

# CS3093D Networks Laboratory - Assignment 2

Anandu B Ajith - B180288CS

14 February 2022

## Question 1

### Instructions

- The Code for TCP server is '1/src/tcp\_server.c' and '1/src/tcp\_client.c'
- The Code for UDP server is '1/src/udp\_server.c' and '1/src/udp\_client.c'
- The fruits code is kept common for both using header file 'fruits.h' and 'fruits.c'
- Code can be compiled by running make, which will place binaries for all in bin/ folder

## Screenshots Question 1

Figure 1: Demo of TCP Client and Server

```
1 2 3 4 18:01 /: 64% muted 12% 1% BSNL WIFI 0 KB/s 0 KB/s 100%
./bin/tcp_server
anandu@inspiron [10:00:18] [~/Desktop/networkslab/assignment_03/1] [main *]
→ % ./bin/tcp_server
Waiting for connection
Accepted connection
Received Message: 1824 "SendInventory"

Sending inventory
Received Message: 1824 "Buy"

Sending message
Purchase Fruit
Received Message: 1824 "Buy"

Sending message
Purchase Fruit
ERROR
Received Message: 1824 "SendInventory"

Sending inventory
Received Message: 1824 "Bye"

Closing the connection
Waiting for connection
Accepted connection
Received Message: 1824 "Buy"

Sending message
Purchase Fruit
Received Message: 1824 "SendInventory"

Invalid message
Received Message: 1824 "SendInventory"

Sending inventory
Received Message: 1824 "Bye"

Closing the connection
Waiting for connection

1:[tmux]*

→ % ./bin/tcp_client
Welcome to store
>SendInventory
Inventory
=====
apple → 42
mango → 7
banana → 5
chikoo → 9
papaya → 2
Total 5 items
>Buy
>Enter FruitName and count
>apple 10
apple → 10 purchased successfully
>Buy
>Enter FruitName and count
>apple 100
Purchase failed (Invalid fruit or count)
>SendInventory
Inventory
=====
apple → 32
mango → 7
banana → 5
chikoo → 9
papaya → 2
Total 5 items
>Bye
Connection closed [18.2s]

anandu@inspiron [10:00:37] [~/Desktop/networkslab/assignment_03/1] [main *]
→ % ./bin/tcp_client
Welcome to store
>Buy
>Enter FruitName and count
>mango 6
mango → 6 purchased successfully
>SendInventory
Invalid request
>SendInventory
Inventory
=====
apple → 32
mango → 1
banana → 5
chikoo → 9
papaya → 2
Total 5 items
>Bye
Connection closed [12.7s]

anandu@inspiron [10:04:32] [~/Desktop/networkslab/assignment_03/1] [main *]
→ % ./bin/udp_client
>Hello
Welcome to store
>SendInventory
Inventory
=====
apple → 42
mango → 7
banana → 5
chikoo → 9
papaya → 2
Total 5 items
>Buy
>Enter FruitName and count
>apple 10
apple → 10 purchased successfully
>Buy
>Enter FruitName and count
>apple 100
Purchase failed (Invalid fruit or count)
>Bye
Connection closed [12.7s]

anandu@inspiron [10:04:40] [~/Desktop/networkslab/assignment_03/1] [main *]
→ %
```

Figure 2: Demo of UDP Client and Server

```
1 2 3 4 18:04 /: 64% muted 12% 8% BSNL WIFI 0 KB/s 0 KB/s 100%
./bin/udp_server
anandu@inspiron [10:04:19] [~/Desktop/networkslab/assignment_03/1] [main *]
→ % ./bin/udp_server
Received Message: 5 "Hello"

Received Message: 13 "SendInventory"

Sending Inventory
Received Message: 3 "Buy"

Buy case1:
Received Message: 13 "Buy[apple 10]"

Buy case2:
Received Message: 3 "Buy"

Buy case1:
Received Message: 14 "Buy[apple 100]"

Buy case2:
ERROR
Received Message: 3 "Buy"

Buy case1:
Received Message: 12 "Buy[mango 3]"

Buy case2:
Received Message: 13 "SendInventory"

Sending Inventory

1:wertera*

→ % ./bin/udp_client
>Hello
Welcome to store
>SendInventory
Inventory
=====
apple → 42
mango → 7
banana → 5
chikoo → 9
papaya → 2
Total 5 items
>Buy
>Enter FruitName and count
>apple 10
apple → 10 purchased successfully
>Buy
>Enter FruitName and count
>apple 100
Purchase failed (Invalid fruit or count)
>Bye
Connection closed [12.7s]

anandu@inspiron [10:04:32] [~/Desktop/networkslab/assignment_03/1] [main *]
→ % ./bin/udp_client
>Enter FruitName and count
>mango 3
mango → 3 purchased successfully
>SendInventory
Inventory
=====
apple → 32
mango → 4
banana → 5
chikoo → 9
papaya → 2
Total 5 items
>Bye
Connection closed [4.8s]

anandu@inspiron [10:04:40] [~/Desktop/networkslab/assignment_03/1] [main *]
→ %
```

## Question 2

### Instructions

- The code is in '2/src' named server.c and client.c
- Common structs for Packet and other constants are defined in common.h
- Code can be compiled by running 'make', which will place all the binaries in bin/ folder
- 'make dummy' command can be used to generate a dummy sequential 50MB file
- Client will print the transmission rate every 0.1 seconds in kbps, and write the same to "stats.dat"
- Server will send "file.bin" and Client will write it to "output", md5sum can be used to verify integrity
- Server implements Stop and Wait ARQ, will send packet to client and wait for ACK until timeout.
- Server calculates and outputs Round Trip Time when it receives successful ACK back.
- 'make graph' command can be used to display the "stats.dat" using gnuplot

## Screenshots Question 2

Figure 3: Screenshot of Server

```
1 5 8
18:39
✓ ACK Received
Round Trip Time: 0.000032
✓ ACK Received
Round Trip Time: 0.000027
✓ ACK Received
Round Trip Time: 0.000027
✗ Waiting for ACK Timed out
✓ ACK Received
Round Trip Time: 0.000124
✗ Waiting for ACK Timed out
✓ ACK Received
Round Trip Time: 0.000114
✓ ACK Received
Round Trip Time: 0.000112
✓ ACK Received
Round Trip Time: 0.000110
✓ ACK Received
Round Trip Time: 0.000113
✗ Waiting for ACK Timed out
✓ ACK Received
Round Trip Time: 0.000116
✓ ACK Received
Round Trip Time: 0.000120
✗ Waiting for ACK Timed out
✓ ACK Received
Round Trip Time: 0.000128
✓ ACK Received
Round Trip Time: 0.000121
✗ Waiting for ACK Timed out
✓ ACK Received
Round Trip Time: 0.000109
✓ ACK Received
Round Trip Time: 0.000064
✓ ACK Received
Round Trip Time: 0.000161
✗ Waiting for ACK Timed out
✓ ACK Received
Round Trip Time: 0.000129
✓ ACK Received
Round Trip Time: 0.000070
✓ ACK Received
Round Trip Time: 0.000053
✓ ACK Received
Round Trip Time: 0.000042
✓ ACK Received
Round Trip Time: 0.000041

Sent 129613 packets
Actual data packets: 104858

1:zsh

/bin/client
anandu@inspiron [10:36:04] [~/Desktop/networkslab/assignment_03/2_udp] [main]
→ % ./bin/client
>GiveMeVideo
Transmission rate = 209 kbps
Received 104858 packets
>bye
Connection closed [3m5.6s]

anandu@inspiron [10:39:11] [~/Desktop/networkslab/assignment_03/2_udp] [main *]
→ % md5sum file.bin output
7bc860f7a2a1ca118b82b62fb9cabb87 file.bin
7bc860f7a2a1ca118b82b62fb9cabb87 output

anandu@inspiron [10:39:16] [~/Desktop/networkslab/assignment_03/2_udp] [main *]
→ %
```

Figure 4: Graph of Transmission rate every 0.1 seconds

