Solution to Exercises From 18 Feb 2004

1 DLC using stop-and-wait flow control is used with a half duplex 448 kbps link to send frames 512 bytes in length. This link is derived by TDM, multiplexed from a broadband fiber connection 1600 miles in length. ACK send time and processing time for both send and receive are negligible. What fraction of the 448 kbps link capacity is utilized?

The efficiency of stop-and-wait protocol is shown on slide 05.29 as: \[ \text{Efficiency} = \frac{\text{useful sending time}}{\text{total time}} \]

which is equal to

\[ \frac{(2 \times T_F)}{(2 \times T_F) + (2 \times D) + P_F + P_A} \]

\( T_F \) is the time to ‘put the bits on The Wire’, so is determined by the number of bits and the bit rate:

\[ T_F = \frac{512 \text{ bytes} \times 8 \text{ bits/byte}}{448000 \text{ bits/sec}} = 0.0091 \text{ sec.} \]

The link propagation delay, \( D \), is just distance/\( \nu_{\text{medium}} \), giving us:

\[ D = (1600 \text{ miles} \times 1609 \text{ m/mile}) / (3 \times 10^8 \text{ m/sec}) \times 0.7 = 0.0123 \text{ sec.} \]

Note we multiply \( c \), the speed of light in vacuum, by 0.7 to give us a reasonable approximation to \( c \) in a wire.

We are told to ignore the processing times, \( P_F \) and \( P_A \) as negligible. Hence, we determine the utilization of the link to be:

\[ \text{Efficiency} = \frac{(2 \times T_F)}{(2 \times T_F) + (2 \times D)} = \frac{(2 \times 0.0091)}{(2 \times 0.0091) + (2 \times 0.0123)} = 0.425 \]

Our link is 42.5% utilized.

2 Repeat the previous problem assuming use of a 448 kbps satellite link with a go-back-N protocol and a full duplex connection with window size 7 frames. Assume no frames are lost due to errors.

From slide 05.50, we note that the efficiency of an error-free link — which we are told this is — is the minimum of \( \{(W \times T_F) / S, 1\} \). We are given that \( W = 7 \), the window size. \( T_F \) has the same value as in the solution to question 1, above, since we have the same framesize and bit rates.

What’s different is the distance: we have to get up to and back from a satellite which is orbiting at a distance of 35,880 km (22,300 miles).

\[ D = (2 \times 35,880,000 \text{ m}) / (3 \times 10^8 \text{ m/sec}) = 0.239 \text{ sec.} \]

Note here we do not reduce \( c \); while some of the travel is done through atmosphere, much occurs in the vacuum of space.

Now we have to work out the timeout, \( S \) which is simply the time to ‘spit bits’, \( T_F \), and the time for the frame to get to and some ack back from our destination, or,
\[ S = T_F + (2 \times D) \]
\[ = 0.0091 + (2 \times 0.239) = 0.487 \text{ sec.} \]

The efficiency, then, is the lesser of 1 or

\[ \frac{(W \times T_F)}{S} \]
\[ = \frac{(7 \times 0.0091)}{0.487} \]
\[ = 0.131 \]

This link’s utilization is 13.1%. Note that this is essentially the best we can hope for since the timeout would, in reality, be slightly larger than the round-trip time \( S \) we computed above, making the denominator larger in the efficiency fraction, and hence reducing the efficiency slightly.

3. Calculate the window size \( W \) needed for 100\% link usage in 2.

We determine this by simply rearranging the expression for efficiency, which we set to 1, to get

\[ W = \frac{S}{T_F} \]
\[ = \frac{0.487}{0.0091} = 53.52 \]

So we conclude that if we set the window size to 54 we should have 100\% utilization of our (error-free) link.

4. Now calculate the link efficiency in 3 if, due to noise in the satellite channel, there is a probability of .12 that any given frame will be lost due to transmission error.

Reality sets in at last: we have to take into account errors due to noise. Again referring to slide 05.50, we see that we use a different expression for efficiency for channels in the presence of errors,

\[ Efficiency = \frac{1}{1 + \left(\frac{p}{(1-p)}\right) \times W} \]
\[ = \frac{1}{1 + \left(\frac{0.12}{(1-0.12)}\right) \times 54} \]
\[ = \frac{1}{1 + (0.136 \times 54)} \]
\[ = \frac{1}{8.36} = 0.12 \]

With \( p = 0.12 \), efficiency on this link drops to 12\%.

Note that the next two problems were not assigned as part of the homework, i.e., you did not have to hand them in. They were suggested exercises to help you plan your solution to the project, and so the solutions appear below.
Refer to Network Workbench module code/dl.cpp. This module contains the algorithms for a reliable DLC using go-back-N. Function dl_send operates in three distinct states: waiting, sending, sending_supv. Link state variable dl_send_state indicates which state it will enter when called. Create a state transition diagram showing what conditions cause transitions among these states. Label each transition arc with a simple description of the reason for the transition. Hint: look in nw.h for definitions and init.cpp for initial state.

To see the sending algorithm explanation, refer to slides 05.85 — 05.87 as well as the source code the question directs you to look at. You should be able to arrive at the following state transition diagram, figure 5.8 in Understanding Internet Protocols:

Refer to Network Workbench module dllogic.cpp. This module contains the decision logic that drives dl.cpp. Near the beginning of the module there is a function stack::LTwindow(byte Nmin, byte Nmax) which returns TRUE if the range between Nmin and Nmax is less than the window size for the DLC, implying there is room for another frame. thereis also a stub for function stack::INwindow(,). Complete the code so this function is TRUE when the range is within the window size, that is, if the range between Nmin and Nmax does not exceed the window size. Hint: INwindow is very similar to LTwindow.

Dr. Pullen provides this function:

```
int testmax = Nmax;
if (Nmin > Nmax) testmax = testmax - DL_WINDOW_MAX;
return(testmax - Nmin <= DL_WINDOW_FRAMES);
```