DVA218, LAB3b

Leslie Dahlberg (ldg14001), Jonathan Larsson (jln14010)

# Introduction

In this report we discuss how to implement a reliable transport protocol upon the existing UDP-protocol. This project is based upon the state machines in the report for LAB3a and written in C for Linux.

Differences between the sketches and our implementation

* The sliding window algorithm used an infinite set of sequence numbers in the sketches. In our program we set the buffer to a fixes size and let sequence numbers wrap around that length with a modulo function.
* In our implementation every package communicates the used window size and receiver can chose to accept it and send it back or return it’s own preference for a window size which the sender will be forced to accept.
* We use the internet checksum to verify all packets including ACKs and handshake packets
* An error generator randomly drops and corrupts packets in order to test the resiliency of our ARQ-mechanism. We use rand()%DROP\_RATE == 1 where drop rate is the inverse probability of a packet being dropped or corrupted to decide when an error should occur.
* To send received packets to the application layer the user of the protocol set a function pointer which will accept incoming data and process it.
* The receiving of packages and placing of packages onto the sending buffer is handled separately from the rest of the finite state machine.

# Debug output