CC Report

Luke Mercuri 17365547

May 2014

1 Usage

- Run with 'make'
- Press 's' key for statistics

2 Implementation

- Seperated into 6 main Functions
- app_down_net() Application down (indirectly) to Network
  - Reads a message in from Application
  - Disables application layer for node
  - Send to network
- net_down_dll() Network down to Datalink Layer
  - Use static routing table to decide a link to take
  - Update sequence
  - Put frame into queue (to be sent)
- dll_down_phys() Datalink Layer down to Physical
  - Decide on length of frame (depending on ACK or DATA)
  - If DATA then start timeout timer
  - Write frame to Physical
- phys_up_dll() Physical up to Datalink Layer
  - Read frame from Physical and send up to Datalink
- dll_up_net() Datalink layer up to Network
  - Check checksum of frame, ensure not corrupt

  - If ACK and the ACK node was expecting then stop timer, remove sent DATA from queue and send next item in queue
  - If ACK but incorrect sequence, ignore/discard
  - If DATA and correct sequence, send an ACK and pass packet up to Network
  - Else ignore duplicate DATA and re-send ACK

- net_up_app() Network either back down to Datalink or up to Application
  - Check whether packet has reached destination or just an intermediary
  - If destination then send up to Application
  - Else send back down to Datalink layer for next hop

3 Assumptions Made

- Links never go down
- CNET's checksum function will not encounter collisions (corrupt frames will not have same checksum as valid frames)