Object Oriented Software Engineering (COMP2003) Report

Updated: 24th October, 2016

Introduction

This program is intended to simulate the sending and receiving of messages between a seven node network. The protocol used is sliding window. It has been implemented using the CNET program written by Chris McDonald from the University of Western Australia. The data link layer and application layer were the only layers deemed necessary by the assignment specification.

1. Writing Functions

The functions necessary for the process of receiving a message from the application layer and writing it to the physical layer. This section contains the functions that represent the application layer, network layer, data link layer and physical layer.

2. Reading Functions

The functions necessary for the process of reading a message from the physical layer and passing it to the application layer. This section determines if the frame is data or an ACK, whether it has reached its final destination and if the frame is corrupt or not.

3. Forwarding Function

If a frame received has not reached its final destination, the forward frame function is used to determine the link the frame must be sent on and to send the frame onwards.

4. Acknowledgement Functions

This section contains the functions for sending and receiving acknowledgments. If the frame received is an acknowledgement it is handled by the acknowledgement received function. The other function sends an acknowledgement for a frame received back to the sender.

5. Timeout Functions

Contained here are function for initialising timers and determining which timer to start. While also handling the actions that must be taken when a timer expires. Every node may contain multiple links, as such a different timer can be initiated for each of these links.

6. Initialisation Function

This section contains a single reboot node function. It is for initialising each of the nodes before sending and receiving of messages may begin.

7. Miscellaneous Functions

Stored here are functions used in the operation of the sliding window protocol. They however are not specific to the sending and receiving process.

8. assignment.h

This file defines the data structures and variables used in the operation of the sliding window protocol. Along with the prototpyes for the functions used.

9. ASSIGNMENT

This file defines the topology of the network that data is being sent on.

The log files contain the output from running the sliding window protocol for each node in the network.

10. Algorithm Choice

Sliding window was chosen as it is more efficient than stop-and-wait. Stop-and-wait only allows the transmission of one message at a time. The next frame can only be sent once the ACK for the last one sent is received. With sliding window multiple frames can be sent before receiving an ACK. With the go-back-n approach chosen if the sender detects an error with frame n, it will reject all frames after n and the sender will retransmit all frames after frame n.

11. Receiving

When a frame is read from the physical layer it can be a data frame or an acknowledgment frame.

12. Data

If it is a data frame a checksum is calculated and compared with the checksum stored in the frame.If they match the frame is passed to the network layer and an ACK is sent to the sender. If this is the intended destination for the message it will be passed to the application layer.

13. Acknowledgement

If it is an ACK all frames with a sequence number lower than that of the ACK are considered to be acknowledged. The timer for the now acknowledged frames are stopped. The frames are also removed from the window, to make room for more frames to be sent. Frames are then loaded from the buffer of frames and loaded into the window to be transmitted.

14. Re-Transmission

There are three possibilities for the re-transmission of a frame. The first is the frame's checksum calculated at the destination did not match the checksum stored in the frame prior to sending. The second option is the frame received at the node is not at it's final destination and must be rerouted and sent onwards to reach its intended destination. The third option is the sequence number of the frame received is not the same as the one expected, but the frame is the same as the checksum matches.

15. Corrupt Frame

Prior to writing the frame to the physical layer a checksum is calculated and stored in the frame. Once received by a node the checksum is recalculated and compared with the one stored in the frame. If the checksums do not match the frame will be ignored and an ACK will not be sent to the sender, causing a timer to expire. Once the timer expires it will trigger the sender to retransmit the frames after the last ACK received.

16. Rerouting Frame

If the frame has not reached its intended destination; an ACK will be sent to the sender to signify it has been received, the link the frame needs to travel on will be found and the frame will be transmitted from the datalink layer to the physical layer.

17. Unexpected Sequence Number

If the checksum matches and the frame is determined to be storing data the next step is to check the sequence number of the frame against the sequence number expected. If the sequence number is is one less than expected, this indicates the ACK for the previous frame was not received by the sender and the same frame was sent again. An ACK for the latest frame received is then sent back. If the sequence number is greater than expected it means one of the first frames of the many sent have not been received.The frames are dropped, which will result in the sender retransmitting the frames in the correct order once again, after the timer runs out.

End of Worksheet