Programming Assignment

Question 1 Report

Computer Networks Final Project – COEN 233

Professor Keyvan Moataghed

Omid Almasi

[redacted]

Computer Science and Engineering

University of Santa Clara

August 2023

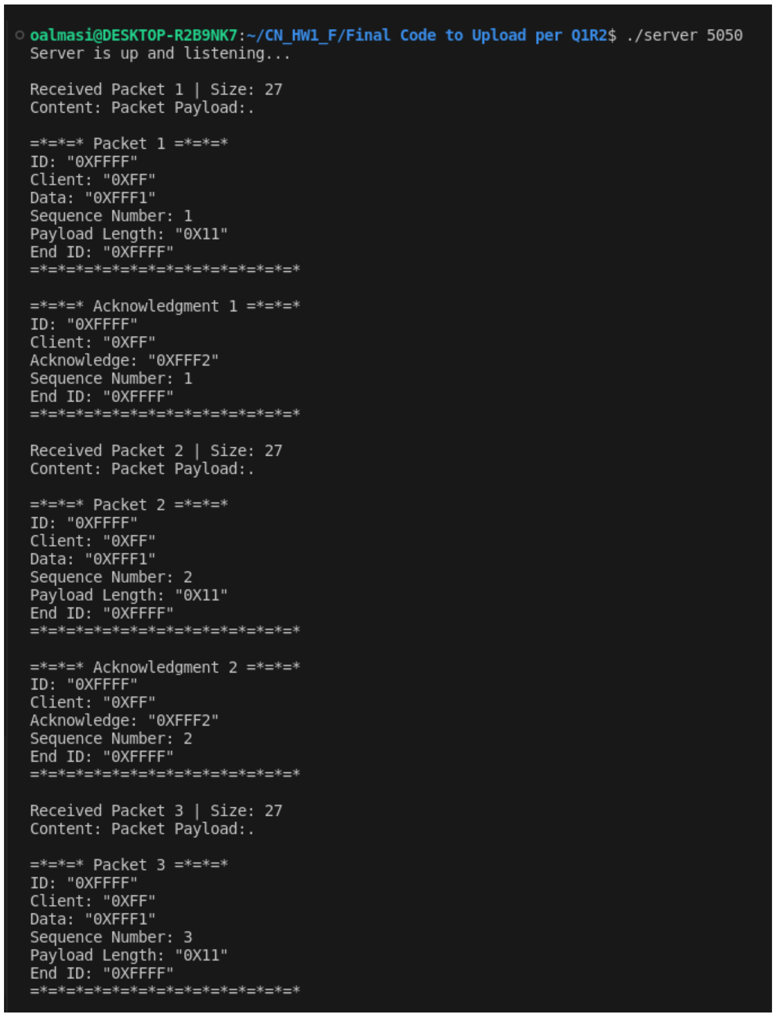
Table of Contents

List of Figures

1. Question No. 1 Client using customized protocol on top of UDP protocol

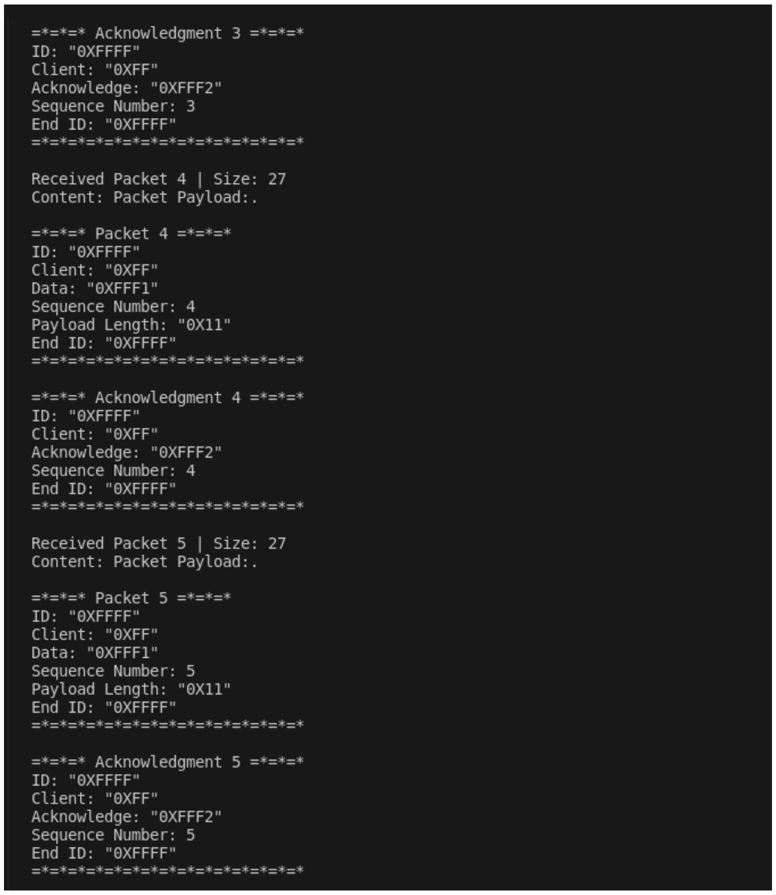
1.1 Procedure

The client sends five packets (Packet 1, 2, 3, 4, 5) to the server. The server acknowledges with ACK receive of each correct packet from client by sending five ACKs, one ACK for each 5 received packets.



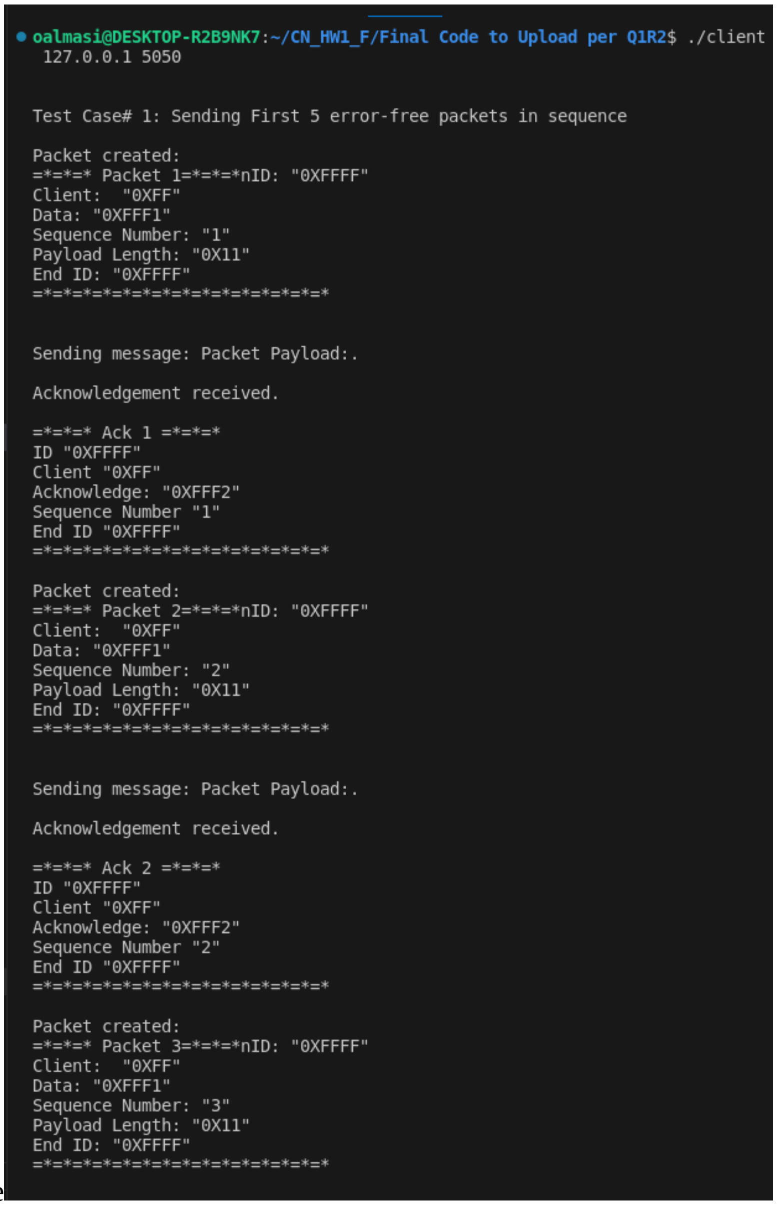
```
○ oalmasi@DESKTOP-R2B9NK7:~/CN_HW1_F/Final Code to Upload per Q1R2$ ./server 5050
  Server is up and listening...

  Received Packet 1 | Size: 27
  Content: Packet Payload:.

  =*=*=* Packet 1 =*=*=*
  ID: "0XFFFF"
  Client: "0XFF"
  Data: "0XFFF1"
  Sequence Number: 1
  Payload Length: "0X11"
  End ID: "0XFFFF"
  =*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

  =*=*=* Acknowledgment 1 =*=*=*
  ID: "0XFFFF"
  Client: "0XFF"
  Acknowledge: "0XFFF2"
  Sequence Number: 1
  End ID: "0XFFFF"
  =*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

  Received Packet 2 | Size: 27
  Content: Packet Payload:.

  =*=*=* Packet 2 =*=*=*
  ID: "0XFFFF"
  Client: "0XFF"
  Data: "0XFFF1"
  Sequence Number: 2
  Payload Length: "0X11"
  End ID: "0XFFFF"
  =*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

  =*=*=* Acknowledgment 2 =*=*=*
  ID: "0XFFFF"
  Client: "0XFF"
  Acknowledge: "0XFFF2"
  Sequence Number: 2
  End ID: "0XFFFF"
  =*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

  Received Packet 3 | Size: 27
  Content: Packet Payload:.

  =*=*=* Packet 3 =*=*=*
  ID: "0XFFFF"
  Client: "0XFF"
  Data: "0XFFF1"
  Sequence Number: 3
  Payload Length: "0X11"
  End ID: "0XFFFF"
  =*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

Figure 1 – Server Output Terminal for first 5 packets (no error) – packet 1 thru 3.



```
=*=*=* Acknowledgment 3 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Acknowledge: "0XFFF2"
Sequence Number: 3
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Received Packet 4 | Size: 27
Content: Packet Payload:.

=*=*=* Packet 4 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 4
Payload Length: "0X11"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

=*=*=* Acknowledgment 4 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Acknowledge: "0XFFF2"
Sequence Number: 4
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Received Packet 5 | Size: 27
Content: Packet Payload:.

=*=*=* Packet 5 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 5
Payload Length: "0X11"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

=*=*=* Acknowledgment 5 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Acknowledge: "0XFFF2"
Sequence Number: 5
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

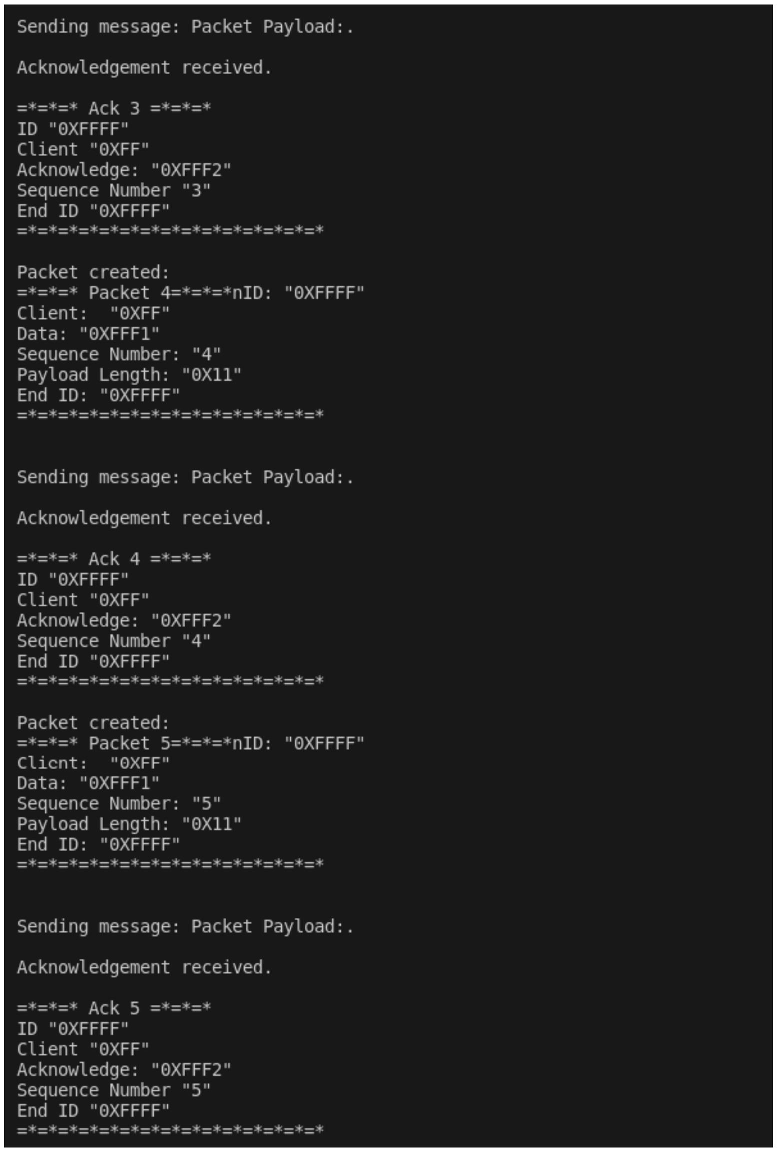
Figure 2 – Continuation of Server Output Terminal for first 5 packets (no error) – packet 3 thru 5.



```
oalmasi@DESKTOP-R2B9NK7:~/CN_HW1_F/Final Code to Upload per Q1R2$ ./client
 127.0.0.1 5050


Test Case# 1: Sending First 5 error-free packets in sequence

Packet created:
=*=*=* Packet 1=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "1"
Payload Length: "0X11"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*


Sending message: Packet Payload:.

Acknowledgement received.

=*=*=* Ack 1 =*=*=*
ID "0XFFFF"
Client "0XFF"
Acknowledge: "0XFFF2"
Sequence Number "1"
End ID "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Packet created:
=*=*=* Packet 2=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "2"
Payload Length: "0X11"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*


Sending message: Packet Payload:.

Acknowledgement received.

=*=*=* Ack 2 =*=*=*
ID "0XFFFF"
Client "0XFF"
Acknowledge: "0XFFF2"
Sequence Number "2"
End ID "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Packet created:
=*=*=* Packet 3=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "3"
Payload Length: "0X11"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

e

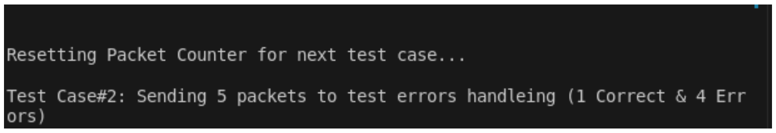
Figure 3 – Client Output Terminal for first 5 packets (no error) – packet 1 thru 3.



```
Sending message: Packet Payload:.

Acknowledgement received.

=*=*=* Ack 3 =*=*=*
ID "0XFFFF"
Client "0XFF"
Acknowledge: "0XFFF2"
Sequence Number "3"
End ID "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Packet created:
=*=*=* Packet 4=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "4"
Payload Length: "0X11"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*


Sending message: Packet Payload:.

Acknowledgement received.

=*=*=* Ack 4 =*=*=*
ID "0XFFFF"
Client "0XFF"
Acknowledge: "0XFFF2"
Sequence Number "4"
End ID "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Packet created:
=*=*=* Packet 5=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "5"
Payload Length: "0X11"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*


Sending message: Packet Payload:.

Acknowledgement received.

=*=*=* Ack 5 =*=*=*
ID "0XFFFF"
Client "0XFF"
Acknowledge: "0XFFF2"
Sequence Number "5"
End ID "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

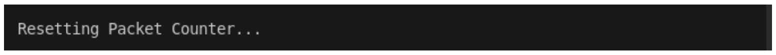
Figure 4 – Continuation of Client Output Terminal for first 5 packets (no error) – packet 3 thru 5.

Now, we need to reset the packet counter to simulate the error handing as described in section 1.2.



```
Resetting Packet Counter for next test case...

Test Case#2: Sending 5 packets to test errors handleing (1 Correct & 4 Err
ors)
```

a) Client



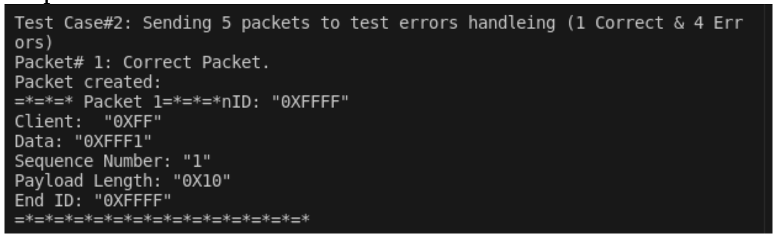
```
Resetting Packet Counter...
```

b) Server

Figure 5 –Output Terminals showing counter has been reset. a) Client, b) Server

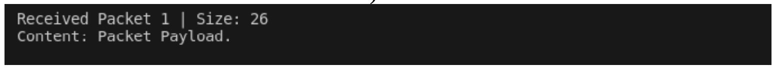
1.2 Error handling

The client then sends another five packets (Packet 1, 2, 3, 4, 5) to the server, emulating one correct packet and four packets with errors.



```
Test Case#2: Sending 5 packets to test errors handleing (1 Correct & 4 Err
ors)
Packet# 1: Correct Packet.
Packet created:
=*=*=* Packet 1=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "1"
Payload Length: "0X10"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

a) Client



```
Received Packet 1 | Size: 26
Content: Packet Payload.
```

b) Server

Figure 6 –Output Terminal for the correct packet (first) of second 5 packets for error handling, a) Client, b) server.

The server acknowledges with ACK receive of correct packet from client, and with corresponding Reject sub codes for packets with errors.

The client will start an ack_timer at the time the packet is sent to the server, if the ACK (Acknowledge) for each packet has not been received during ack_timer period by client before expiration of timer then client should retransmit the packet that was sent before.
The timer can be set at 3 seconds (recommended) and a retry counter should be used for resending the packet. If the ACK for the packet does not arrive before the timeout, the client will retransmit the packet and restart the ack_timer, and the ack_timer should be reset for a total of 3 times.
If no ACK was received from the server after resending the same packet 3 times, the client should generate the following message and display on the screen: "Server does not respond".

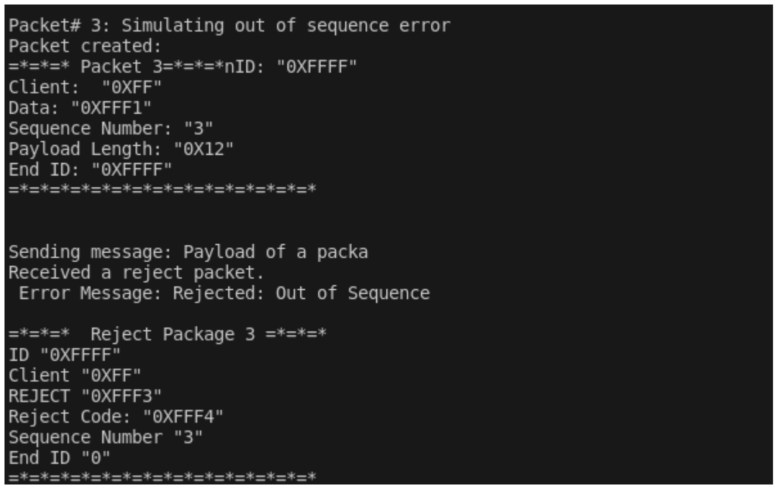
ACK error handling:

If the ACK timer expires and the ACK from server has not been received by client, an error message should be displayed on the screen by client prior to resending the packet.

Reject error handling with sub code:

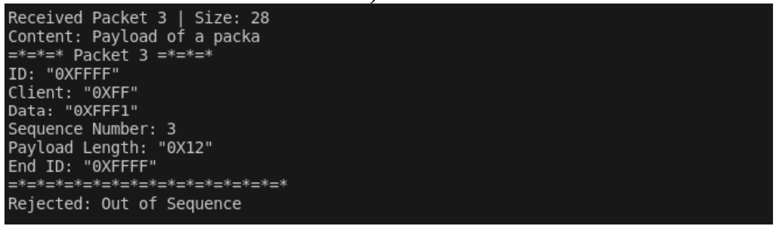
- **Case-1:** An error message should be displayed on the screen when the received packet at server is not in sequence with expected packet from client; an error message should be generated by server and sent to client.

For example, if server receives packets 0, 1 and then 3, packet 3 is out of sequence because the server is expecting the packet 2 after receiving the packet 1. The server will not increment the expected sequence number until packet 2 has been received.



```
Packet# 3: Simulating out of sequence error
Packet created:
=*=*=* Packet 3=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "3"
Payload Length: "0X12"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*


Sending message: Payload of a packa
Received a reject packet.
 Error Message: Rejected: Out of Sequence

=*=*=*  Reject Package 3 =*=*=*
ID "0XFFFF"
Client "0XFF"
REJECT "0XFFF3"
Reject Code: "0XFFF4"
Sequence Number "3"
End ID "0"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

a) Client



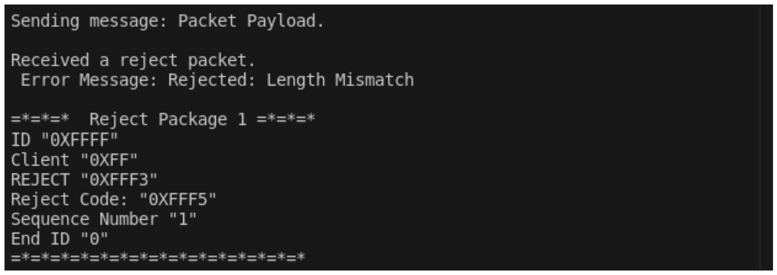
```
Received Packet 3 | Size: 28
Content: Payload of a packa
=*=*=* Packet 3 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 3
Payload Length: "0X12"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
Rejected: Out of Sequence
```

b) Server

Figure 7 –Output Terminal for the Case-1, a) Client, b) server.

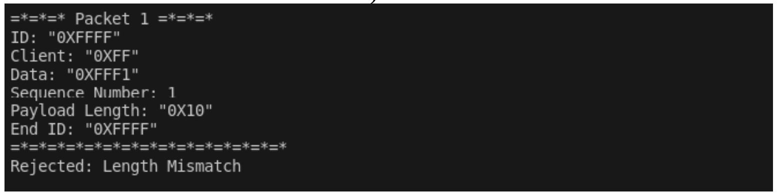
- **Case-2:** The server receives a packet which its length field does not match the length of data in the payload's field, an error message should be generated by server and send to the client.

For example, if the length field of a received packet indicates the data payload is 125 bytes, but the actual payload is only 12 bytes, this packet has a length mismatch error.



```
Sending message: Packet Payload.

Received a reject packet.
 Error Message: Rejected: Length Mismatch

=*=*=*  Reject Package 1 =*=*=*
ID "0XFFFF"
Client "0XFF"
REJECT "0XFFF3"
Reject Code: "0XFFF5"
Sequence Number "1"
End ID "0"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

a) Client



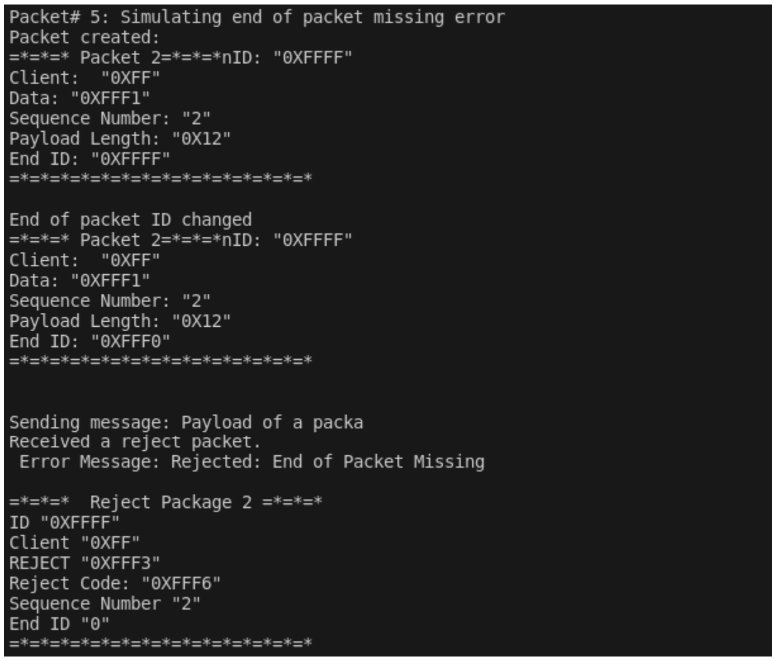
```
=*=*=* Packet 1 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 1
Payload Length: "0X10"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
Rejected: Length Mismatch
```

b) Server

Figure 8 –Output Terminal for the Case-2, a) Client, b) server.

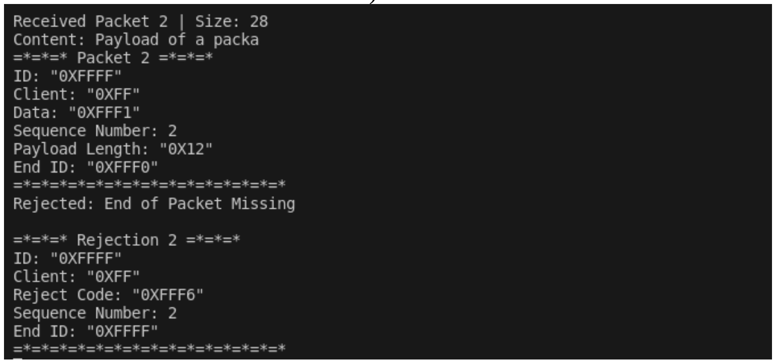
- **Case-3:** The server receives a packet which does not have the End of Packet Identifier, an error message should be generated by server and send to the client.

For example, if the last byte of the packet is xFFF0, this packet has a missing end of packet identifier error.



```
Packet# 5: Simulating end of packet missing error
Packet created:
=*=*=* Packet 2=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "2"
Payload Length: "0X12"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

End of packet ID changed
=*=*=* Packet 2=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "2"
Payload Length: "0X12"
End ID: "0XFFF0"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*


Sending message: Payload of a packa
Received a reject packet.
 Error Message: Rejected: End of Packet Missing

=*=*=*  Reject Package 2 =*=*=*
ID "0XFFFF"
Client "0XFF"
REJECT "0XFFF3"
Reject Code: "0XFFF6"
Sequence Number "2"
End ID "0"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

a) Client



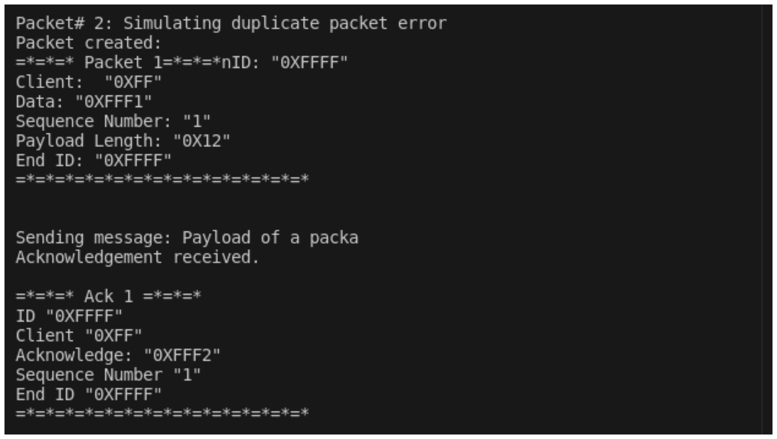
```
Received Packet 2 | Size: 28
Content: Payload of a packa
=*=*=* Packet 2 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 2
Payload Length: "0X12"
End ID: "0XFFF0"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
Rejected: End of Packet Missing

=*=*=* Rejection 2 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Reject Code: "0XFFF6"
Sequence Number: 2
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

b) Server

Figure 9 –Output Terminal for the Case-3, a) Client, b) server.

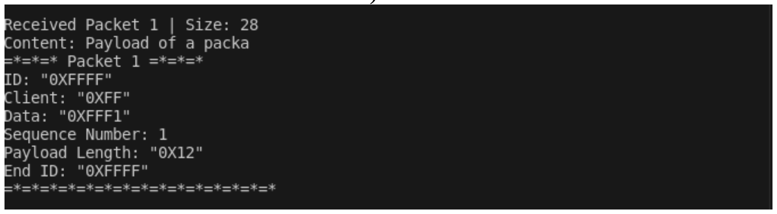
- **Case-4:** The server receives a duplicated packet (sequence number), an error message should be generated by server and send to the client.

For example, if server receives packets 0, 1 and then packet 1 again, the second packet 1 is a duplicate packet, the server will not increment the expected sequence number.



```
Packet# 2: Simulating duplicate packet error
Packet created:
=*=*=* Packet 1=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "1"
Payload Length: "0X12"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*


Sending message: Payload of a packa
Acknowledgement received.

=*=*=* Ack 1 =*=*=*
ID "0XFFFF"
Client "0XFF"
Acknowledge: "0XFFF2"
Sequence Number "1"
End ID "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

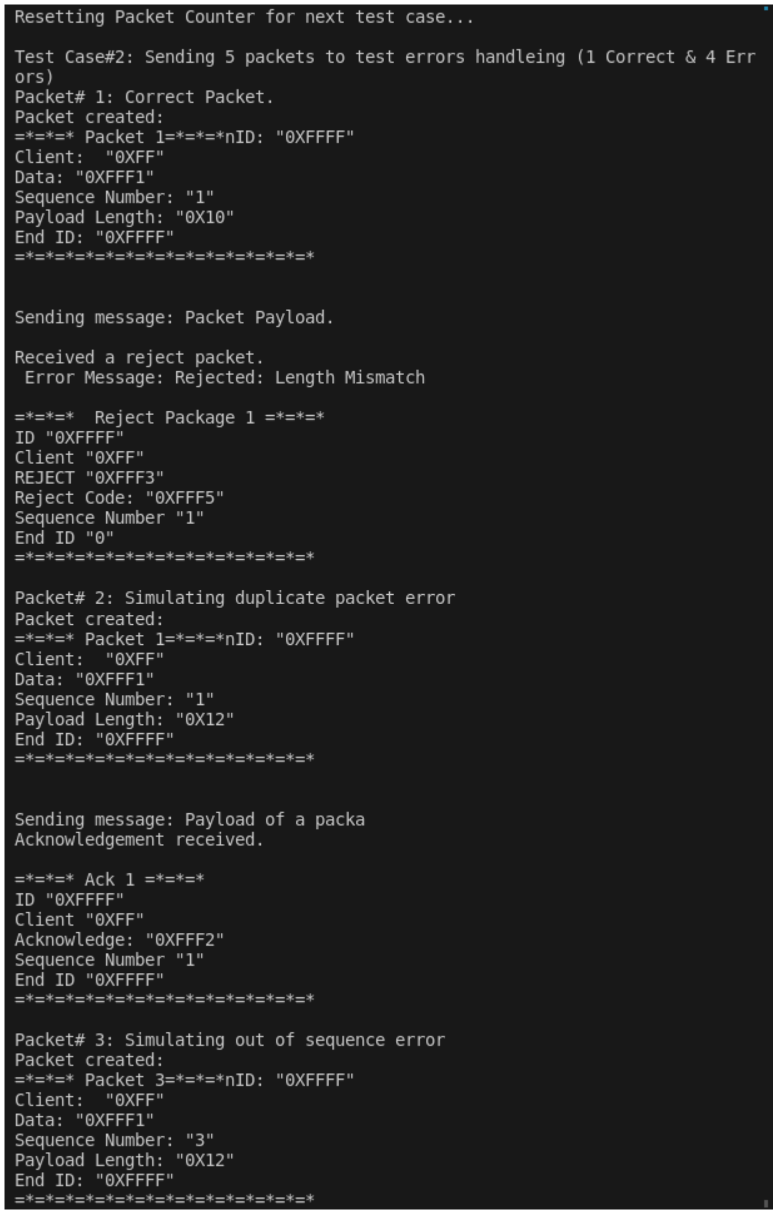
a) Client



```
Received Packet 1 | Size: 28
Content: Payload of a packa
=*=*=* Packet 1 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 1
Payload Length: "0X12"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

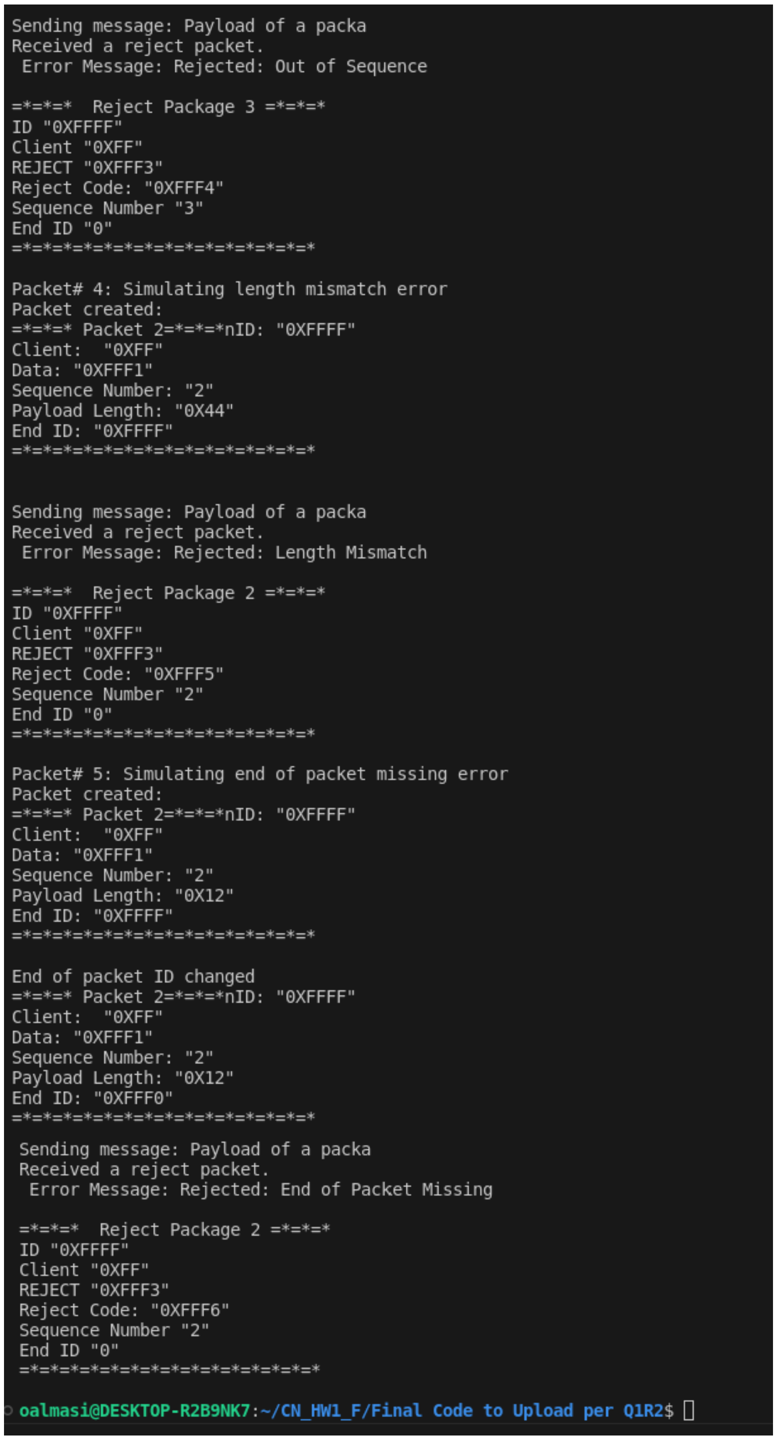
b) Server

Figure 10 –Output Terminal for the Case-4, a) Client, b) server.



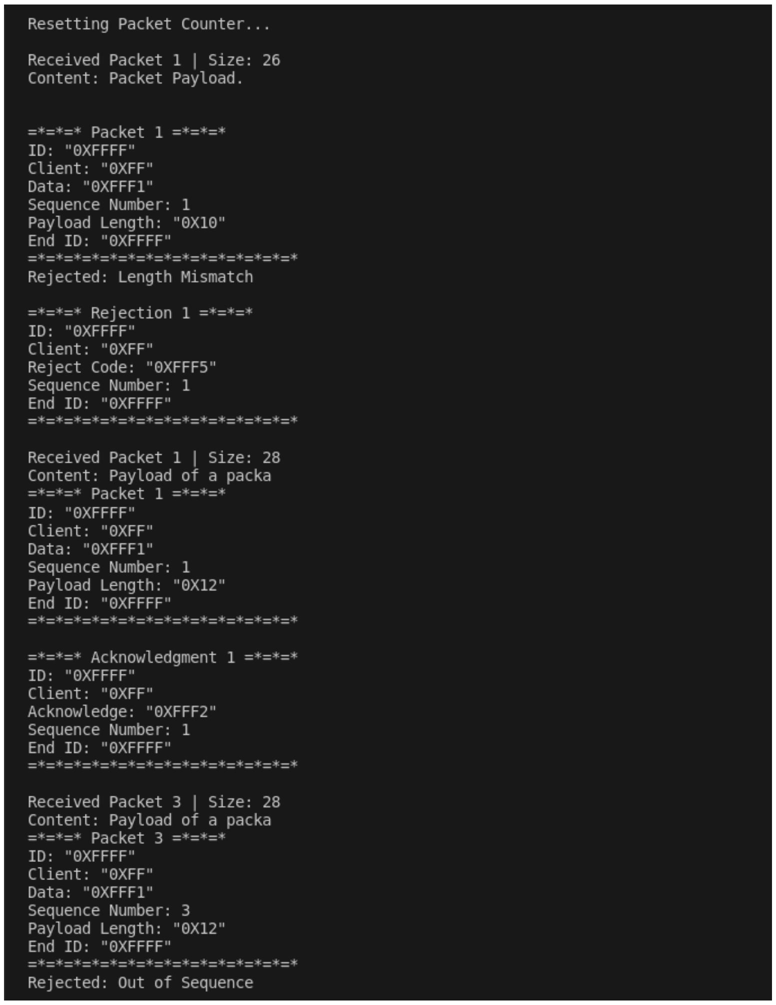
```
Resetting Packet Counter for next test case...

Test Case#2: Sending 5 packets to test errors handleing (1 Correct & 4 Err
ors)
Packet# 1: Correct Packet.
Packet created:
=*=*=* Packet 1=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "1"
Payload Length: "0X10"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*


Sending message: Packet Payload.

Received a reject packet.
 Error Message: Rejected: Length Mismatch

=*=*=*  Reject Package 1 =*=*=*
ID "0XFFFF"
Client "0XFF"
REJECT "0XFFF3"
Reject Code: "0XFFF5"
Sequence Number "1"
End ID "0"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Packet# 2: Simulating duplicate packet error
Packet created:
=*=*=* Packet 1=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "1"
Payload Length: "0X12"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*


Sending message: Payload of a packa
Acknowledgement received.

=*=*=* Ack 1 =*=*=*
ID "0XFFFF"
Client "0XFF"
Acknowledge: "0XFFF2"
Sequence Number "1"
End ID "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Packet# 3: Simulating out of sequence error
Packet created:
=*=*=* Packet 3=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "3"
Payload Length: "0X12"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

Figure 11 – Client Output Terminal for second 5 packets for error handling– All errors in series.



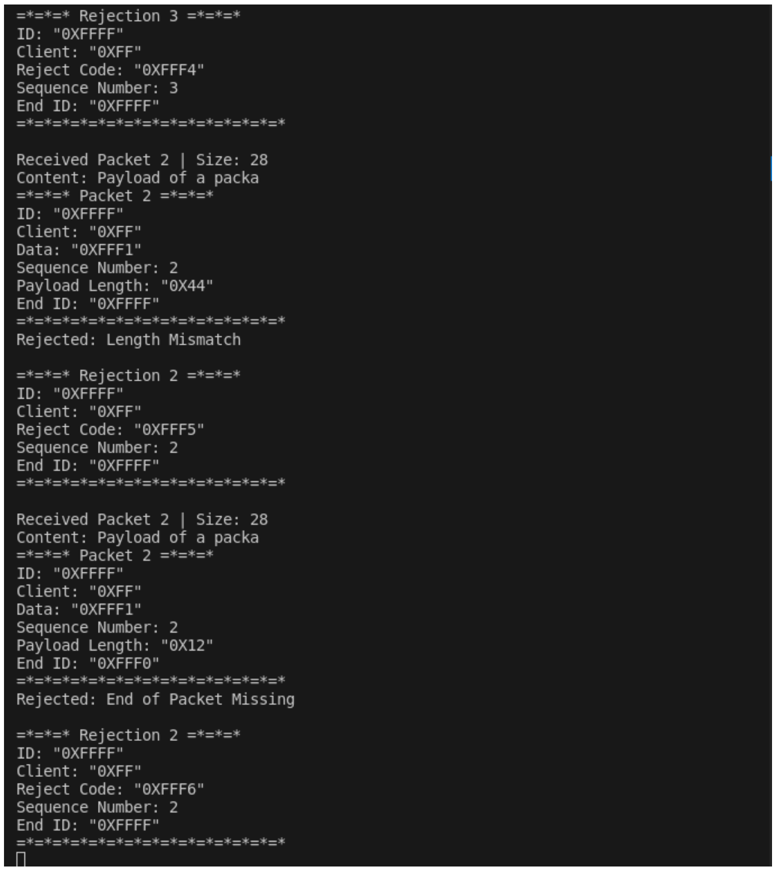
```
Sending message: Payload of a packa
Received a reject packet.
 Error Message: Rejected: Out of Sequence

=*=*=*  Reject Package 3 =*=*=*
ID "0XFFFF"
Client "0XFF"
REJECT "0XFFF3"
Reject Code: "0XFFF4"
Sequence Number "3"
End ID "0"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Packet# 4: Simulating length mismatch error
Packet created:
=*=*=* Packet 2=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "2"
Payload Length: "0X44"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*


Sending message: Payload of a packa
Received a reject packet.
 Error Message: Rejected: Length Mismatch

=*=*=*  Reject Package 2 =*=*=*
ID "0XFFFF"
Client "0XFF"
REJECT "0XFFF3"
Reject Code: "0XFFF5"
Sequence Number "2"
End ID "0"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Packet# 5: Simulating end of packet missing error
Packet created:
=*=*=* Packet 2=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "2"
Payload Length: "0X12"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

End of packet ID changed
=*=*=* Packet 2=*=*=*nID: "0XFFFF"
Client:  "0XFF"
Data: "0XFFF1"
Sequence Number: "2"
Payload Length: "0X12"
End ID: "0XFFF0"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
 Sending message: Payload of a packa
 Received a reject packet.
  Error Message: Rejected: End of Packet Missing

 =*=*=*  Reject Package 2 =*=*=*
 ID "0XFFFF"
 Client "0XFF"
 REJECT "0XFFF3"
 Reject Code: "0XFFF6"
 Sequence Number "2"
 End ID "0"
 =*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

oalmasi@DESKTOP-R2B9NK7:~/CN_HW1_F/Final Code to Upload per Q1R2$
```

Figure 12 – Continuation of Client Output Terminal for second 5 packets for error handling– All errors in series.



```
Resetting Packet Counter...

Received Packet 1 | Size: 26
Content: Packet Payload.


=*=*=* Packet 1 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 1
Payload Length: "0X10"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
Rejected: Length Mismatch

=*=*=* Rejection 1 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Reject Code: "0XFFF5"
Sequence Number: 1
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Received Packet 1 | Size: 28
Content: Payload of a packa
=*=*=* Packet 1 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 1
Payload Length: "0X12"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

=*=*=* Acknowledgment 1 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Acknowledge: "0XFFF2"
Sequence Number: 1
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Received Packet 3 | Size: 28
Content: Payload of a packa
=*=*=* Packet 3 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 3
Payload Length: "0X12"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
Rejected: Out of Sequence
```

Figure 13 – Server Output Terminal for second 5 packets for error handling– All errors in series.



```
=*=*=* Rejection 3 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Reject Code: "0XFFF4"
Sequence Number: 3
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Received Packet 2 | Size: 28
Content: Payload of a packa
=*=*=* Packet 2 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 2
Payload Length: "0X44"
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
Rejected: Length Mismatch

=*=*=* Rejection 2 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Reject Code: "0XFFF5"
Sequence Number: 2
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*

Received Packet 2 | Size: 28
Content: Payload of a packa
=*=*=* Packet 2 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Data: "0XFFF1"
Sequence Number: 2
Payload Length: "0X12"
End ID: "0XFFF0"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
Rejected: End of Packet Missing

=*=*=* Rejection 2 =*=*=*
ID: "0XFFFF"
Client: "0XFF"
Reject Code: "0XFFF6"
Sequence Number: 2
End ID: "0XFFFF"
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*
```

Figure 14 – Continuation of Server Output Terminal for second 5 packets for error handling– All errors in series.