

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, $(1 - p)^n \approx e^{-np}$.

$$\begin{aligned} P[\text{no error in the entire file}] &= (1 - p)^n \approx e^{-np}, \text{ for } n \gg 1, p \ll 1 \\ &= e^{-8} = 3.35 \times 10^{-4} \end{aligned}$$

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:

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

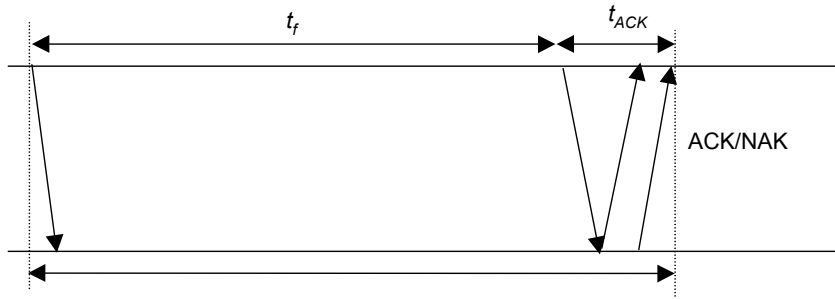
A block has no errors if all subblocks have no errors, so

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

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 $\approx t_{\text{timeout}}$
- 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

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

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

$$t_{\text{total}} | i \text{ transmissions} = i \cdot t_0$$

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

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

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

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

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

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

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

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.

18:

Message Size	65536 bytes
Max Packet Size	2048 bytes
Packet Header	32 bytes
Available for info	2016 bytes
# of packets needed	32.51 packets
Total	33 packets

bit error rate	1E-06	
bits/packet	16384	
Probability of error in packet	0.016251	$1 - (1 - \text{bit_error_rate}) ^ (\text{bits/packet})$
Propagation speed	2E+05 Km/s	
Distance	1000 Km	
Bandwidth	1.5 Mb/s	

We assume that the ACK error, the ACK time, and processing time are negligible.

$T_{\text{prop}} = \text{distance} / \text{propagation speed} = 0.0050 \text{ s}$
 $T_f = \text{packet size} / \text{bandwidth} = 0.0109 \text{ s}$
 $T_0 = T_{\text{prop}} + T_f = 0.0159 \text{ s}$
 $P_f = \text{probability of error in packet} = 0.016251$

$$E[T_{\text{total}}] = T_0 / (1 - P_f) = 0.0162$$

There is pipelining effect that occurs as follows: After the first packet arrives at switch 1, two transmissions take place in parallel. The first packet undergoes stop-and-wait on the second link while the second packet undergoes stop-and-wait in the first link. The packet arriving at the switch cannot begin transmission on the next link until the previous packet has been delivered, so there is an interaction between the transmission times of the two packets. We will neglect this effect. The time to send every packet over two links is then the initial packet transmission time + 33 additional packet times, and so the average time is $E[T_{\text{total}}] * 34 = 0.522 \text{ seconds}$.

32. Suppose that Selective Repeat ARQ is modified so that ACK messages contain a list of the next m frames that it expects to receive.

Solutions follow questions:

- a. How does the protocol need to be modified to accommodate this change?

First, the frame header needs to be modified to accommodate the list of frames to receive. It can be a fixed or a variable number of slots. NAK won't be necessary because the receiver explicitly indicates which frames need to be transmitted. Second, the transmitter operation must change to retransmit frames according to the received list. If the list contains the m oldest frames that are yet to be received, then the list can be used to skip retransmissions of frames that have already been received.

- b. What is the effect of the change on protocol performance?

The performance will increase in cases with high error rate or in cases where the delay is high. A single frame can ask for the retransmission of several frames. The drawback is the overhead in the header and the increased protocol complexity relative to pure Selective-Repeat ARQ.

