COMP3331 Assignment Report

I used Python 3 to complete this assignment and the project includes two files: sender.py and receiver.py.

Sender.py

It can receive configuration information through command line parameters, such as sender port, receiver port, file to be sent, maximum window size, retransmission timeout, packet loss probability, and out of order probability. When initializing the Sender class, a UDP socket is created and two threads are started to listen to the receiver's response and handle timeouts, respectively. The main functions include sending SYN, DATA, and FIN packets, processing received ACK packets, and handling timeout retransmissions.

The function **send_pack** can process the data from the source file, send different segments to receiver and record log information. **Timeout and receiver function** can handle timeout event and receive the ack segment from receiver.

For data structure, I used a dictionary to store segment type and a dataclass SendPack to store every packet.

```python
# headType
header_dict = {
    "DATA": 0,
    "ACK": 1,
    "SYN": 2,
    "FIN": 3,
}

# dataclass
@dataclass
class SendPack:
    seq_no = 0
    sent_packets = []

    def update_seqNo(self, content_len: int):
        self.seq_no = (self.seq_no + content_len) % MAXSEQNO
```

Receiver.py

It can receive data packets from the sender through the UDP protocol and processes them according to the protocol's specified operations.

Firstly, it sets the port number of the receiver and sender, create a UDP socket and bind it to the address of the receiver, initialize various variables and data structures.
In the main loop, the receiver continuously receives packets from the sender and

performs different processing based on the packet type. Then, it processed different types of segment packet. When a SYN packet is received, it indicates that the connection is being established. The receiver will transition the state to ESTABLISHED and record the time when the connection was established. When receiving a DATA packet, the receiver will cache the packet and send an ACK confirmation message to the sender, while recording the time, serial number, and length of the packet's arrival.

When a FIN packet is received, it indicates that the sender has completed the data transmission, and the receiver will transition the status to TIMP_WAIT. After waiting for a period of time, it will then transition to the Closed state. Also, the receiver can record the log information.