**Nanyang Technological University**

Name: Ang Hao Jie

Matriculation No.: U1721621G

Lab Group: TS6

Course: CZ3006 Net-Centric Computing

Assignment 1 Report

**Implementation**

**Full-duplex data communication**

A full-duplex data communication is used in this assignment where are two separate simplex data channel. Thus, data can be transmitted in both directions on a single circuit at the same time.

In this assignment, Virtual Machine 1 (VMach 1) and Virtual Machine 2 (VMach 2) can both send and receive data at the same time (shown in Figure 1a, 1b, and 1c). When VMach 1 send data in one channel, it can also simultaneously receive data in another channel. So does VMach 2. A field, known as ‘kind’, is in the frame header to differentiate data from acknowledgement (assignment shown in Figure 1d).

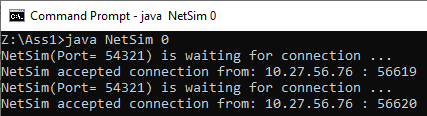


Figure 1a

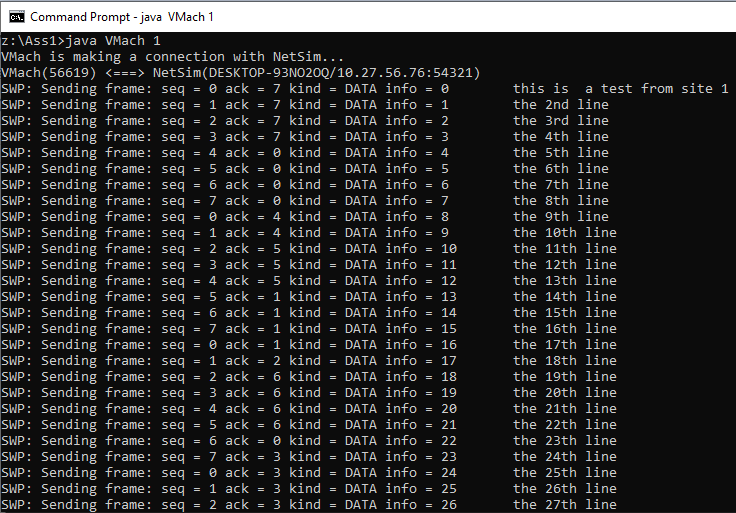


Figure 1b

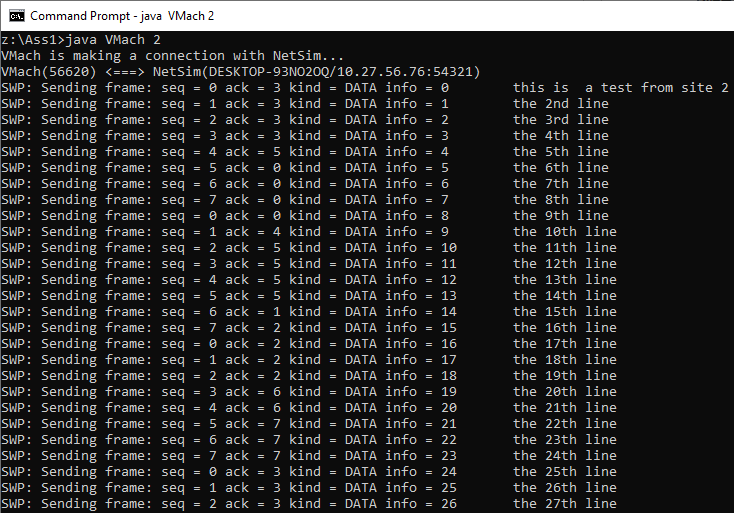


Figure 1c

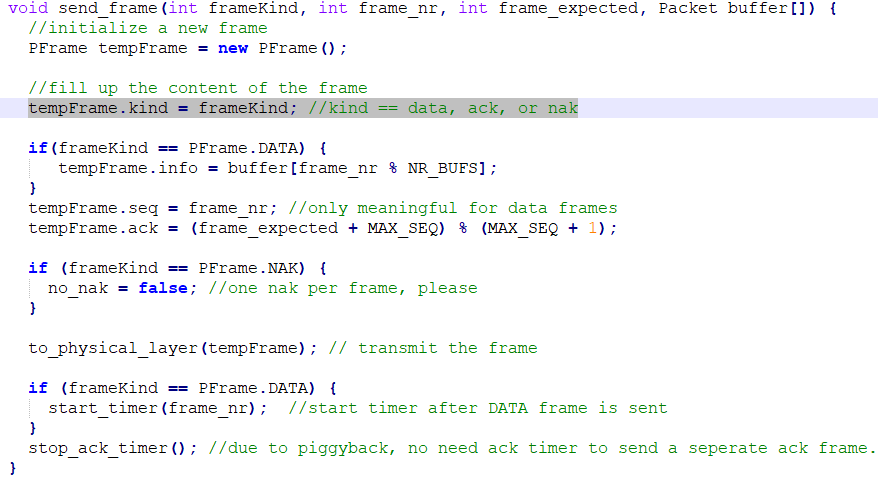


Figure 1d

**In-order delivery of packets to the network-layer**

In the event of FRAME\_ARRIVAL, if it is a data frame, the data will be stored in a buffer. The frames can arrive the physical-layer in random order as they will all be stored in the in\_buf[] buffer in-order. To deliver packets to the network-layer in-order, it is ensured by having a WHILE loop with a condition that is only true when the frame for lower edge of the sliding window has arrived undamaged (in Figure 2). In the WHILE loop, packets stored in in\_buf[] buffer will be delivered to the network layer in order.

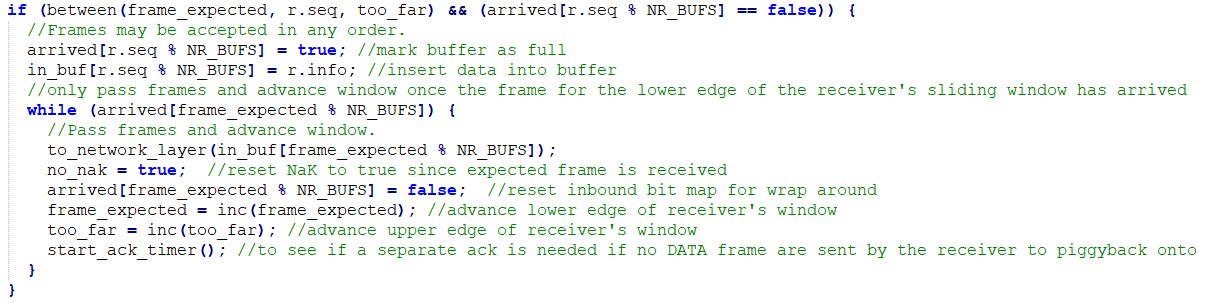


Figure 2

**Selective repeat retransmission strategy**

The difference between Go-Back-N strategy and selective repeat retransmission strategy is selective repeat retransmission strategy allows receiver to accept frames in random order or even accept frames that are not the expected frame, lower edge of the sliding window. This is accomplished by having an incoming buffer that stores the packets in-order before sending them to the network-layer (in Figure 3). When the expected frame, at the lower edge of the sliding window, is not received, a Negative Acknowledgement will be sent to the sender for retransmission of the frame.

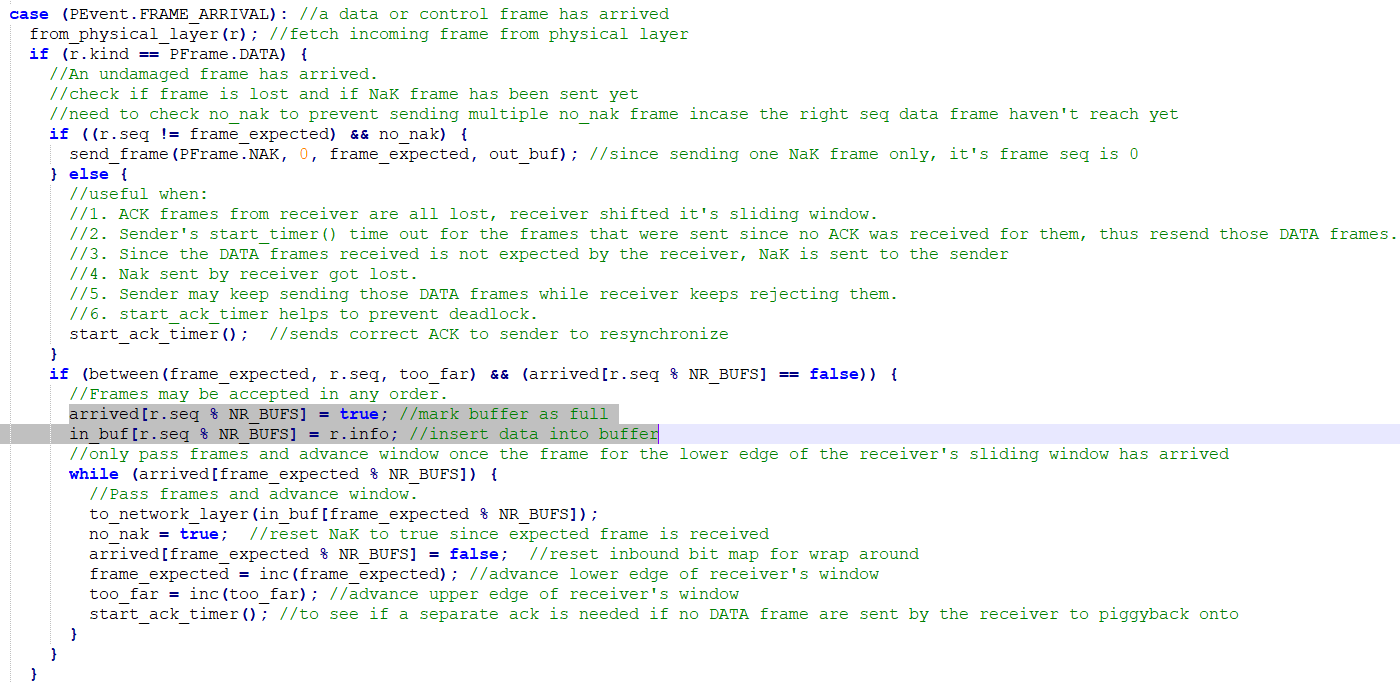


Figure 3

**Synchronization with the network-layer by granting credits**

During enabling of network-layer, the sliding window size is passed to enable\_network\_layer() (in Figure 4a). In the enable\_network\_layer(), the sliding window size sets the number of credits granted in the network-layer so that when the number of frames sent exceeds window size, the network-layer will be disabled (in Figure 4b).

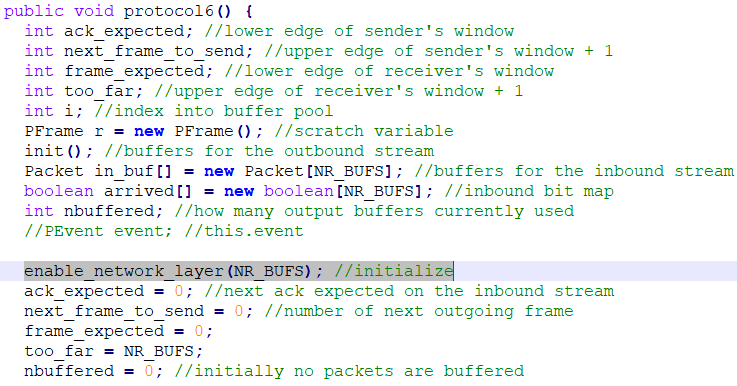


Figure 4a

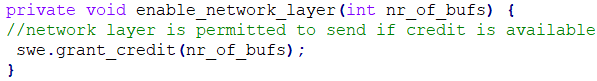


Figure 4b

**Negative acknowledgement**

During transmission of data frames from sender to receiver, frames may be lost or damaged. When a frame is lost, an IF statement will check if the frame received is expected and if no recent Negative Acknowledgement NaK frame has been sent (in Figure 5a and 5c). If both conditions are true, a NaK will be sent by the receiver.

If a frame is damaged, it will be detected by checking the checksum and thus resulting in a checksum error event. When a checksum error event occurs, NaK will be sent by the receiver (in Figure 5b and 5c).

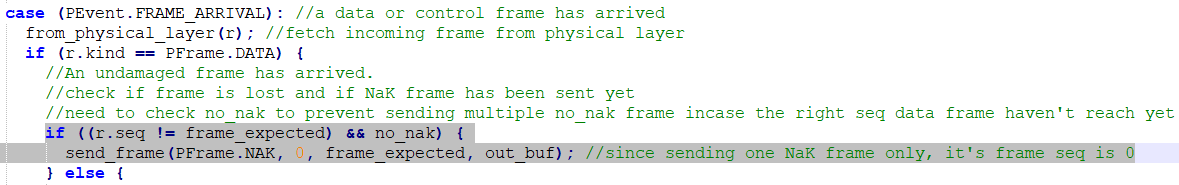


Figure 5a

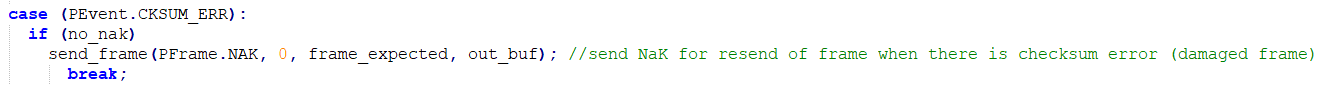


Figure 5b

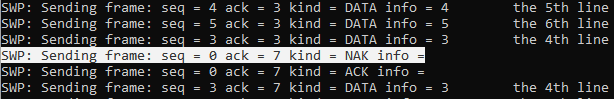


Figure 5c

**Separate acknowledgment when the reverse traffic is light or none**

Acknowledgment are usually piggybacked on reverse outgoing frames to reduce wasted bandwidth. However, there are scenarios when the reverse traffic is light or none. If the wait is too long, the sender’s data frame timer may time out due to not receiving acknowledgement from the receiver. This results in retransmission of data frame to the receiver.

Therefore, the solution is to implement an acknowledgement timer. The acknowledgement timer will be begin when an undamaged data packet is sent to the network-layer (in Figure 6a). The acknowledgement timer will be terminated if the acknowledgement piggybacks on a reverse outgoing frame (in Figure 6b). If not, once time is up, the acknowledgement timer will trigger an ACK\_TIMEOUT event, thus a separate acknowledgement is sent (in Figure 6c and 6e). The acknowledgement timer has to be shorter than the data frame timer (in Figure 6d).

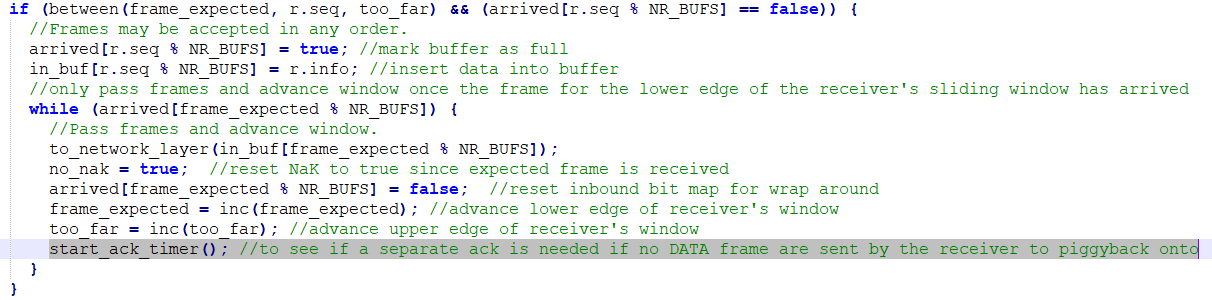


Figure 6a

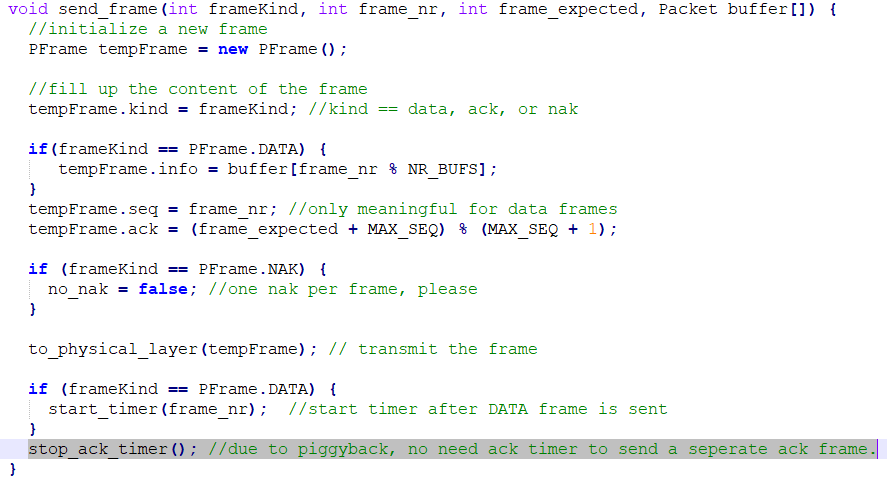


Figure 6b

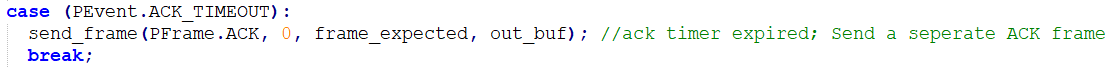


Figure 6c

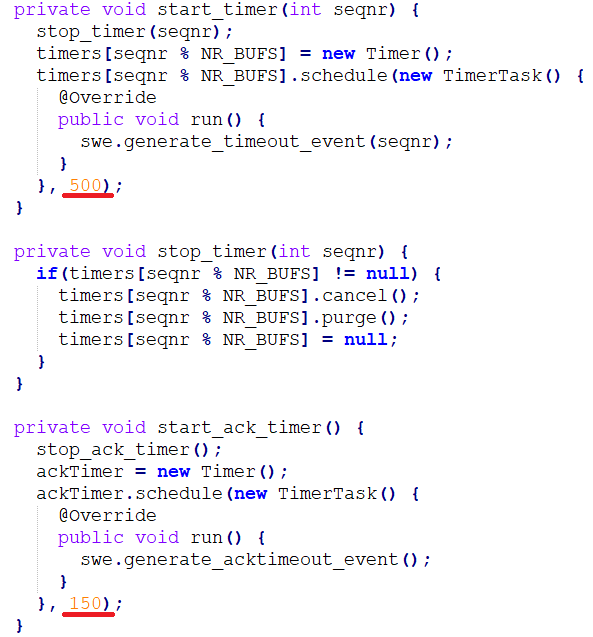


Figure 6d

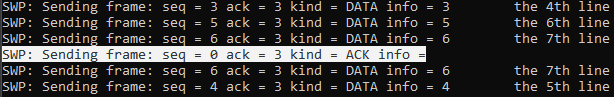


Figure 6e

**Java Source File (SWP.java)**

