UDP client:-

```
#include <stdio.h>
#include <arpa/inet.h>
#include <fcntl.h>
#include <unistd.h>
int main()
{
int soc, n;
Char buffer[1024], msg[100];
Struct Sockaddr_in addr;
Soc = Socket (PF_INET, SOCK_DPRM, 0);
addr.sin_family = AF_INET;
addr.sin_port = htons(7891);
addr.sin_addr.s_addr = inet_addr("127.0.0.1");
Printf("\n Client is connected to server");
Printf("\n enter message");
Scanf("%s", msg);
Send to (soc, msg, sizeof(msg), 0, (Struct Sockaddr
)&addr, sizeof(addr));
Close(soc);
return 0;
}
```

Output:-

Client is connected to server
Enter message: Hi

ns3 Program

TCP Performance in wired Network

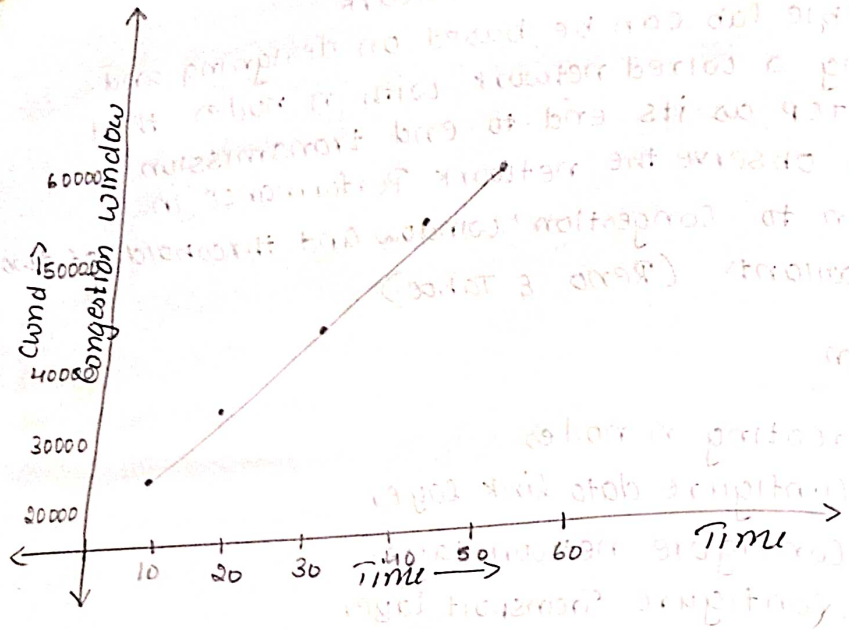
Possible lab can be based on designing and simulating a wired network with n nodes that utilizes TCP as its end-to-end transmission protocol observe the network performance in addition to congestion window and threshold of two TCP variants (Reno & Tahoe)

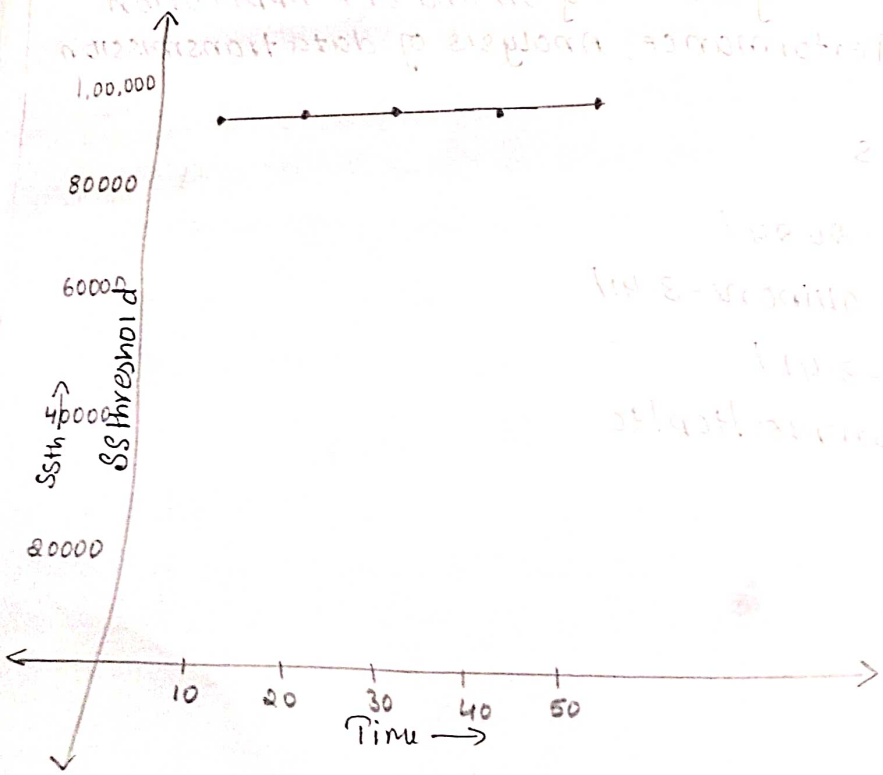
Algorithm

Step 1 : Creating n nodes
Step 2 : Configure data link layer
Step 3 : Configure network layer
Step 4 : Configure Transport layer
Step 5 : Sending Data by ON and OFF Application
Step 6 : Performance, Analysis of data transmission

Commands

```
Cd ns3-2004/
Cd ns-allinone-3.41/
Cd ns-3.41/
Cp examples/tcp/tc
```





Conclusion:

Congestion window and ssthrethold:

Analyzing the behaviour of the congestion window and ssthreshold for both TCP Reno & Tahoe variants under different network loads & scenarios

TCP (Reno Vis Tahoe)

Comparing the Performance & behavior of TCP Reno & Tahoe in term of their to handle congestion & recover from loss

ns3 Program

TCP Performance in wireless Network

Possible lab can be based on designing and simulating a wireless network that utilizes TCP as its end to end transmission Protocol, and determine bandwidth and traffic

Algorithm

Step1: Creating n nodes

Step 2: Configure data link layer

Step 3: Configure network layer

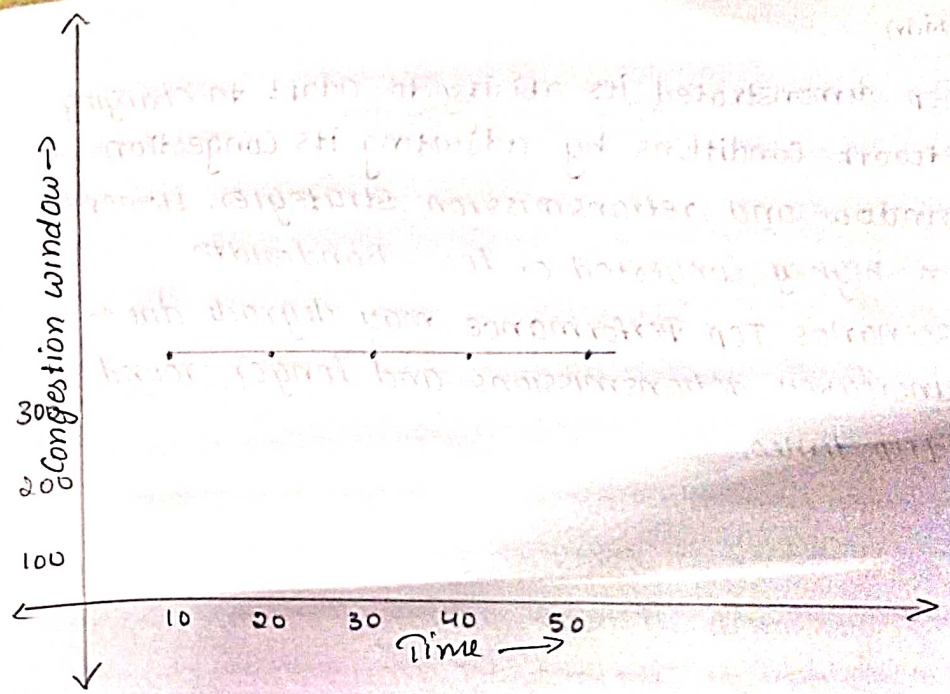
Step 4: Configure transport layer

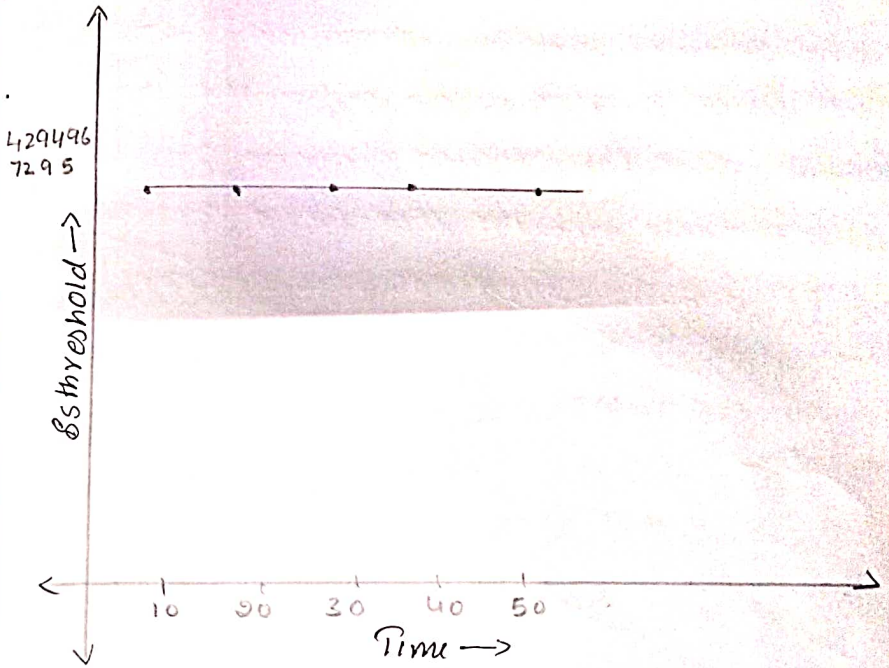
Step 5: Sending Data by ON and OFF application

Step 6: Performance, Analysis of data transmission

Commands

```
cd  ns-2024/
cd  ns-allinone-3-41/
cd  ns-3.41/
cp  examples/tcp/
```





Conclusion

Tcp demonstrated its ability to adapt to changing network conditions by adjusting its congestion window and retransmission strategies. However, in highly congested or low-bandwidth scenarios TCP Performance may degrade due to increased retransmissions and longer round trip times

ns3 Program

IEEE 802.11n wifi network with multiple types of services:-

Possible lab can be based on designing and simulating an IEEE 802.1 in wifi network with multiple TOS. Considering the number of stations. the HT MCS value (0 to 7) the channel width (20 or 40 MHz) and grand interval (long or short) observe the network performance in terms of aggregated UDP throughput. The user can also specify the distance between the access point and the stations(in meters) and can specify whether RTS/CTS is used or not.

Algorithm

Step 1 : Creating n nodes
Step 2 : Configure data link layer
Step 3 : Configure Network layer
Step 4 : Configure transport layer
Step 5 : Sending data by ON & OFF application
Step 6 : Performance Analysis of data transmission

Commands

```
Cd ns-2024/
Cd ns-allinone-3.41/
Cd ns.3.41/
Cp examples/wireless/wifi-multi-tos.cc scratch/
gedit scratch/wifi-multi-tos.cc
```

Output:

Aggregated throughput : 46.1266 Mbit/s
if useRts = false

Aggregated throughput : 44.9561 Mbit/s
if useRts = True

Conclusion

Designing and simulating an IEEE 802.11n Wi-Fi network with multiple types of services presents an opportunity to evaluate network performance comprehensively. Through varying parameter such as the number of stations, HT MCS values, channel width, guard interval, distance between access point and stations and the utilization of RTS/CTS one can gauge the impact on aggregated UDP throughput. This lab execute excercise provides valuable insight into the real-world application of IEEE 802.11n standards how network configuration influence performance under different scenarios