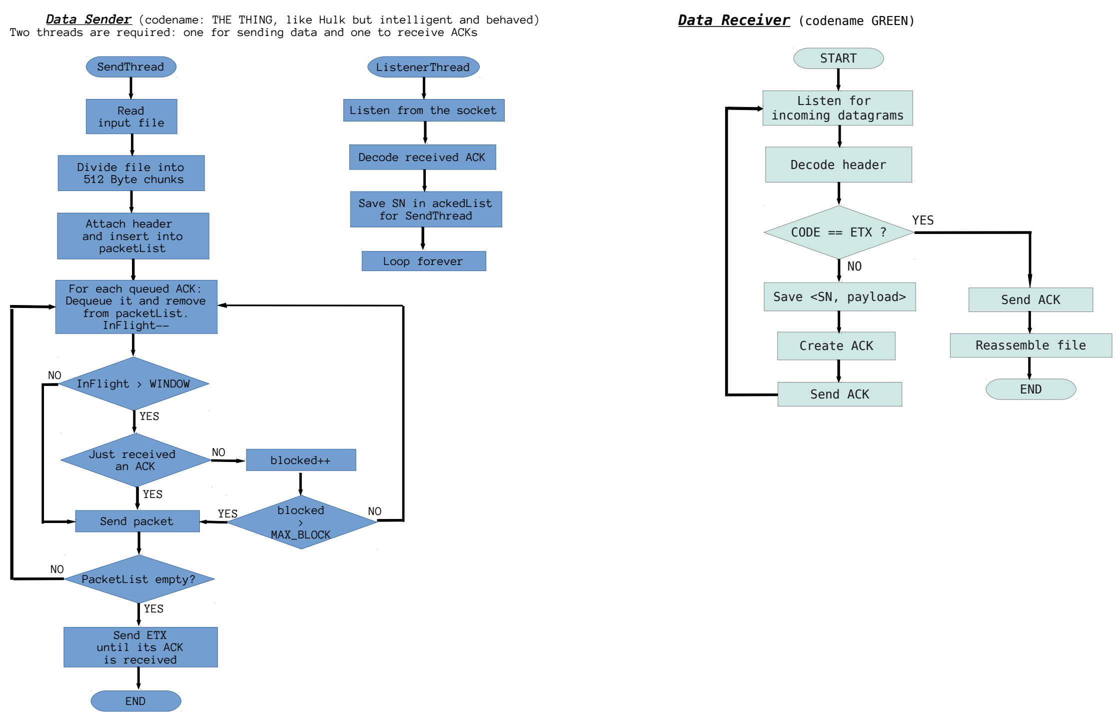
HIPSTER, the alternative file transfer protocol

1 Introduction

Is this really needed? A new protocol for file transfer - over UDP - with a bad channel in the middle. Such fun!

2 Protocol description

blabla
Some flowcharts Define DR, DS, TC Compare HIPSTER with stop and wait and with a simple delay between packets. Tell them we tried to use the delay before the poor man's sliding window.



(a) DS flowchart

(b) DR flowchart

Figure 1: HIPSTER protocol flowchart

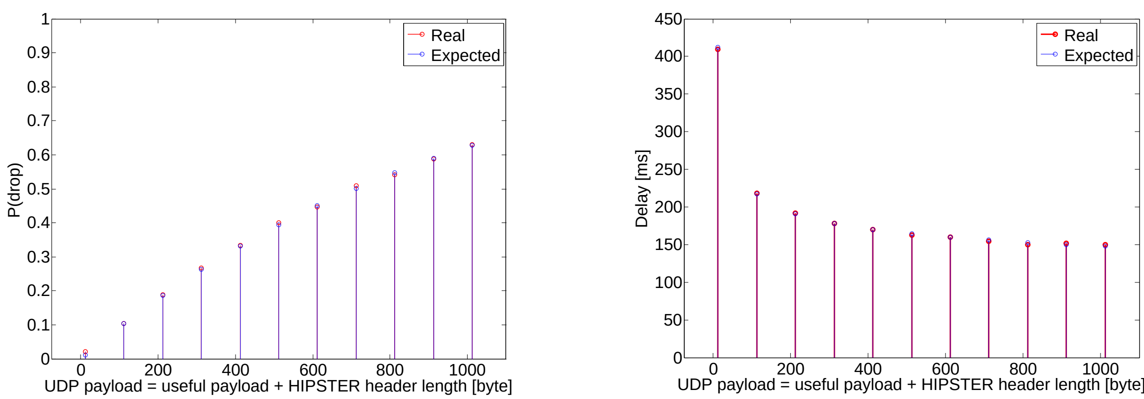
3 Transport Channel test results

The Transport Channel (TC) module simulates a bad channel among the DS and DR. It drops received UDP packets of length L with probability $P_{drop} = 1 - \exp(-L/1024)$ and forward the remaining ones with a delay distribuited according to an expontial random variable with mean $1024/\ln(L)$.
The TC was tested in loopback, by measuring the delay between the transmission of a packet by DS and its reception at DR and

```
- The channel needs to know the destination (address and port) for the packet
  in order to forward it

- The receiver needs to know the size of the payload. LENGTH is 10 bytes long so the maximum payload length is 1024 Bytes

- SEQUENCE NUMBER and CODE are used for signalling

0             15|16    26|27   31
|-------------------------------|
|     DESTINATION IP (ipv4)     |      // Destination:   0 to 31
|-------------------------------|      // Dst Port:      32 to 47
|   DST PORT     | LENGTH | CODE |
|-------------------------------|      // Length:        48 to 57
|        SEQUENCE NUMBER        |      // Code:          58 to 63
|-------------------------------|      // SN:            64 to 95
|                               |      // Data:          96 to (96+Length)
|              DATA             |
|                               |
|-------------------------------|

* DESTINATION IP:
        If the dst IP is 22.2.19.92, then in the header byte[0]=22, byte[1]=2, byte[2]=19, byte[3]=92
* PORT:
        The most significant byte is in byte[4], the least significant byte is in byte[5]

* CODE:

        0 -> Regular data packet
        1 -> ACK (sequence number is the same as the packet being ACKed)
        2 -> ETX (End of Transmission)
```

Figure 2: HIPSTER packet structure

the ratio between sent and received packets. The payload of UDP packet used in this test was in the range from 12 byte (HIPSTER header length, thus the size of an ACK) to 1012 byte (actual size of HIPSTER packets). Each measurement was taken 10 times. The results are in figure 3. The TC follows accurately the theoretical model, the few discrepancies are related to the Java random number generator and the finiteness of measurements. The distribution of the delays introduced by the TC in a transmission fits the required exponential RV according to the Kolmogorov-Smirnov test and the comparison of the cumulative distributive functions (CDF) of the two can be seen in figure 4.



(a) Dropping probability

(b) Mean delay

Figure 3: TC statistics

4 Performance analysis

5 Concusion

O RLY? YUP!

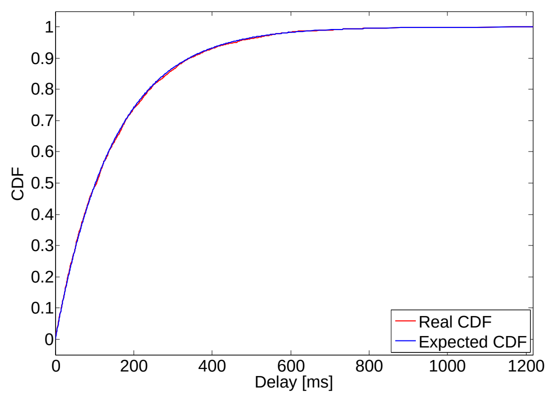


Figure 4: CDF of the delays introduced by TC, $L_{UDP} = 1012$ byte