– Lab1 –
Sliding Windows

1 General Instructions

In this lab you will simulate an existing protocol stack, draw diagrams, and make conclusions. The goal is that you should get an understanding for *sliding windows* protocol. The lab can also be seen as a preparation for Lab 3, where you among other things will implement *sliding windows*.

This assignment can be done in groups of two or three students. If you work in a group of three, in Lab 3 you have to design and implement *both* Go-Back-N and Selective Repeat. All students in the group must participate actively in the design, implementation and demonstration of the solutions.

Be sure to be well prepared prior to the scheduled lab and use the time to ask questions and to demonstrate your solutions.

A Linux-based tool, *mmls*, simulating the sliding window protocol, is provided for this lab. All students must be active and be prepared to **motivate** the answers to the lab assistant. **Note that you will not be approved by just showing the simulation results, you must be able to clearly motivate and draw conclusions from your simulation results.**

Please read through the entire document **before** you begin with the lab.

Deliverables

- An oral demonstration to the lab assistant.
- Hand in the lab specification with your answers to the lab assistant (in connection to the oral demonstration).
- Upload a scanned version of the report (containing your name and student ID) to Canvas. Note that **each** student should do the upload as grading in Canvas is done individually for every student even if the work is done in groups.
- Please check the deadline on Canvas. After this date, demonstration of Labs 2-3 will be prioritized. Lab 1 can be presented if there are free time-slots in the next lab occasions, or during the next instance of the course.

Arafat Sulaiman

Name (student 1)

Asn20020

User ID (e.g., abc01234)

Jacob Johannson

Name (student 2)

jjn20030

User ID (e.g., abc01234)

Name (student 3)

User ID (e.g., abc01234)

2022-04-06

Date

2 Introduction

In this lab you will simulate a sliding window protocol. When performing simulations, you should make notes and later analyze the results to draw conclusions. The goal of the lab is to get a deeper understanding of the sliding window protocol and data communication in general.

Depending on the Linux type, i.e., 32-bit or 64-bit, you should run one of the two provided simulation files (mmls32 or mmls64). To check the Linux type go to System Settings -> Details -> OS type.

The simulation tool is self-explanatory. You can set the following parameters for the protocol in the simulation: frame size in bytes, link speed in kbps, number of frames, propagation delay in ms, and window size in number of frames. In order to set the parameters, you should use a command in a Linux terminal window. An example of running the simulation is:

./mmls64 -f 20 -l 100 -n 100 -p 10 -w 16

Which means 20 bytes for the frame size, 100 kbps for the link speed, 100 frames, 10 ms propa- gation delay, and 16 frames for the window size.

The result in the terminal is the time to send the whole 100 frames. The result for the above example is:

Frame 100: ACK received at time 196.160 ms from start.

Note: When running the simulator you might get a permission problem by Linux. In this case, you need to change the mode of the runnable file by typing the following command in the Linux terminal: **chmod +x mmls64** or **chmod +x mmls32**

3 Sliding Windows

You should start by measuring the effectiveness of a sliding window. Use the following basic settings: *frame size* = 20 bytes, *propagation delay* = 10 ms, *bit speed* = 100 kbps, and *window size* = 1 (which is the same as a "stop-and-wait" protocol).

Measure the time to send 100 frames from node #0 to node #1 with different window sizes and write down the results in Table 1. The time it takes to send 100 frames is shown in the Linux terminal.

Window size Frame size= 20 bytes, Bit speed=100kbps, Propagation delay=**10ms**	**Time to send 100 frames (ms)**
1	2192
2	1097.76
4	553.280
8	290.240
16	196.160
32	196.160
64	196.160

Table 1:

Next, plot your measured values in the diagram (Figure 1) below.

Also, mark in the diagram in which part that the transfer speed is limited by the window size.

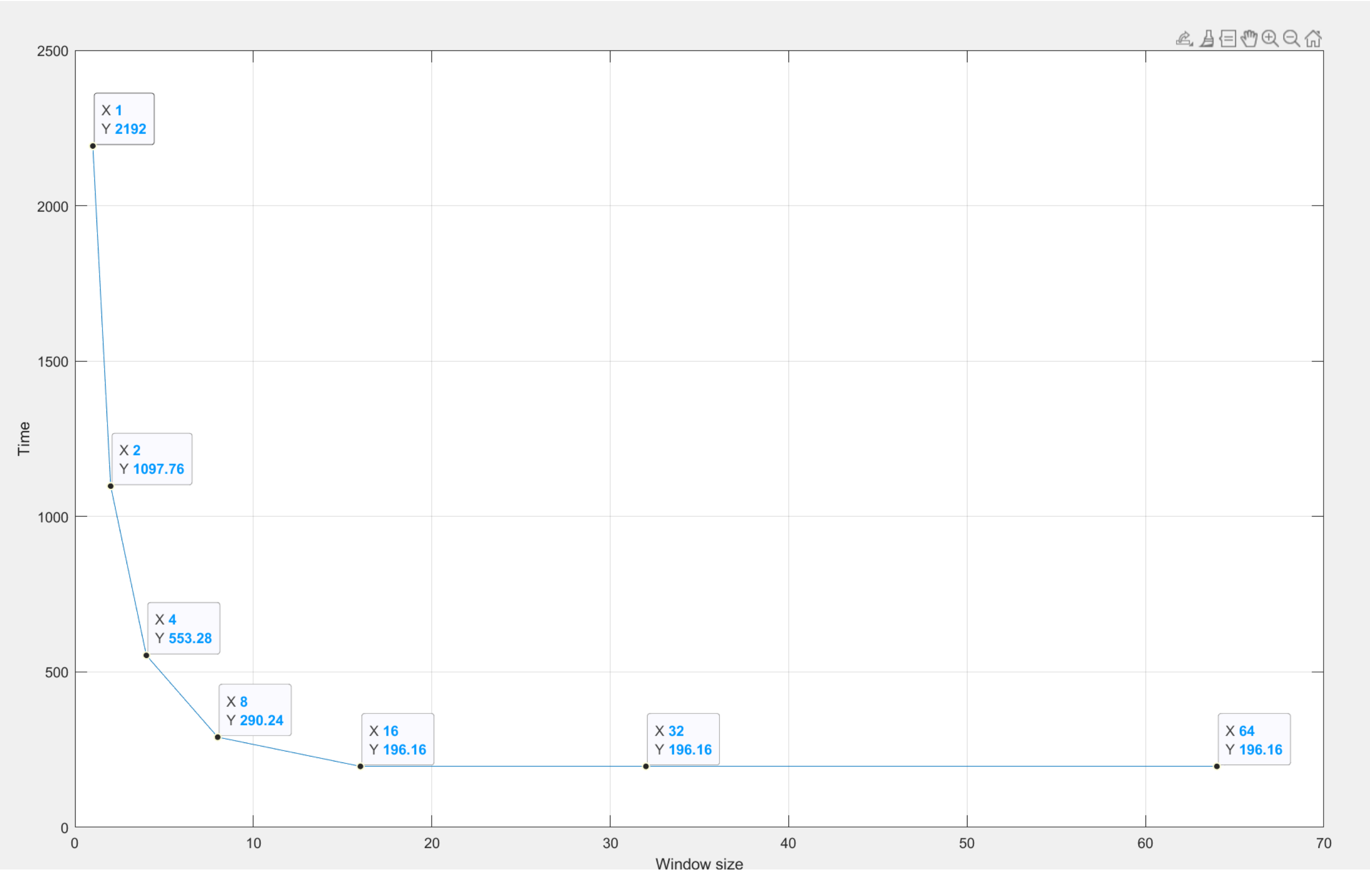

Figure 1: Diagram (Propagation delay=**10ms**)

Are the measured values what we could expect?

It is always suitable to make an estimation about the expected result. We start by looking at a "stop-and-wait" protocol, i.e., when W=1. 100 frames, where each consists of 20 bytes, is 2000 bytes data. The bit speed is 100kbps, which means that each frame takes less than 2ms to send. Note that it will take some time to handle each packet (from node to medium), e.g. packet creation, queue-transfer, medium-encoding etc. This handling time (from node to medium) does **not** include the time it takes for a packet to reach its destination through the medium. The delay from node to node is known as the *propagation delay* and it depends on the length and the type of the medium (*bit-speed* and *propagation delay* are illustrated in Figure 2 below). However, we can approximate the handling time with zero (it is small when comparing to the propagation delay T_P). A frame can be sent and an ACK received in $2 * T_P = 20$ms. It should take approximately $100 * 2 * T_P$ to send all frames, which is 2000 ms. This value should (hopefully) be similar to your measured value.

Now measure how fast this transfer can be done if the window size is so big that maximum speed can be achieved. Set window size to 100. Sending 2000 bytes takes a little less than 200 ms to send if the *bit-speed* is 100kbps. This value should also (hopefully) be similar to the measured value.

Now increase the propagation delay T_P to the double, i.e., 20ms. Redo all measurements with the new setting and write the results in Table 2 below.

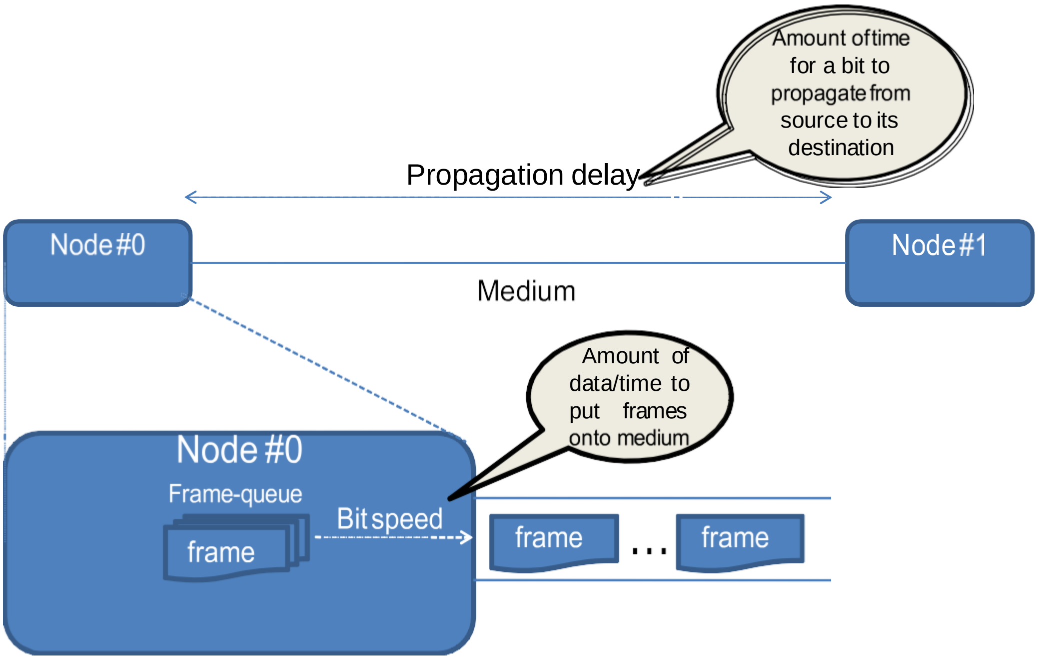

Figure 2: Bit speed and propagation delay

Window size Frame size =20 bytes, Bit speed=100kbps, Propagation delay=**20ms**	**Time to send 100 frames (ms)**
1	4192
2	2097.76
4	1053.28
8	550.24
16	298.72
32	216.16
64	216.16

Table 2:

Before you plot the measured values in the diagram below (Figure 3), try to guess how the diagram should look like, and why!

Start by drawing the curve from the previous measurements (where the propagation delay was 10ms) so that you have something to compare the new curve with. Next, add the new values (where the propagation delay T_P was 20 ms).

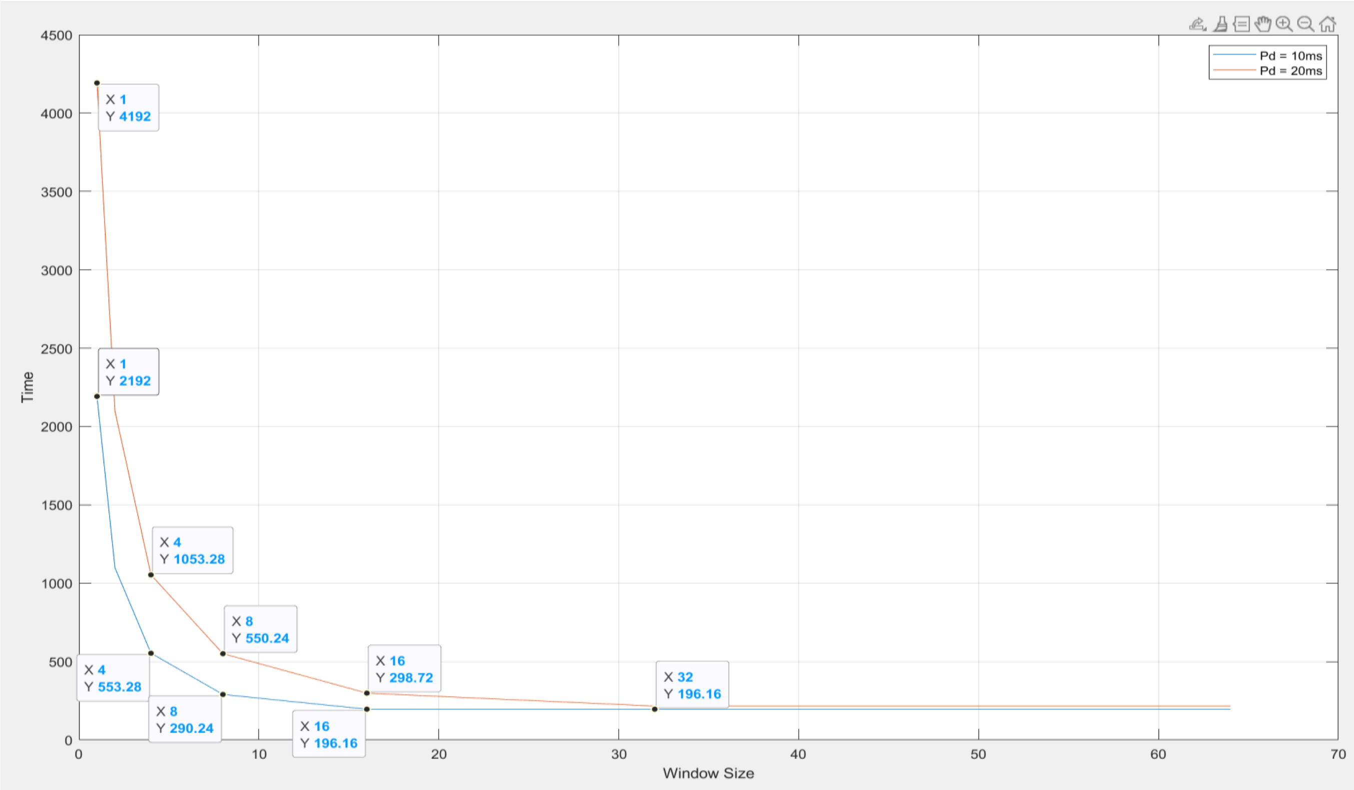

Figure 3: Diagram (Propagation delay=**20ms**)

Explain (in terms of *bit speed*, *propagation delay* and *window size*) the appearance of your plotted curves. Why does the curve suddenly change shape, i.e. why does the transfer time decrease and why does it stabilize suddenly (and not decrease anymore)? Motivate why the curve stabilizes at that particular window size.

When the window is > 8, the sender have time to get an ACK back before sending all its frames constrained by the window size to the receiver.

$$(Roundtrip\ time)\ RTT(1) = 2Tp + \frac{16}{Bitspeed}$$

$$(Linkspeed)\ L(windowSize) = \frac{176 * windowSize}{Bitspeed}$$

$$(WindowSize)\ W \leq \frac{2 * Tp * Bitspeed + 16}{176} \quad (Limited\ by\ WindowSize)$$

With Bitspeed/linkspeed = 100kbps, Tp = 10ms, Framesize=20bytes, Wmax=12.5. Which means that the maximum number of frames on the link is 12.5 frames after that it's constrained by the link speed. Therefore the curve stabilizes.

Setup the following values: *frame size*=20, *window size*=16, *bit speed* =100kbps, and *propagation delay*=10ms.

How much can the *propagation delay* be increased before the transfer speed is limited by the window size? It can be difficult to determine the exact point since it a gradual change, but you can use the point where the transfer time has increased by 10%. At some point, the transfer time should increase dramatically, explain why it happens at this point (i.e. at this propagation delay)? The answer should be motivated in terms of *window size*, *bit speed* and *propagation delay*. Please construct a formula showing the connections between the above mentioned parameters.

- Propagation Delay affects the turnaround time.
- At propagation delay around 14ms, the transfer time increased by 10% at the time 214.72ms, the sender has sent all the frames constrained by the window size before receiving the next ACK.

This is explained by the formula: $Tp = \frac{Wmax * (Framesize * 8 + 16)}{2 * Bitspeed}$

$$(Roundtrip\ time)\ RTT(1) = 2Tp + \frac{16}{Bitspeed}$$

$$(Linkspeed)\ L(windowSize) = \frac{176 * windowSize}{Bitspeed}$$

$$(Propagation\ Delay)\ Tp \geq \frac{176 * windowSize - 16}{2 * Bitspeed} \quad (Limited\ by\ Propogation\ Delay)$$

How much can the *bit speed* be increased before the transfer speed is limited by the window size? Use a fixed propagation delay at **10ms**. The total transfer time will off course be decreased when you increase the bit speed, but the transfer time will, at some point, (almost) not be decreased anymore. Why does the transfer time stabilize at this bit speed? The answer should be motivated in terms of *window size*, *bit speed* and *propagation delay*.

The bit speed can be calculated using the following formula

$$Bitspeed \geq \frac{176 * windowSize - 16}{2 * Tp}$$

When the bitspeed is around 140kbps, it’s going to be limited by the window size.

Since the bitspeed is going to be enough for all the frames the transfer time stabilizes.

Setup the following values: *window size*=16, and *bit speed* =50 kbps. Measure the time it takes to send 100 frames for different values of T_P (see Table 3 below). Also calculate the measured time in effective transfer speed and plot the values in the diagram (Figure 4).

Propagation delay Error probability=**0%**, Bit speed=50kbps, Window size=16	**Time to send 100 frames** (in milliseconds)	**Effective transfer speed** (2000 bytes data = 16000 bits)
1 ms	354.32	45156.92
5 ms	362.32	44159.86
10 ms	372.32	42973.79
15 ms	382.32	41849.76
20 ms	392.32	40783.03
25 ms	402.32	39769.34
30 ms	457.44	34977.26
35 ms	527.44	30335.20
40 ms	597.44	26780.93

Table 3:

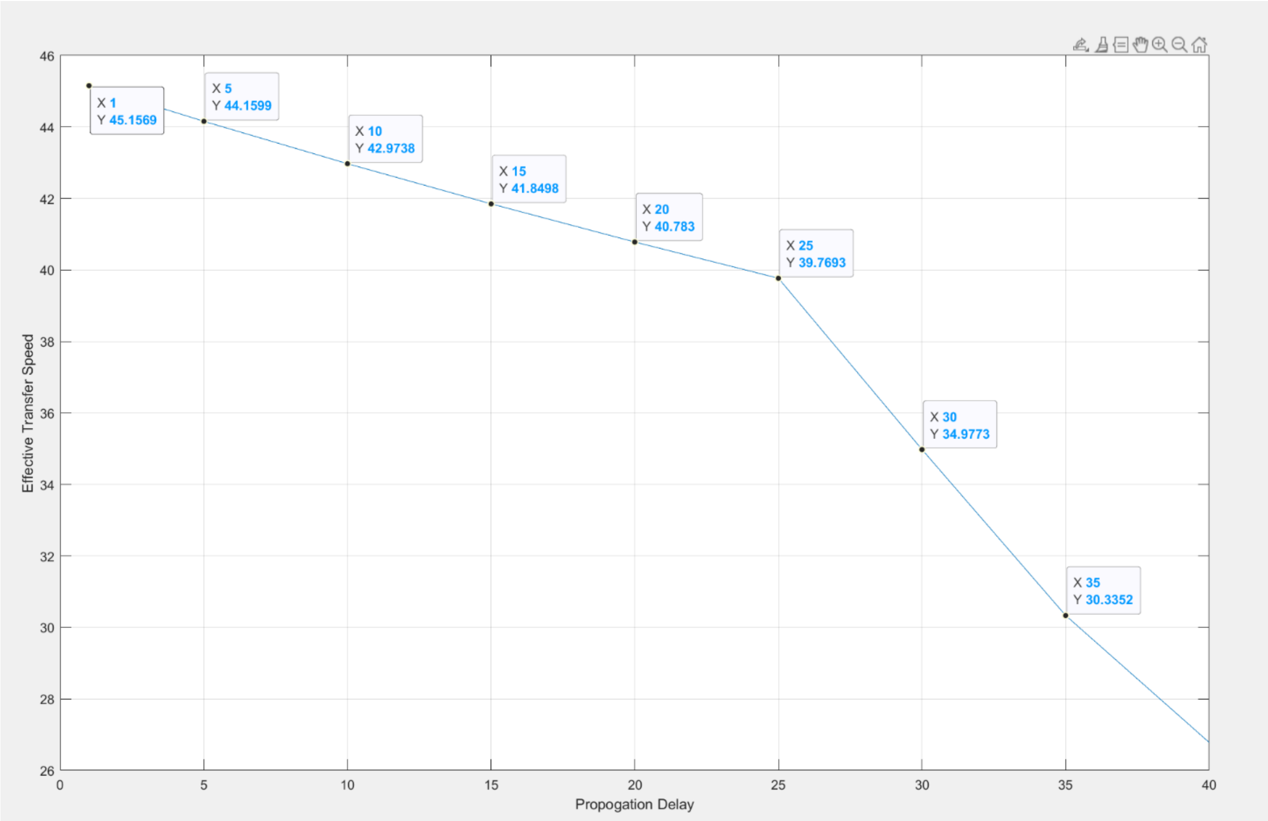

Figure 4: Diagram (Effective transfer speed)

If everything is correct you will clearly see when the window size is insufficient to maintain maximum transfer speed. At approximately which propagation delay does this occur? Motivate your answer in terms of *window size*, *bit speed* and *propagation delay*.

At propagation delay 25 ms the bit speed (50 kbps) is high enough for the sender to send all the frames and filled up the window with the window size 16 before receiving the next ACK which means that the send must wait and therefore the transfer speed limited.