**RELIABLE DATA TRANSFER PROTOCOL**

SS4005 Sai Tarun Sathyan

**Objective:** The goal is to produce and demonstrate a protocol that can guarantee the delivery of the packets. The packets produced must also move easily through the regular Internet. The resulting protocol must, in theory, perform better than TCP with smaller packets and hopefully fewer packets sent.

**Approach:** Since we have to improve upon TCP and build a faster yet reliable protocol. I chose to use UDP and make it more reliable and provide congestion control. To do so, I made sure to address the packet loss issues in UDP and also standardize every transfer with a packet recovery process. We do not need to add and explicit checksum since, UDPs checksum is good enough.

|  |  |
| --- | --- |
| **TCP** | **UDP** |
| Connection oriented | Connectionless |
| Ordered | Unordered |
| Has traffic control | No traffic control |
| High overhead | Low overhead |

How to make UDP more reliable:

* To make UDP more reliable I have added a SYN packet at the start of each transmission. After which the receiver will send back an Ack for this SYN packet indicating we are good to go. This acts like a pseudo handshake.
* After this pseudo handshake we start transmitting the data.
* The data is broken down into multiple smaller packets and each packet contains a header and the payload.
* For each packet sent, the sender waits for an ack before sending the next packet. In-case a packet was lost during transmission, the sender retransmits this packet. Every packet is light-weight due to the smaller header/payload when compared to TCP.

The different types of packets used are :-

1. SYN packet

|  |  |
| --- | --- |
| Flag | File Name |
| Source IP | |

Flag:

* 1: Text File
* 2: Image File
* 3: Command
* 4: Terminating connection

1. Header

|  |  |  |  |
| --- | --- | --- | --- |
| Source | SEQ | Last | Destination |

SEQ: Sequence number of that particular packet

Last : Indicates if this packet is the final one or not

1. Data Packet

|  |
| --- |
| Header |
| Data |

1. ACK

|  |
| --- |
| FLAG |

Flag: Indicates that the packet was received successfully

Packet Transfer Looks Similar to this:

