Computer Networks 2
Assignment 2

Team:

G Vishal Siva Kumar	CS18BTECH11013
K Vamshi Krishna Reddy	CS18BTECH11024
P Sai Varshittha	CS18BTECH11035
T Krishna Prashanth	CS18BTECH11045

TASK 1

10 FTP requests :

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 8.92 secs (11.2147 MB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).

226 Transfer complete.
104857600 bytes received in 8.89 secs (11.2488 MB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 8.73 secs (11.4492 MB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
\150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 9.40 secs (10.6423 MB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 8.82 secs (11.3351 MB/s)
ftp>
```

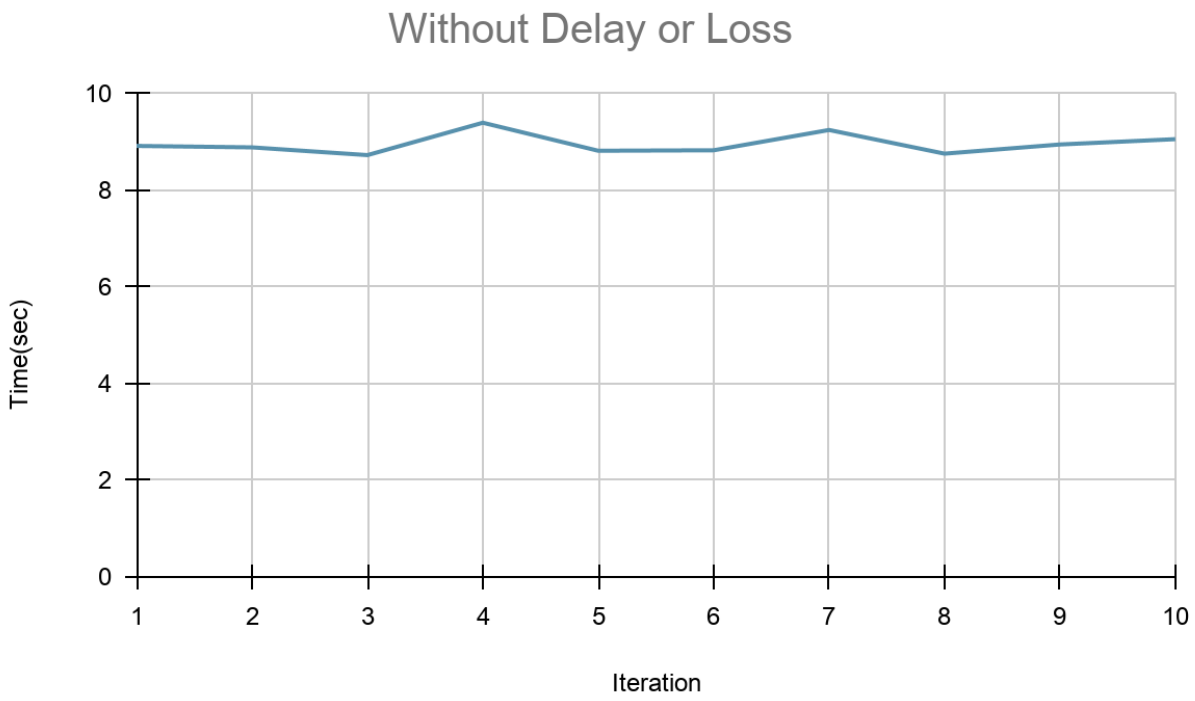
```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 8.83 secs (11.3264 MB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 9.25 secs (10.8121 MB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 8.76 secs (11.4154 MB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 8.95 secs (11.1731 MB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 9.06 secs (11.0431 MB/s)
ftp>
```


Without Delay or Loss
Time(sec)
0
2
4
6
8
10
1
2
3
4
5
6
7
8
9
10
Iteration


Without Delay or Loss
Speed(MB/s)
0
2
4
6
8
10
12
1
2
3
4
5
6
7
8
9
10
Iterations

With 50 ms delay and 5% packet loss on both sides:

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).


226 Transfer complete.
104857600 bytes received in 1326.85 secs (77.1754 kB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 1334.45 secs (76.7357 kB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 1368.70 secs (74.8154 kB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 1348.37 secs (75.9438 kB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 1414.15 secs (72.4109 kB/s)
ftp>
```

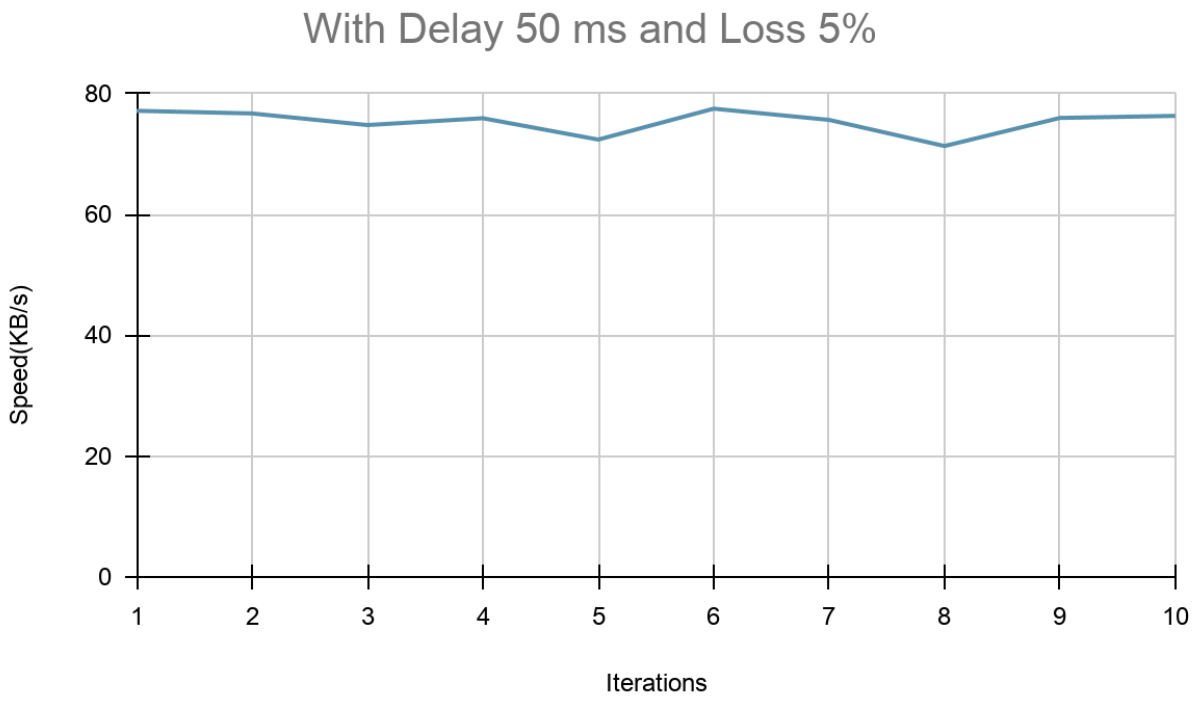
```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 1320.85 secs (77.5256 kB/s)
ftp>
```

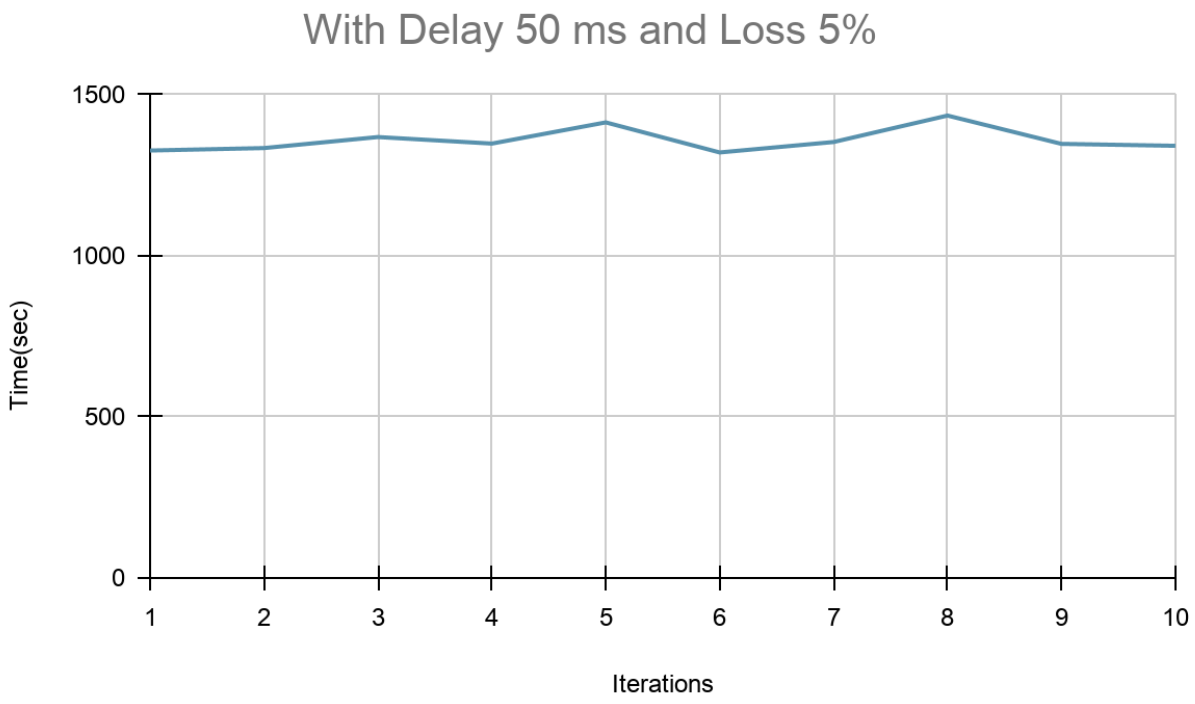
```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 1353.26 secs (75.6694 kB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 1435.33 secs (71.3427 kB/s)
ftp>
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 1347.49 secs (75.9934 kB/s)
```

```
ftp> recv CS3543_100MB
local: CS3543_100MB remote: CS3543_100MB
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for CS3543_100MB (104857600 bytes).
226 Transfer complete.
104857600 bytes received in 1341.47 secs (76.3342 kB/s)
```


With Delay 50 ms and Loss 5%
Speed(KB/s)
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10
Iterations


With Delay 50 ms and Loss 5%
Time(sec)
1500
1000
500
0
1
2
3
4
5
6
7
8
9
10
Iterations

TASK 2

How to run :

- Compile sender and receiver files using g++ with -pthread flag
- First start receiver with port as argument, then start the sender with receiver ip, port number and file name as command line arguments.
- File received at the receiver's side is named as "received_file".

Execution:

```
$ g++ receiver.cpp -pthread
$ ./a.out <port_number>

$ g++ sender.cpp -pthread
$ ./aout <receiver_ip> <port_number> <file_name>
```

Files :

- sender.cpp :
- receiver.cpp :

Header structure:

```
typedef struct pkt
{
    int retransmit;
    uint32_t ts;          /* time stamp */
    uint32_t seqNum;      /* store sequence # */
    uint32_t dataSize;    /* datasize , normally it will be 1460*/
    char buf[MAXBUFSIZE]; /* Data buffer */
} pkt;
```

Retransmit 32 bit
Timestamp 32 bit
Sequence Number 32 bit
Data Size 32 bit
DATA variable with size max size 1400

Working:

- We used c++ for implementing this assignment.
- In sender and receiver, there are two threads.
- In sender, there is a separate thread which receives acknowledgement numbers which runs parallel to the main thread which sends the data.
- In receiver, there is a separate thread which sends acknowledgements which runs parallel to the main thread which receives the file data.
- Packets are divided into chunks of 1400 bytes. Each packet is associated with a sequence number.
- Acknowledgement numbers are directly sent/received without any headers.
- A large array of nodes/packets is maintained.
- File size is first sent from sender to receiver.
- Receiver waits for all the *filesize* bytes to be received.
- Child thread in receiver continuously iterates over the array checking whether a packet is received. If not, it sends the corresponding Seq number as Ack to the sender as a request to send it again (NACK functionality). The converse of it happens on the sender side.
- When the complete file is received by the receiver, ACK number of -1 is sent signaling the sender that the transmission is complete.
- To ensure thread safety, we used C++ std::mutex.
- We tried to suppress the link bandwidth congestion by delaying each packet transfer from the sender's side by a small amount using "usleep()".
- Flow control is maintained as the receiver maintains a node array which is used to request the missed packets with the help of sequence numbers.
- Packet loss is detected by the receiver as it checks the node array maintained and requests the sender for the missing packets.

Measurements

Measured at receiver side

Without delay or packet loss:

Iteration	Time (sec)	Speed (MB/s)
1	18.1606	5.50643
2	17.8876	5.59047
3	18.2913	5.46707
4	18.6606	5.35889
5	18.5541	5.38965
6	18.9135	5.28722
7	19.0845	5.23986
8	19.7214	5.07064
9	19.1097	5.23925
10	18.891	5.29353
Overall		

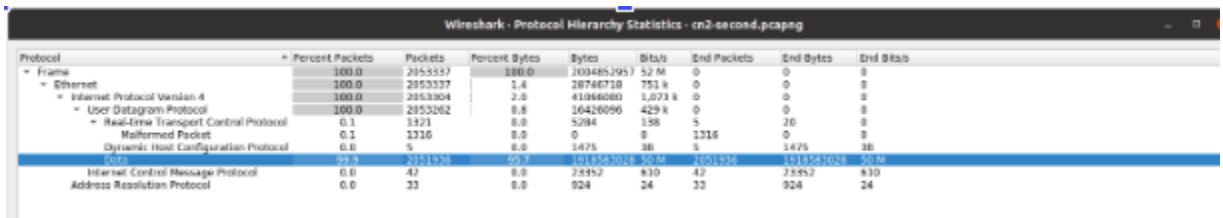
With delay and packet loss:

Iteration	Time (sec)	Speed (MB/s)
1	36.4805	2.74119
2	18.475	5.41273
3	18.2528	5.4786
4	18.6327	5.3669
5	19.6004	5.10193
6	19.2935	5.18308
7	20.1163	4.9711
8	19.1451	5.22326
9	19.6695	5.084
10	19.0969	5.23645
Overall		

Wireshark statistics

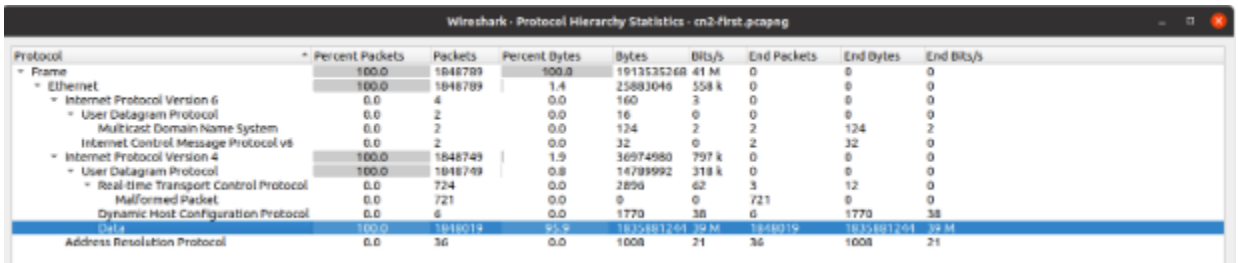
We applied wireshark on the receiver interface to measure the throughput.
We measured once for 10 attempts of our file transfer without delay or packet loss and once again for 10 attempts of file transfer with delay and packet loss.

Without delay or packet loss:
Overall UDP data throughput: 6.25 MB/s (50 Mbps)
This also includes the ACK packet data



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	2053337	100.0	2034852957	52 M	0	0	0
Ethernet	100.0	2053337	1.4	28746718	751 k	0	0	0
Internet Protocol Version 4	100.0	2053304	2.0	41066080	1,071 k	0	0	0
User Datagram Protocol	100.0	2053262	0.8	16426096	429 k	0	0	0
Real-time Transport Control Protocol	0.1	1321	0.0	5284	138	5	20	0
Malformed Packet	0.1	1316	0.0	0	0	1316	0	0
Dynamic Host Configuration Protocol	0.0	5	0.0	1475	38	5	1475	38
Data	99.3	2039136	99.3	1913535268	50 M	2039136	1913535268	50 M
Internet Control Message Protocol	0.0	42	0.0	23352	610	42	23352	610
Address Resolution Protocol	0.0	33	0.0	924	24	33	924	24

With delay and packet loss:
Overall UDP data throughput: 4.875 MB/s (39 Mbps)
This also includes the ACK packet data



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	1848789	100.0	1913535268	41 M	0	0	0
Ethernet	100.0	1848789	1.4	25883046	558 k	0	0	0
Internet Protocol Version 6	0.0	4	0.0	160	3	0	0	0
User Datagram Protocol	0.0	2	0.0	16	0	0	0	0
Multicast Domain Name System	0.0	2	0.0	124	2	2	124	2
Internet Control Message Protocol v6	0.0	2	0.0	32	0	2	32	0
Internet Protocol Version 4	100.0	1848749	1.9	36974980	797 k	0	0	0
User Datagram Protocol	100.0	1848749	0.8	14789992	318 k	0	0	0
Real-time Transport Control Protocol	0.0	724	0.0	2896	62	3	12	0
Malformed Packet	0.0	721	0.0	0	0	721	0	0
Dynamic Host Configuration Protocol	0.0	6	0.0	1770	38	6	1770	38
Data	100.0	1848019	95.9	1835881264	39 M	1848019	1835881264	39 M
Address Resolution Protocol	0.0	36	0.0	1008	21	36	1008	21

Note:

Receiver must be started before the sender as receiver will be waiting to receive the file size from the sender which is sent by the sender when it starts.