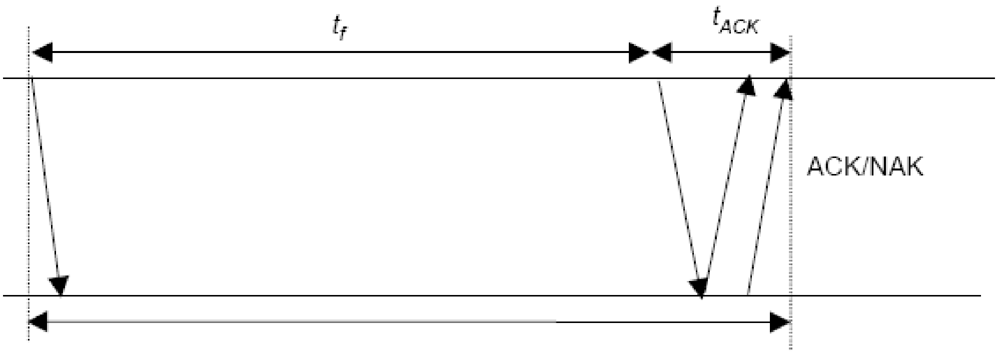
A block has no errors if all subblocks have no errors, so

P[no error in block] $= P$[no errors in subblock]$^N = ((1 - p)^{n/N})^N = (1 - p)^n$

So simply dividing the blocks does not help.

c. Suppose the propagation delay is negligible, explain how Stop-and-Wait ARQ can help deliver the file in error-free form. On the average how long does it take to deliver the file if the ARQ transmits the entire file each time?

Refer to the following figure for the discussion.



We assume the following:

- t_0 = basic time to send a frame and receive the ACK/NAK $\approx t_{timeout}$
- t_{total} = total transmission time until success
- n_f = number of bits/frame
- n_a = number of bits per ACK
- n_t = number of transmissions
- P_f = probability of frame transmission error

$t_0 = t_f + t_{ACK} = n_f/R + n_a/R \quad (t_{prop} \approx 0)$.

$P[n_t = i] = P[\text{one success after } i - 1 \text{ failure}] = (1 - P_f)\, P_f^{\,i-1}$

$t_{total} \mid i$ transmissions $= i.t_0$

$$E[t_{total}] = \sum_{i=1}^{\infty} i t_0 P[n_t = i] = t_0(1 - P_f)\sum_{i=1}^{\infty} iP_f^{\,i-1} = t_0(1 - P_f)/(1 - P_f)^2 = t_0/(1 - P_f)$$

Here, $n_f = n >> n_a$ thus $t_0 \approx t_f = n/R$; and $P_f = 1 - P[\text{ no error}] = 1 - e^{-np}$

$E[\text{total}] = n/R\ (1 - P_f) = n/[Re^{-np}] = 8 / (3.35 \times 10^{-4}) = 23{,}847$ seconds = 6.62 hours!

The file gets through, but only after many retransmissions.

d. Now consider breaking up the file into N blocks. (Neglect the overhead for the header and CRC bits.) On the average how long does it take to deliver the file if the ARQ transmits the blocks one at a time? Evaluate your answer for N = 80, 800, and 8000.

For 1 block $P_f = 1 - P_b = 1 - (1 - p)^{n/N}$ and $n_f = n/N$

if $t_{prop} \approx 0$ and $n_a << n/N$: $t_0^{\,b} = n_f/R = n/NR$

$T_b = E[t_{total}^b] = t_0^b / (1 - P_f) = n(1-p)^{-n/N} / NR$ average time to transmit one block

$T = E[t_{total}] = N\, T_b = n(1-p)^{-n/N} / R = 8\,(1-p)^{-n/N} = 8\, e^{np/N}$ if $n/N >> 1$, $p << 1$

- $N = 80 \Rightarrow T \approx 8\, e^{0.1} = 8.84$ sec
- $N = 800 \Rightarrow T \approx 8\, e^{0.01} = 8.08$ sec
- $N = 8000 \Rightarrow T \approx 8\, e^{0.001} = 8.008$ sec

Each subblock has a higher probability of arriving without errors, and so requires fewer retransmissions to deliver error free. The overall delay is reduced dramatically.

e. Explain qualitatively what happens to the answer in part (d) when the overhead is taken into account.

As N increases, the effect of overhead becomes more significant because the headers constitute a bigger fraction of each subblock.

33. A telephone modem is used to connect a personal computer to a host computer. The speed of the modem is 56 kbps and the one-way propagation delay is 100 ms.

Solutions follow questions:

a. Find the efficiency for Stop-and-Wait ARQ if the frame size is 256 bytes; 512 bytes. Assume a bit error rate of 10^{-4}.

First we have the following:

$P_f = 1 - (1 - 10^{-4})^{n_f}$
$n_f = 256 \times 8 = 2048$ or $n_f = 512 \times 8 = 4096$
$t_{prop} = 100$ ms

$n_o = 0$
$n_a = 64$ bits
$t_{proc} = 0$

Using the results in Equation 5.4,

$$\eta = (1 - P_f)\frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})}{n_f}R}$$

= 0.125 ($n_f = 2048$)
= 0.177 ($n_f = 4096$)

b. Find the efficiency of Go-Back-N if three-bit sequence numbering is used with frame sizes of 256 bytes; 512 bytes. Assume a bit error rate of 10^{-4}.

Given that $W_S = 2^3 - 1 = 7$, we can calculate that the window size is:

$$\frac{n_f \times W_S}{R} = 256ms$$

Since this is greater than the round trip propagation delay, we can calculate the efficiency by using the results in Equation 5.8.

$$\eta = (1 - P_f)\frac{1 - \frac{n_o}{n_f}}{1 + (W_S - 1)P_f}$$

= 0.385 (n_f = 2048)

= 0.220 (n_f = 4096)

52. Perform the bit stuffing procedure for the following binary sequence: 1101111111011111110101.

Solution:

The inserted stuff bits are underlined.

1101111111011111110101 → 1101111<u>0</u>1101111<u>0</u>110101

54. Consider the PPP byte stuffing method. What are the contents of the following received sequence of bytes after byte destuffing:

0x7D 0x5E 0xFE 0x24 0x7D 0x5D 0x7D 0x5D 0x62 0x7D 0x5E

Solution:

<u>0x7D 0x5E</u> 0xFE 0x24 <u>0x7D 0x5D</u> <u>0x7D 0x5D</u> 0x62 <u>0x7D 0x5E</u>

→ 0x7E 0xFE 0x24 0x7D 0x7D 0x62 0x7E

Chapter8

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

50 .156.10.10
00110010 .10011100.00001010.00001010
Class A

250.10.24.96
11111010.00001010.00011000.01100000
Class E

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**

8.9. 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.