COMPSCI 314 S2T Assignment 2
2009
Flow Control

Department of Computer Science
The University of Auckland

*This assignment contributes **5%** of your overall course mark. Submit your assignment in **PDF** format to the **Assignment Drop Box**. Include all **workings** and **explanations**. Marks will be deducted for ambiguous solutions. Zero marks are awarded if the answers contain no explanation. Also, refer to the Departmental Policy on Cheating on Assignments.*

Assignment Drop Box *(*https://adb.ec.auckland.ac.nz/adb/*).*
Departmental Policy on Cheating on Assignments *(*http://www.cs.auckland.ac.nz/CheatingPolicy.php*)*

[Total: 50 marks]

Sample Solution for markers

Q1. Stop and wait. [20 marks]

Consider a data link from Auckland to Invercargill, using an optical fibre connection. Assume the distance is 1200 km ($1.2 x 10^6$ m), and that the speed of an optical signal is $2x10^8$ m/s. Assume that the transmission capacity of the link is 10 Gb/s (10^{10} bits/s), and that the frame size is 1000 bytes.

a) What is the one-way delay for a data frame to travel between Auckland and Invercargill? [5 marks]
$(1.2 \times 10^6) / (2 \times 10^8)$ seconds = 0.006 seconds = 6 milliseconds

b) Do we also need to consider the time taken for a computer to output a frame at 10 Gb/s? [5 marks]
1000 bytes = 8000 bits at 10^{10} bits/s = 8×10^{-7} seconds = 0.8 microseconds.
Negligible, do not need to consider this *relative* to the one-way delay in Q1a.

[Students need to justify why it is negligible to gain a full mark]

[Note: The answer holds true only for a single frame: Q1d and Q2a should clarify that sending a series of N frames will start to add up some time.]

c) Consider a stop-and-wait protocol sending 1000 byte frames and waiting for an ACK after each frame. Assuming no frames or ACKs are lost, calculate the achieved bit rate in b/s and the efficiency of the protocol. [5 marks]

This is the same as the sliding window with N=1.
Round trip time (RTT) is 12 milliseconds, so we can send 8000 bits every 12 milliseconds, that is 8000 / 0.012 = 666666.66667… = 666,667 bits/s.
Note that this ignores an initial 'time to send' frame of both 1000 byes (0.8 microseconds) and 64 bytes (0.0512 microseconds), i.e. 8000 / 0.0120008512 = 666619.3811319… = 666,619 bit/s.

The efficiency: divided into 10 Gbit/s, about 0.0067%.

d) Calculate the bandwidth-delay product for the link, in megabytes. [5 marks]
10^{10} bits/s x 0.006s = 6 x 10^7 bits = 6 / (8 x 10^7 bytes) = 7.5 megabytes.

Q2. Fixed window [20 marks]

For the same link, now assume that a "window" of N frames is allowed on the link at one time.

a) What is the efficiency for N=10, N=1000, N=10,000, N=100,000 ? [10 marks]
N=10, 0.067%
N=1,000, 6.67%
N=10,000, 66.67%
N=100,000, 666.67%

[Anyone who says that efficiency is 667% needs to think again, it is not possible to utilize more than 100%, i.e. > 10Gb/s]
[If N is large, the rest of the slots would be empty and unused, but it could be useful for taking into account of errors/retransmissions.]
Two cases can be considered: unrestricted and window-oriented.
An easiest way to check this is to relate the total 'time to transmit' with the RTT. When it is greater than or equal to RTT, then we know that there is 'no waiting' time – we can achieve 100%.

In Q1b, we find that it takes 0.8 microseconds to transmit a single DATA (1000B) frame. So, when N=10, it takes 8 microseconds and so on. When N=10,000, it takes 0.008s to transmit the series of (DATA) frames. As for ACK (64B), when N=10, it takes 0.512 microseconds. When N=10,000, it takes 0.000512s to transmit the series of (ACK) frames.
Here, when N=100,000, it takes **0.08s** to transmit the series of (DATA) frames. This is bigger than what we've found as the round trip times, 0.012s – this means that we'd receive the ACK frames back while *still* sending out from buffer. The important point is that, we are limited by the (fixed) RTT, which is 0.012s – we shouldn't spend more time than the RTT to send frames!

The following shows the total 'time to transmit' for both DATA and ACK frames.
When N=10, 8.512 microseconds
When N=1,000, 0.0008512s
When N=10,000, 0.008512s (This means that for every N=10,000 sent, it needs to wait (idle) for about 0.012 – 0.008512 = 0.003488 seconds.
When N=100,000, 0.08512s (as mentioned, this value is > 0.012s)

[Note, it is also plausible to set the user-data (e.g. 7500 bit per frame), such as in the Asg-1 Q4d. Also, the effect of using small frame vs large frame sizes do impact on the overall efficiency ratio, e.g. larger the frame size, less the N, and more the efficiency ratio. Similarly, smaller the frame size, more the N, and less the efficiency ratio. This also relates with Q3b.]

c) For a sliding window protocol to use this link with high efficiency, what is a suitable window size (measured as a number of frames)? [5 marks]
Twice the BDP would be good. That is, N = 12 x 10^7 bits / 8000 bits = 15000 frames. In this case the efficiency = 100%. This is related with Q2a, when N=15,000 we can achieve 100%. Approx. frame numbers are okay.

d) Assume the protocol has a go-back-N feature. What will happen to the efficiency if one packet is lost? [5 marks]
It will not be ACKed so the window will stop and nothing will be transmitted until the receiver tells the transmitter to go back and restart. So the efficiency will be zero until this happens. [Or you could say that the efficiency will drop for a while and then build up to 100% again. There's no need to do a complicated calculation.]

Q3. Shared link [10 marks]

We have assumed so far that there is a single user on the link. Now assume there are many users.

a) If a large number of users are sharing the link, should they each use the window size you found in Q2 c). Explain your answer. [5 marks]
No, they should use a smaller size, since each one will have a small share of the link capacity so the observed BDP per user will be smaller, e.g. BDP / number_of_users

b) If a user's frames are not all the same length, should their window size be based on the smallest, the average, or the largest packet in use? Explain your answer. [5 marks]
The average. What really matters is the transmission time of the total number of bits or bytes in the window, not the number of frames or their individual sizes.
15MB (twice the BDP) can be utilized 100% in various ways by having the DATA frame sizes different, e.g. mixture of both 1000 byte and 1500 bytes – we could, say, send 6667 frames of 1500 bytes (0.008s) and 5000 frames of 1000 bytes (0.004s). Both adds up to 0.012s, the RTT.

[Note, students mentioning either small / large need to explain thoroughly why and how, to gain some marks]
