**Reliable Udp File Transfer(ruft)**

**Sirshak Das(sirdas@indiana,edu)**

# Targaryen Server and Martell Client(ruft\_server.bin and ruft\_client.bin)

**I.Header Design**

##Ruft Header##  
+--------------------------------+  
| Flags(16 bits) |AdvWin(16 bits)|  
+--------------------------------+  
| Acknowledgement No(32 bits) |  
+--------------------------------+  
| Sequence no(32 bits) |  
+--------------------------------+  
| Payload Length(32 bits) |  
+--------------------------------+  
| Payload |  
| |  
. .  
. .  
. .  
| |  
+--------------------------------+

This is the application header for a ruft packet.

**Fields**

* **Flags(16bits)-** This has some necessary flags:
  + **is\_ack(first bit):** set if the packet is an acknowledgment
  + **is\_data\_pkt(second bit):** set if this is data packet unset if control packet
  + **is\_last\_pkt(third bit):** set if this is the last packet in communication
* **Advertised Win(16 bits):** Designed to be used for window negotiation between receiver and sender but left unused as receiver and sender negotiate it offline.
* **Acknowledgement No(32 bits):** Acknowledgement no stating the byte received till now.
* **Sequence No(32 bits):** Sequence no indicating the starting byte of the transfer byte stream.
* **Payload Length(32 bits):** Payload length indicating the length of the payload expected. Max is hardcoded as 1280 bits(MSS).
* **Payload(Variable length):** Depending on payload length max hardcoded as 1280 bits(MSS).

**Note:** All packets transferred on wire are on Network Order it is converted to host order in the respective programs i.e application layer.

**II. Basic Protocol Packets exchange**

**Targaryen(ruft\_server) Martell(ruft\_client)**

**file\_rqst**



**reqst\_recvd**  **file\_size received**

**file\_seg transfer**  **ack sent back**



**multiple segments** 

**based on wnd\_size** 



**.**

**.**

**.**

 **cumulative ack sent**

Things supported

1. Flow control using sliding window negotiated offline
2. Duplicate Acknowledgements
3. Congestion Control phases :
   1. Slow Start
   2. Congestion Avoidance
4. Adaptive retransmission using
   1. Jacobson/Karels algorithm
5. On client side
   1. No drop behaviour
   2. Variable packet loss
   3. High latency communication

**III. Implementation Details**

1. **Flow Control**

This has been done using sliding window. The server decides the window s size seeing the state of the server. Based on that it sends all the segments present in the window and waits for the ack. If packets are lost then it retransmits accordingly after timeout. Once all the packets have been acked using cumulative ack from the server the server resets the window size based on the state of the server and perform the sending action again

1. **Duplicate Acknowledgements**

In case of out of order delivery Duplicate ACK is sent from the client side. Upon receiving this the server only transfers one data segment corresponding to the ack. After which the client replies back with cumulative ack.

1. **Congestion Control**
   1. **Slow Start**

The server starts on slow start and then moves to congestion avoidance based on the distinct no of acks it got during the window transfer. If it crosses the threshold then the server moves to congestion avoidance.

* 1. **Congestion Avoidance**

The server after moving to congestion avoidance increases the window size linearly compared to exponentially in slow start.

1. **Adaptive retransmission**

The server and client uses adaptive retransmission by dynamically calculating the timeout based on Jacobson/Karels algorithm.

/\*The Jacobson/Karels algo according to the book\*/

est\_rtt = (0.875)\*est\_rtt + (0.125)\*(rtt);

dev\_rtt = (0.75)\*dev\_rtt + (0.25)\*(abs(rtt-est\_rtt));

timeout = est\_rtt + 4\*dev\_rtt;

1. **Client side** 
   1. **No Drop Mode**

In this mode the client does not perform any drops on its own.

* 1. **Variable Packet Loss**

This is done by deciding a random packet to be dropped in the receiver window in client side using rand() function.

rd\_index = recv\_ctr + rand()%(rwnd\_seg - recv\_ctr);

* 1. **High Latency Communication**

This is done by introducing sleep using usleep for microseconds granularity. The default sleep introduced is 1000 microseconds for each packet received.

usleep(DELAY);

**IV. Statistics Collected**

**SMALL FILE(File Size: 6 bytes)**

*No Drop Mode*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_small.txt 0 5120

File Recieved

Time taken in micro secs: 3919

**SERVER**

File Sent

Total Packets Stats:

TX: 5 RX: 2

Slow Start Packets Stats:

TX: 5 RX: 1 TX%: 100 RX%: 50

Congestion Avoidance Packets Stats:

TX: 0 RX: 0 TX%: 0 RX%: 0

*Variable Packet Loss*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_small.txt 1 5120

File Recieved

Time taken in micro secs: 3982

**SERVER**

Total Packets Stats:

TX: 6 RX: 2

Slow Start Packets Stats:

TX: 6 RX: 1 TX%: 100 RX%: 50

Congestion Avoidance Packets Stats:

TX: 0 RX: 0 TX%: 0 RX%: 0

*High Latency Communication*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_small.txt 2 5120

File Recieved

Time taken in micro secs: 5384

[sirdas@silo client]$

**SERVER**

File Sent

Total Packets Stats:

TX: 8 RX: 2

Slow Start Packets Stats:

TX: 8 RX: 1 TX%: 100 RX%: 50

Congestion Avoidance Packets Stats:

TX: 0 RX: 0 TX%: 0 RX%: 0

**MED FILE(20110 bytes Approx 20.11 KB)**

*No Drop Mode*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_med.txt 0 5120

File Recieved

Time taken in micro secs: 12590

**SERVER**

File Sent

Total Packets Stats:

TX: 23 RX: 6

Slow Start Packets Stats:

TX: 9 RX: 3 TX%: 39 RX%: 50

Congestion Avoidance Packets Stats:

TX: 14 RX: 2 TX%: 60 RX%: 33

*Variable Packet Loss*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_med.txt 1 5120

File Recieved

Time taken in micro secs: 21398

**SERVER**

File Sent

Total Packets Stats:

TX: 50 RX: 25

Slow Start Packets Stats:

TX: 36 RX: 16 TX%: 72 RX%: 64

Congestion Avoidance Packets Stats:

TX: 14 RX: 8 TX%: 28 RX%: 32

*High Latency Communication*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_med.txt 2 5120

File Recieved

Time taken in micro secs: 35066

**SERVER**

File Sent

Total Packets Stats:

TX: 159 RX: 14

Slow Start Packets Stats:

TX: 145 RX: 13 TX%: 91 RX%: 92

Congestion Avoidance Packets Stats:

TX: 14 RX: 0 TX%: 8 RX%: 0

**Medium Large TXT(File Size: 1009368 bytes Approx: 1MB )**

*No Drop Mode*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_med\_large.txt 0 5120

File Recieved

Time taken in secs: 1

**SERVER**

File Sent

Total Packets Stats:

TX: 1173 RX: 106

Slow Start Packets Stats:

TX: 385 RX: 4 TX%: 32 RX%: 3

Congestion Avoidance Packets Stats:

TX: 788 RX: 101 TX%: 67 RX%: 95

*Variable Packet Loss*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_med\_large.txt 1 5120

File Recieved

Time taken in secs: 2

**SERVER**

File Sent

Total Packets Stats:

TX: 3052 RX: 1009

Slow Start Packets Stats:

TX: 2259 RX: 767 TX%: 74 RX%: 76

Congestion Avoidance Packets Stats:

TX: 793 RX: 241 TX%: 25 RX%: 23

*High Latency Communication*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_med\_large.txt 2 5120

File Recieved

Time taken in secs: 3

**SERVER**

File Sent

Total Packets Stats:

TX: 1586 RX: 275

Slow Start Packets Stats:

TX: 793 RX: 182 TX%: 50 RX%: 66

Congestion Avoidance Packets Stats:

TX: 793 RX: 92 TX%: 50 RX%: 33

**LARGE PACKET TXT(file size: 5348511 bytes Approx: 5.3MB)**

*No Drop Mode*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_large.txt 0 5120

File Recieved

Time taken in secs: 1

**SERVER**

File Sent

Total Packets Stats:

TX: 4185 RX: 107

Slow Start Packets Stats:

TX: 8 RX: 3 TX%: 0 RX%: 2

Congestion Avoidance Packets Stats:

TX: 4177 RX: 103 TX%: 99 RX%: 96

*Variable Packet Loss*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_large.txt 1 5120

File Recieved

Time taken in secs: 23

**SERVER**

Total Packets Stats:

TX: 46138 RX: 2543

Slow Start Packets Stats:

TX: 41953 RX: 2015 TX%: 90 RX%: 79

Congestion Avoidance Packets Stats:

TX: 4185 RX: 527 TX%: 9 RX%: 20

*High Latency Communication*

**CLIENT**

[sirdas@silo client]$ ./ruft\_client.bin localhost 20010 file\_large.txt 2 5120

File Recieved

Time taken in secs: 165

**SERVER**

[sirdas@silo reliable\_udp\_file\_transfer]$ ./ruft\_server.bin 20010 5120

File Sent

Total Packets Stats:

TX: 68535 RX: 4973

Slow Start Packets Stats:

TX: 64355 RX: 4819 TX%: 93 RX%: 96

Congestion Avoidance Packets Stats:

TX: 4180 RX: 153 TX%: 6 RX%: 3

====================================END=======================================