

CZ3006: Net-Centric Computing

Assignment 1 Report

Yi Zhiyue (U1620162A)

Luo Bingyi(U1620274F)

Remark:

This assignment is completed jointly. Code implementation and report are done through group discussion

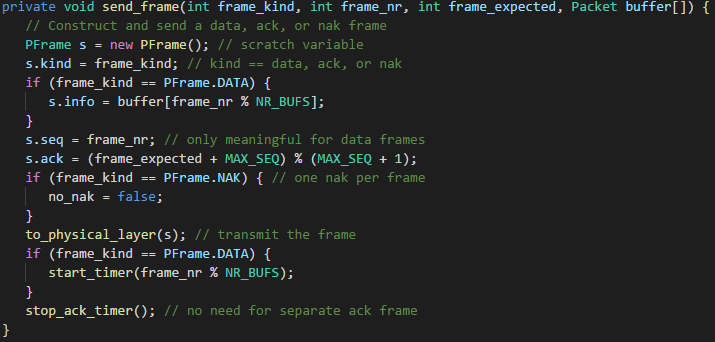
# Summary

|  |  |
| --- | --- |
| Sliding window protocol features | Status |
| Full-duplex data communication | Finished |
| In-order delivery of packets to the network-layer | Finished |
| Selective repeat retransmission strategy | Finished |
| Synchronization with the network-layer by granting credits | Finished |
| Negative acknowledgement | Finished |
| Separate acknowledgement when the reverse traffic is light or none | Finished |

# Features Implementation

## Full-duplex data communication

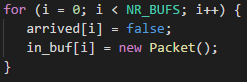
Data frames can be transmitted in both directions with a forward transmission (for data) and a reverse transmission (for acknowledgement).



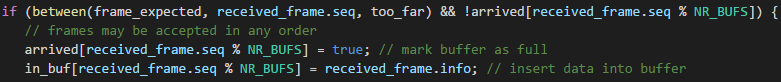
In the code implementation, we use the send\_frame method to send either a data, acknowledgement or a negative acknowledgement (Nak) frame to realize the full-duplex data communication.

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

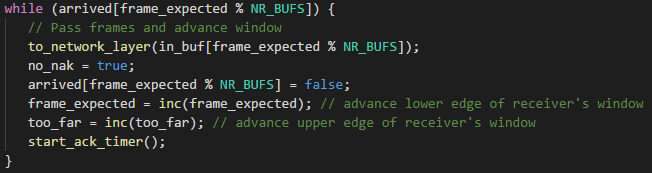
Protocol 6 accepts frames out of order but passes packets to the network layer in order. The receiver has a buffer *in\_buf* reserved for each sequence number within its fixed window. The buffer is declared as Packet in\_buf[] = new Packet[NR\_BUFS] . Associated with each buffer is a bit declared as boolean arrived[] = new boolean[NR\_BUFS] telling whether the buffer is full or empty. Initially, the bit *arrived* is set to false and each buffer is empty.



Whenever a frame arrives, its sequence number is checked by the function *between* to see if it falls within the window. If so and if it has not already been received, it is accepted and stored, thus the buffer is marked as true and the data from the frame is inserted into the buffer.



The next packet expected by the network layer must be kept within the data link layer and not passed to the network layer until all the lower-numbered frames have already been delivered to the network layer in the correct order.



## Selective repeat retransmission strategy

Selective repeat is combined with having the receiver send a negative acknowledgement (Nak) when it detects an error, for example, when it receives a checksum error or a frame out of sequence. Naks stimulate retransmission before the corresponding sender timer expires and thus improve performance. The Nak frame is stored in the *kind* of the frame. When the *kind* field is the Nak frame, the corresponding frame will be retransmitted.





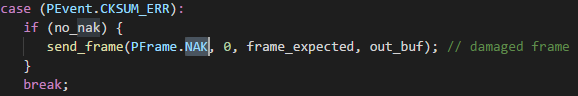
## Synchronization with the network-layer by granting credits

The receiver should keep track of whether a Nak has already been sent for a given frame. The variable *no\_nak* is true if no Nak has been sent yet for *frame\_expected*. If the Nak gets mangled or lost, the sender will eventually time out and retransmit the missing frame. If the wrong frame arrives after a Nak has been sent and lost, *no\_nak* will be true and the auxiliary time *ackTimer* will be started. When it expires, an ack will be sent to resynchronize the sender to the receiver’s current status.

## Negative acknowledgement

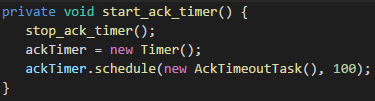
The Nak will be sent through the frame *kind* field when it detects an error such as when it receives a checksum error or a frame out sequence.





## Separate acknowledgement when the reverse traffic is light or none

When an in-sequence data frame arrives, an auxiliary time *ackTimer* is started by *start\_ack\_timer.*



If no reverse traffic has presented itself before this timer expires, a separate acknowledgement frame is sent.



# Output Result of Different Quality Levels

## Receive\_files in each quality level

For the results in each quality levels, only in quality level 0, the send\_file1 and send\_file2 are transmitted without any error.

|  |  |  |
| --- | --- | --- |
|  | Receive\_file\_1 | Receive\_file\_2 |
| Quality Level 0 |  |  |
| Quality Level 1 |  |  |
| Quality Level 2 |  |  |
| Quality Level 3 |  |  |