Summary

We will once again have two applications (UDP_Client.py and UDP_Server.py) but this time we will be using the UDP protocol, **not TCP** to communicate across the network. This means we have to create the reliable data transfer protocol (rdt) we saw in our textbook, specifically, we will create rdt2.2 for this assignment.

We will be using the following format for our Pseudo UDP packet:

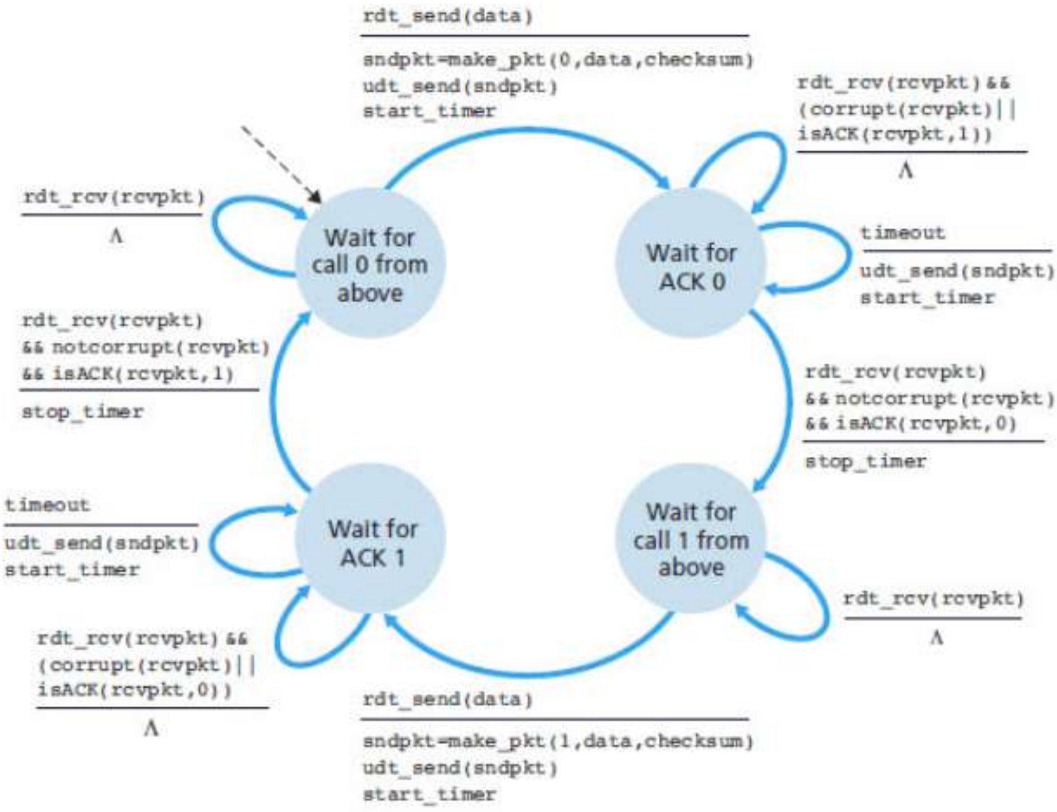
ACK	SEQ	DATA	CHKSUM

ACK – Indicates if packet is ACK or not. Valid values (1 or 0)
SEQ – Sequence number of packet. Valid values (1 or 0)
DATA – Application Data (8 bytes)
CHKSUM – MD5 Checksum of packet (32 Bytes)

UDP_Client –

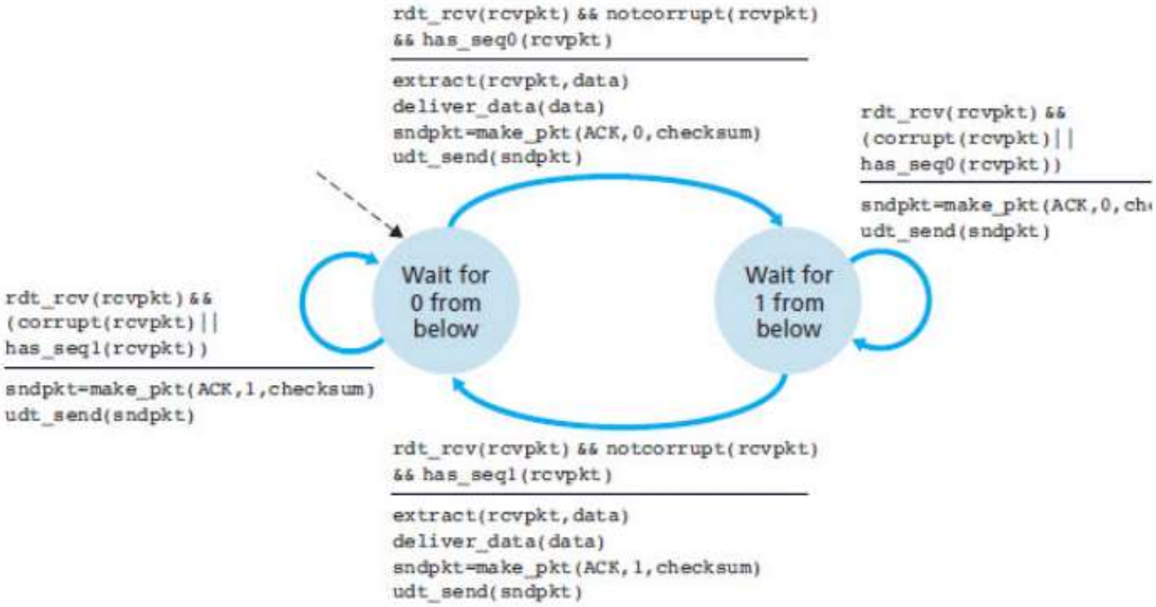
This app must connect to the UDP_Server app via UDP (you can use the local loopback address of 127.0.0.1 along with any port number you wish) then send three separate packets containing the following information:

NCC-1701
NCC-1664
NCC-1017



UDP_Server –

This app will establish a UDP socket and listen for incoming connections from the client. When a connection is made the server will consume the data and follow the rdt2.2 process as shown in chapter 3 of the course textbook.



Configuring Loopback Adapter to Introduce Delays and Losses -

To introduce packet losses and delays we will use this command:

```
sudo tc qdisc add dev lo root netem delay 10ms reorder 50% 50% loss 40%
```

This command will delay packets for 10ms (50% of the time) and lose 40% of packets. This queueing discipline runs on your loopback adapter so you must use the loopback address in both your applications.

To revert the loopback adapter to normal behaviour run the following command:

```
sudo tc qdisc del root dev lo
```

After you have configured the loopback adapter for delays and losses you will run your applications. Keep in mind that from the screenshots we should be able to see the lost or delayed packets based on what your applications are displaying on the screen.

The output from the UDP_Server and UDP_Client should display a line of text for each of the following actions:

- Received Packet (with all packet values shown)
- Packet Checksum Compare Result (ie. Corrupt or not corrupt)
- Sent Packet (with all packet values shown)
- Timer Expired