Fall 2017

San Jose State University
Department of Electrical Engineering

EE-286-01- WIRELESS AND MOBILE NETWORKING

PROJECT-1

Group-2

Submitted to –

Prof. Pedro Santacruz

Submitted by –

Name – **Shivang Singh**
SJSU ID – **010939851**

-- October 10th, 2017 --

Part - 1 - Point to Point Communication

Ans.1. The distance between the two nodes in the network – 100 m.

Ans.2. We are running for 0.570 seconds.

Ans.3. The transmitter start transmitting data packets from the 11th packet from 10.1.1.1 to 10.1.1.2.

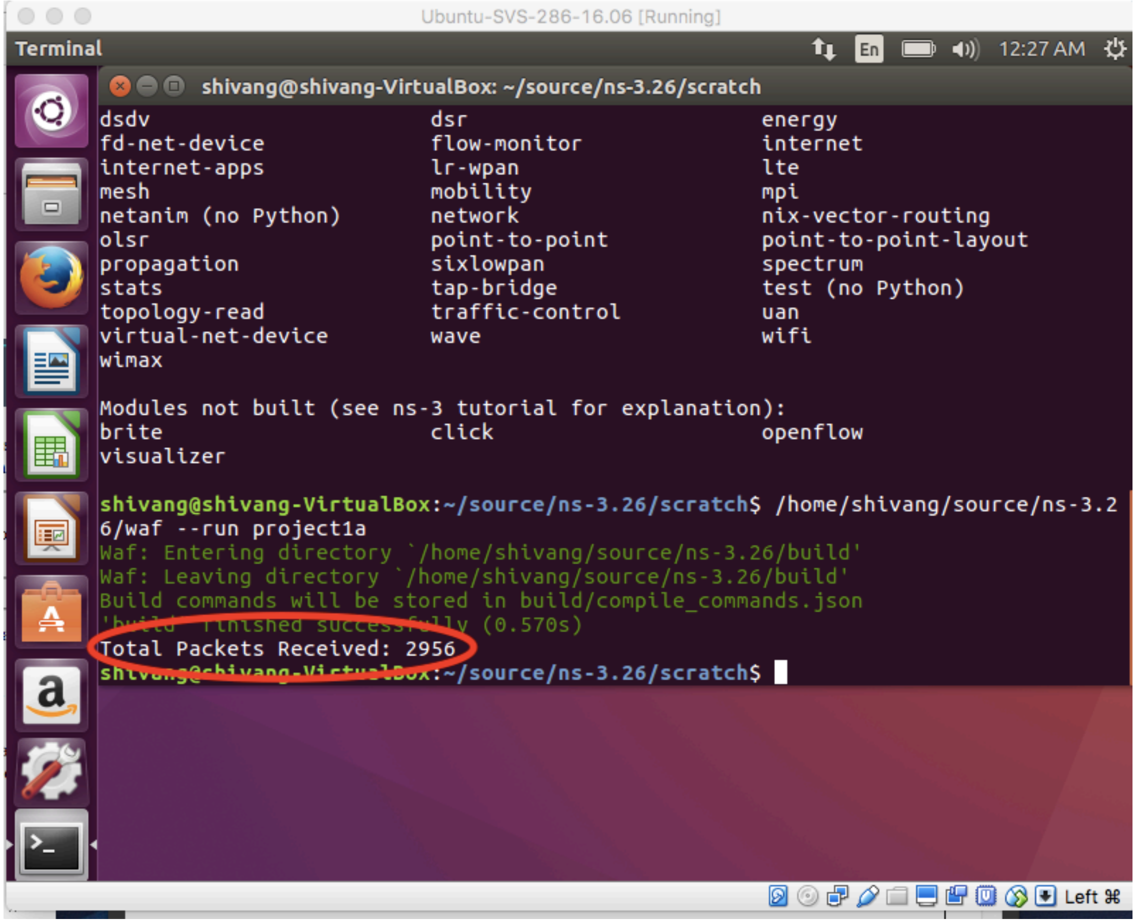
Ans.4. The throughput of the network is –

Throughput = Total Size/Simulation Time

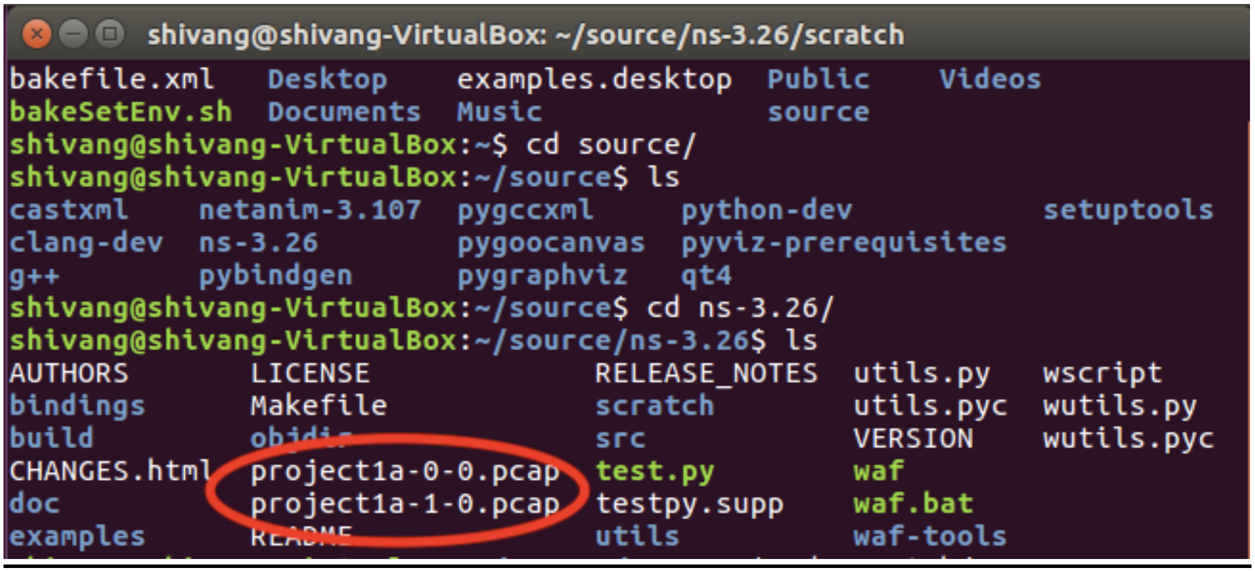
= (Packet Size * Total number of packets)/Simulation Time

= $(1064*8 * 2956) / (14.994465) * 10^{-6}$

= 1.6780 Mbps

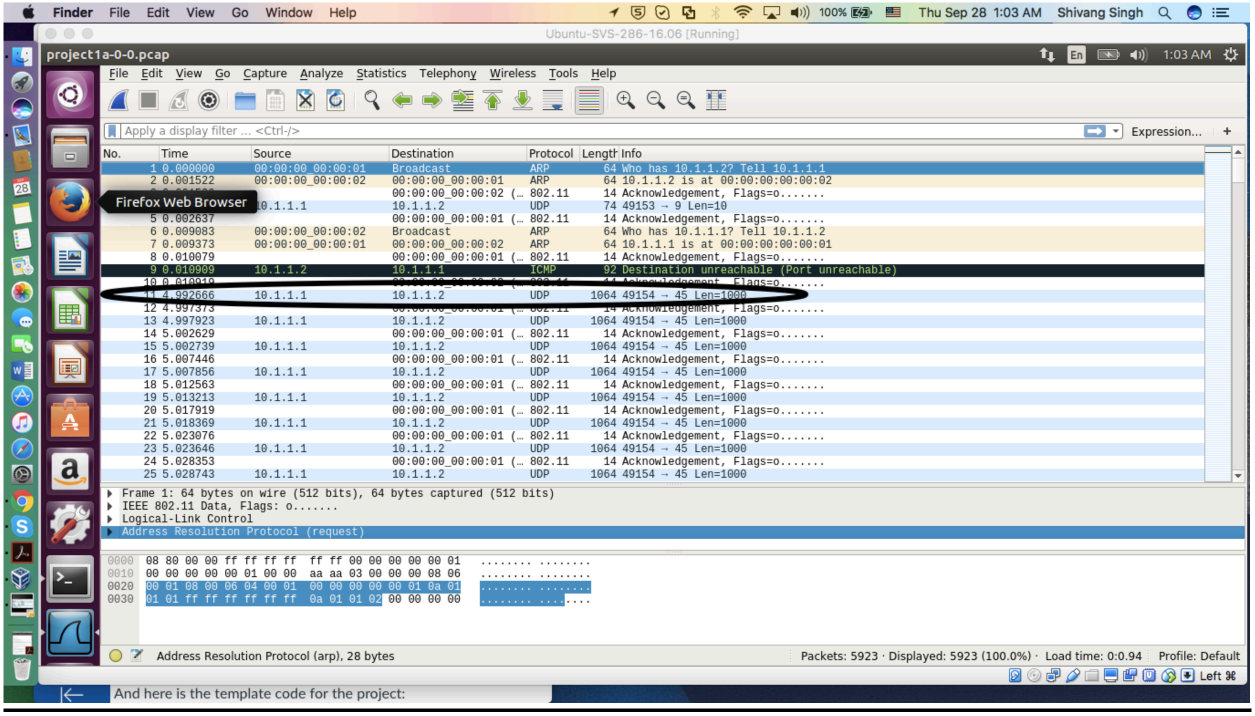

Ubuntu-SVS-286-16.06 [Running]
Terminal
En
12:27 AM
shivang@shivang-VirtualBox: ~/source/ns-3.26/scratch
Left ⌘

```
dsdv                      dsr                       energy
fd-net-device             flow-monitor              internet
internet-apps             lr-wpan                   lte
mesh                      mobility                  mpi
netanim (no Python)       network                   nix-vector-routing
olsr                      point-to-point            point-to-point-layout
propagation               sixlowpan                 spectrum
stats                     tap-bridge                test (no Python)
topology-read             traffic-control           uan
virtual-net-device        wave                      wifi
wimax

Modules not built (see ns-3 tutorial for explanation):
brite                     click                     openflow
visualizer

shivang@shivang-VirtualBox:~/source/ns-3.26/scratch$ /home/shivang/source/ns-3.2
6/waf --run project1a
Waf: Entering directory `/home/shivang/source/ns-3.26/build'
Waf: Leaving directory `/home/shivang/source/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.570s)
Total Packets Received: 2956
shivang@shivang-VirtualBox:~/source/ns-3.26/scratch$ 
```



shivang@shivang-VirtualBox: ~/source/ns-3.26/scratch

```
bakefile.xml   Desktop    examples.desktop  Public    Videos
bakeSetEnv.sh  Documents  Music             source
shivang@shivang-VirtualBox:~$ cd source/
shivang@shivang-VirtualBox:~/source$ ls
castxml    netanim-3.107  pygccxml     python-dev            setuptools
clang-dev  ns-3.26        pygoocanvas  pyviz-prerequisites
g++        pybindgen      pygraphviz   qt4
shivang@shivang-VirtualBox:~/source$ cd ns-3.26/
shivang@shivang-VirtualBox:~/source/ns-3.26$ ls
AUTHORS       LICENSE             RELEASE_NOTES  utils.py   wscript
bindings      Makefile            scratch        utils.pyc  wutils.py
build         objdir              src            VERSION    wutils.pyc
CHANGES.html  project1a-0-0.pcap  test.py        waf
doc           project1a-1-0.pcap  testpy.supp    waf.bat
examples      README              utils          waf-tools
```



Finder File Edit View Go Window Help
Thu Sep 28 1:03 AM Shivang Singh
Ubuntu-SVS-286-16.06 [Running]
project1a-0-0.pcap
1:03 AM
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Firefox Web Browser

```
No.  Time        Source             Destination           Protocol Length Info
   1 0.000000    00:00:00_00:00:01  Broadcast             ARP        64 Who has 10.1.1.2? Tell 10.1.1.1
   2 0.001522    00:00:00_00:00:02  00:00:00_00:00:01     ARP        64 10.1.1.2 is at 00:00:00:00:00:02
                                    00:00:00_00:00:02 (… 802.11      14 Acknowledgement, Flags=o.......
                  0.1.1.1           10.1.1.2              UDP        74 49153 → 9 Len=10
   5 0.002637                       00:00:00_00:00:01 (… 802.11      14 Acknowledgement, Flags=o.......
   6 0.009083    00:00:00_00:00:02  Broadcast             ARP        64 Who has 10.1.1.1? Tell 10.1.1.2
   7 0.009373    00:00:00_00:00:01  00:00:00_00:00:02     ARP        64 10.1.1.1 is at 00:00:00:00:00:01
   8 0.010079                       00:00:00_00:00:01 (… 802.11      14 Acknowledgement, Flags=o.......
   9 0.010909    10.1.1.2           10.1.1.1              ICMP       92 Destination unreachable (Port unreachable)
  10 0.010919                       00:00:00_00:00:02 (… 802.11      14 Acknowledgement, Flags=o.......
  11 4.992666    10.1.1.1           10.1.1.2              UDP      1064 49154 → 45 Len=1000
  12 4.997373                       00:00:00_00:00:01 (… 802.11      14 Acknowledgement, Flags=o.......
  13 4.997923    10.1.1.1           10.1.1.2              UDP      1064 49154 → 45 Len=1000
  14 5.002629                       00:00:00_00:00:01 (… 802.11      14 Acknowledgement, Flags=o.......
  15 5.002739    10.1.1.1           10.1.1.2              UDP      1064 49154 → 45 Len=1000
  16 5.007446                       00:00:00_00:00:01 (… 802.11      14 Acknowledgement, Flags=o.......
  17 5.007856    10.1.1.1           10.1.1.2              UDP      1064 49154 → 45 Len=1000
  18 5.012563                       00:00:00_00:00:01 (… 802.11      14 Acknowledgement, Flags=o.......
  19 5.013213    10.1.1.1           10.1.1.2              UDP      1064 49154 → 45 Len=1000
  20 5.017919                       00:00:00_00:00:01 (… 802.11      14 Acknowledgement, Flags=o.......
  21 5.018369    10.1.1.1           10.1.1.2              UDP      1064 49154 → 45 Len=1000
  22 5.023076                       00:00:00_00:00:01 (… 802.11      14 Acknowledgement, Flags=o.......
  23 5.023646    10.1.1.1           10.1.1.2              UDP      1064 49154 → 45 Len=1000
  24 5.028353                       00:00:00_00:00:01 (… 802.11      14 Acknowledgement, Flags=o.......
  25 5.028743    10.1.1.1           10.1.1.2              UDP      1064 49154 → 45 Len=1000

▸ Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
▸ IEEE 802.11 Data, Flags: o.......
▸ Logical-Link Control
▸ Address Resolution Protocol (request)

0000  08 80 00 00 ff ff ff ff  ff ff 00 00 00 00 00 01   ........ ........
0010  00 00 00 00 00 01 00 00  aa aa 03 00 00 00 08 06   ........ ........
0020  00 01 08 00 06 04 00 01  00 00 00 00 00 01 0a 01   ........ ........
0030  01 01 ff ff ff ff ff ff  0a 01 01 02 00 00 00 00   ........ ........
```

Address Resolution Protocol (arp), 28 bytes — Packets: 5923 · Displayed: 5923 (100.0%) · Load time: 0:0.94 — Profile: Default

And here is the template code for the project:

Part-2- Three-Node Network

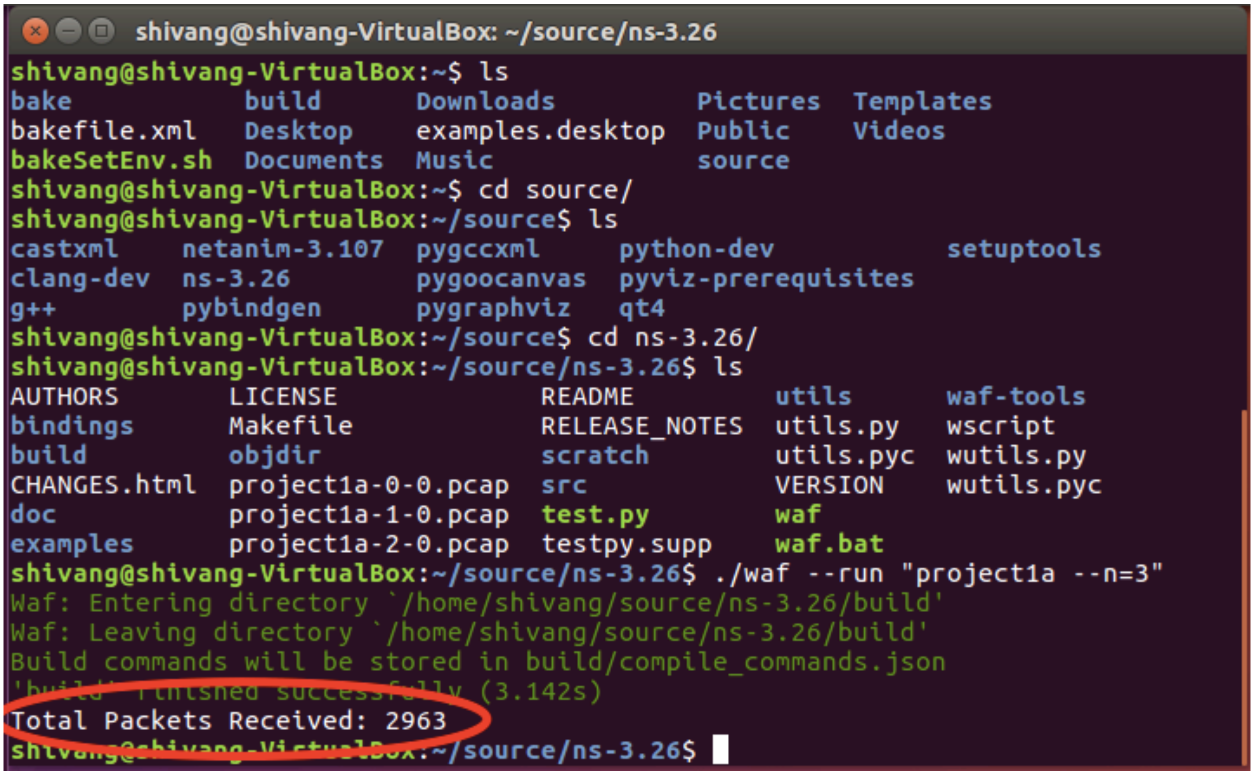
Ans.1. The throughput of the network is –

Throughput = Total Size * Total number of packets/Simulation Time
= (1064*8*2963)/ (14.992452) *10^{-6}
= 1.68225 Mbps

Ans.2. ***Yes***, we are receiving more number of packets than before, when we were doing with a single node.
We are now receiving 2963 packets, rather than 2956, which are 7 packets more than the previous one.

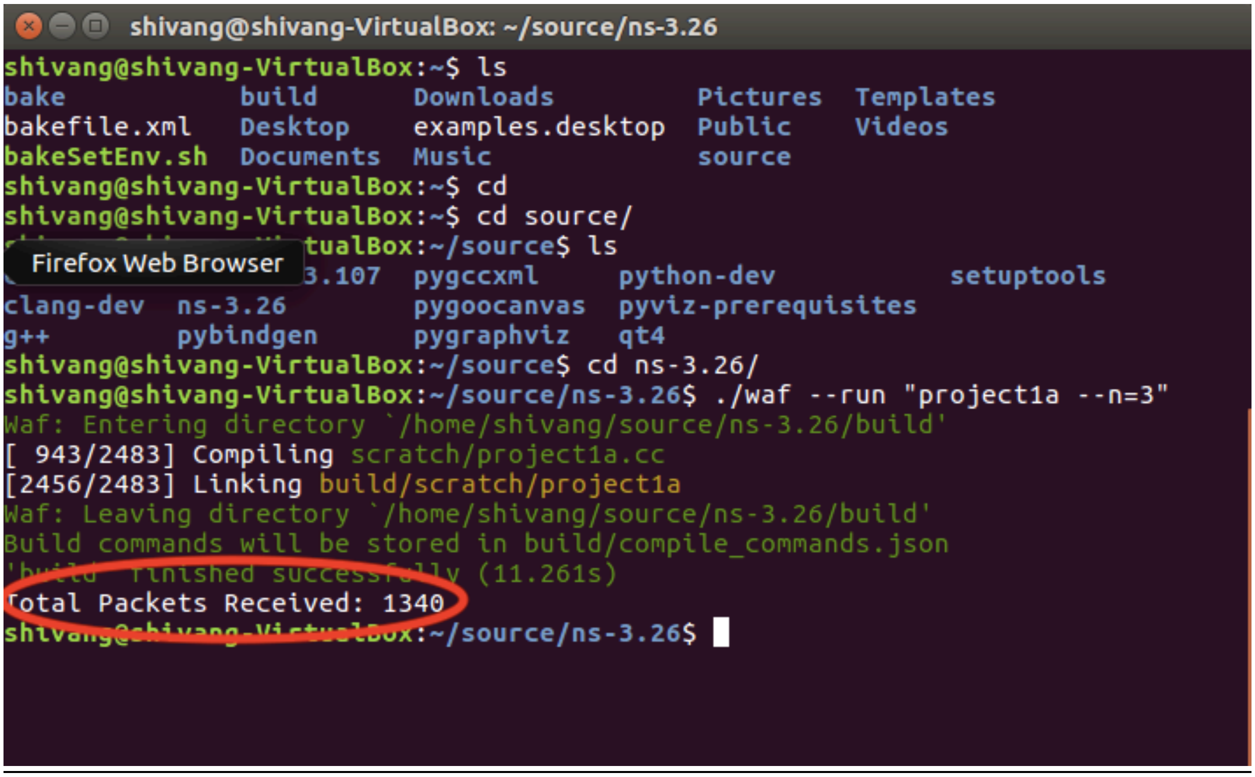
Ans.3. The total time taken for transmitting all packets (2963) is 14.992699 seconds.
Time for a single packet = Packet Size/Bit Rate
= 1064*8/1.68225*10^{-6}
= 0.0050598 seconds.

The factors which determine how long does it take to calculate time for transmission of packets includes the processing time of the router; the time spent by the packet in routing queues and the distance between the nodes because if the distance increases, since it increases the propagation delay too.

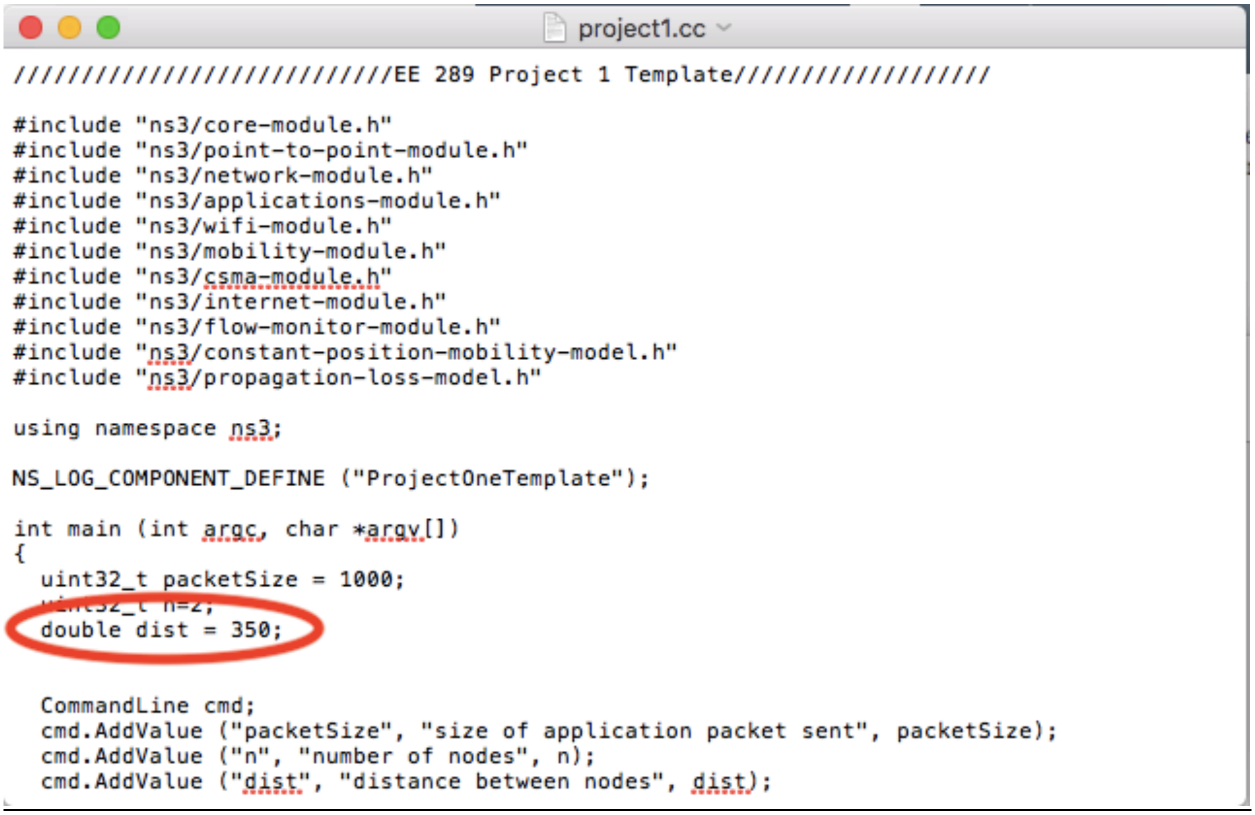


```
shivang@shivang-VirtualBox:~$ ls
bake           build      Downloads         Pictures  Templates
bakefile.xml   Desktop    examples.desktop  Public    Videos
bakeSetEnv.sh  Documents  Music             source
shivang@shivang-VirtualBox:~$ cd source/
shivang@shivang-VirtualBox:~/source$ ls
castxml    netanim-3.107  pygccxml     python-dev            setuptools
clang-dev  ns-3.26        pygoocanvas  pyviz-prerequisites
g++        pybindgen      pygraphviz   qt4
shivang@shivang-VirtualBox:~/source$ cd ns-3.26/
shivang@shivang-VirtualBox:~/source/ns-3.26$ ls
AUTHORS       LICENSE             README         utils      waf-tools
bindings      Makefile            RELEASE_NOTES  utils.py   wscript
build         objdir              scratch        utils.pyc  wutils.py
CHANGES.html  project1a-0-0.pcap  src            VERSION    wutils.pyc
doc           project1a-1-0.pcap  test.py        waf
examples      project1a-2-0.pcap  testpy.supp    waf.bat
shivang@shivang-VirtualBox:~/source/ns-3.26$ ./waf --run "project1a --n=3"
Waf: Entering directory `/home/shivang/source/ns-3.26/build'
Waf: Leaving directory `/home/shivang/source/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.142s)
Total Packets Received: 2963
shivang@shivang-VirtualBox:~/source/ns-3.26$
```

Part – 3 – Increasing the distance



```
shivang@shivang-VirtualBox:~$ ls
bake           build      Downloads         Pictures  Templates
bakefile.xml   Desktop    examples.desktop  Public    Videos
bakeSetEnv.sh  Documents  Music             source
shivang@shivang-VirtualBox:~$ cd
shivang@shivang-VirtualBox:~$ cd source/
shivang@shivang-VirtualBox:~/source$ ls
                   3.107  pygccxml      python-dev            setuptools
clang-dev  ns-3.26        pygoocanvas   pyviz-prerequisites
g++        pybindgen      pygraphviz    qt4
shivang@shivang-VirtualBox:~/source$ cd ns-3.26/
shivang@shivang-VirtualBox:~/source/ns-3.26$ ./waf --run "project1a --n=3"
Waf: Entering directory `/home/shivang/source/ns-3.26/build'
[ 943/2483] Compiling scratch/project1a.cc
[2456/2483] Linking build/scratch/project1a
Waf: Leaving directory `/home/shivang/source/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (11.261s)
Total Packets Received: 1340
shivang@shivang-VirtualBox:~/source/ns-3.26$ 
```



```
/////////////////////////////EE 289 Project 1 Template//////////////////

#include "ns3/core-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/wifi-module.h"
#include "ns3/mobility-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/flow-monitor-module.h"
#include "ns3/constant-position-mobility-model.h"
#include "ns3/propagation-loss-model.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("ProjectOneTemplate");

int main (int argc, char *argv[])
{
  uint32_t packetSize = 1000;
  uint32_t n=2;
  double dist = 350;


  CommandLine cmd;
  cmd.AddValue ("packetSize", "size of application packet sent", packetSize);
  cmd.AddValue ("n", "number of nodes", n);
  cmd.AddValue ("dist", "distance between nodes", dist);
```

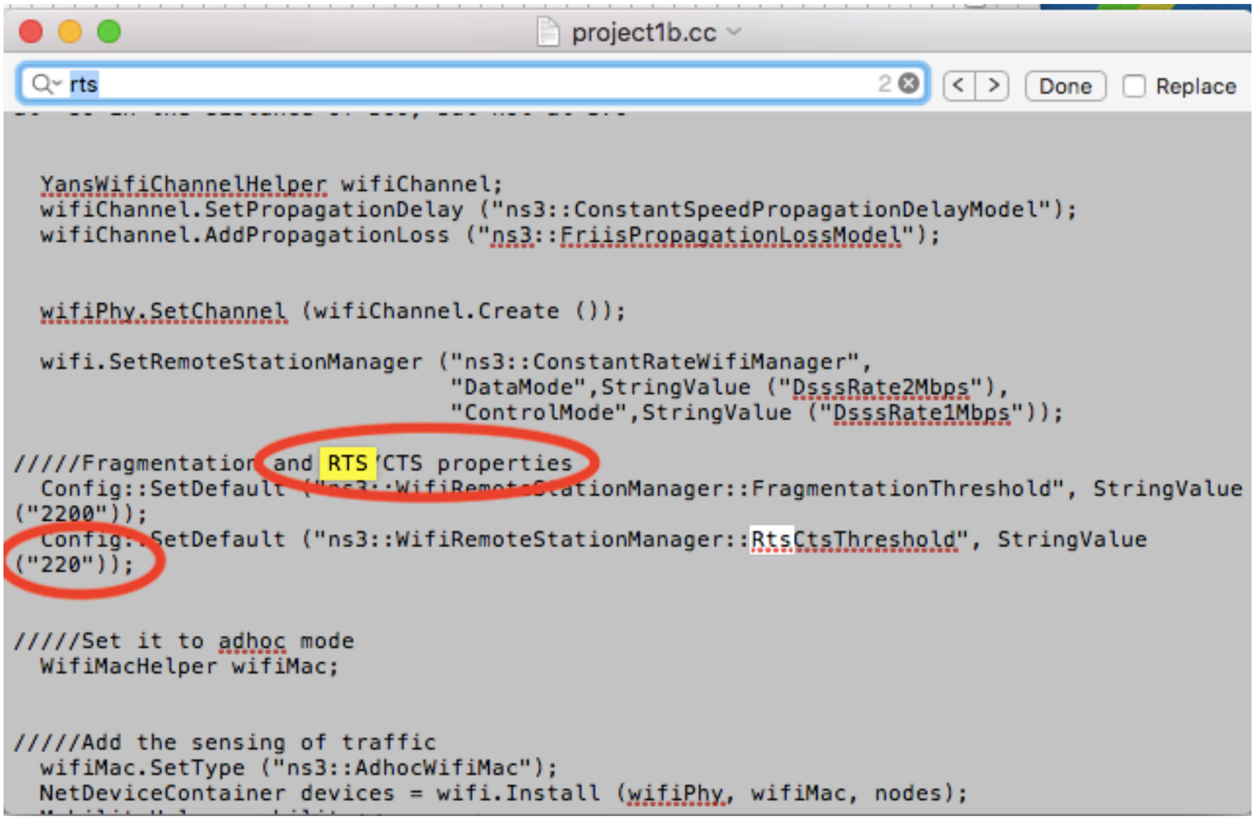
Ans.1. The number of packets I received are 1340.

Ans.2. Initially when the distance was 100 then the number of packets were 2956; but when I changed the distance to 350 then the number of packets received was 1340. There is a decrement of 1616 packets.

This is due to the increase in distance, packet loss due to collision has increased and the sender is unable to sense that the packet transfer, due to which packet number is decreased.

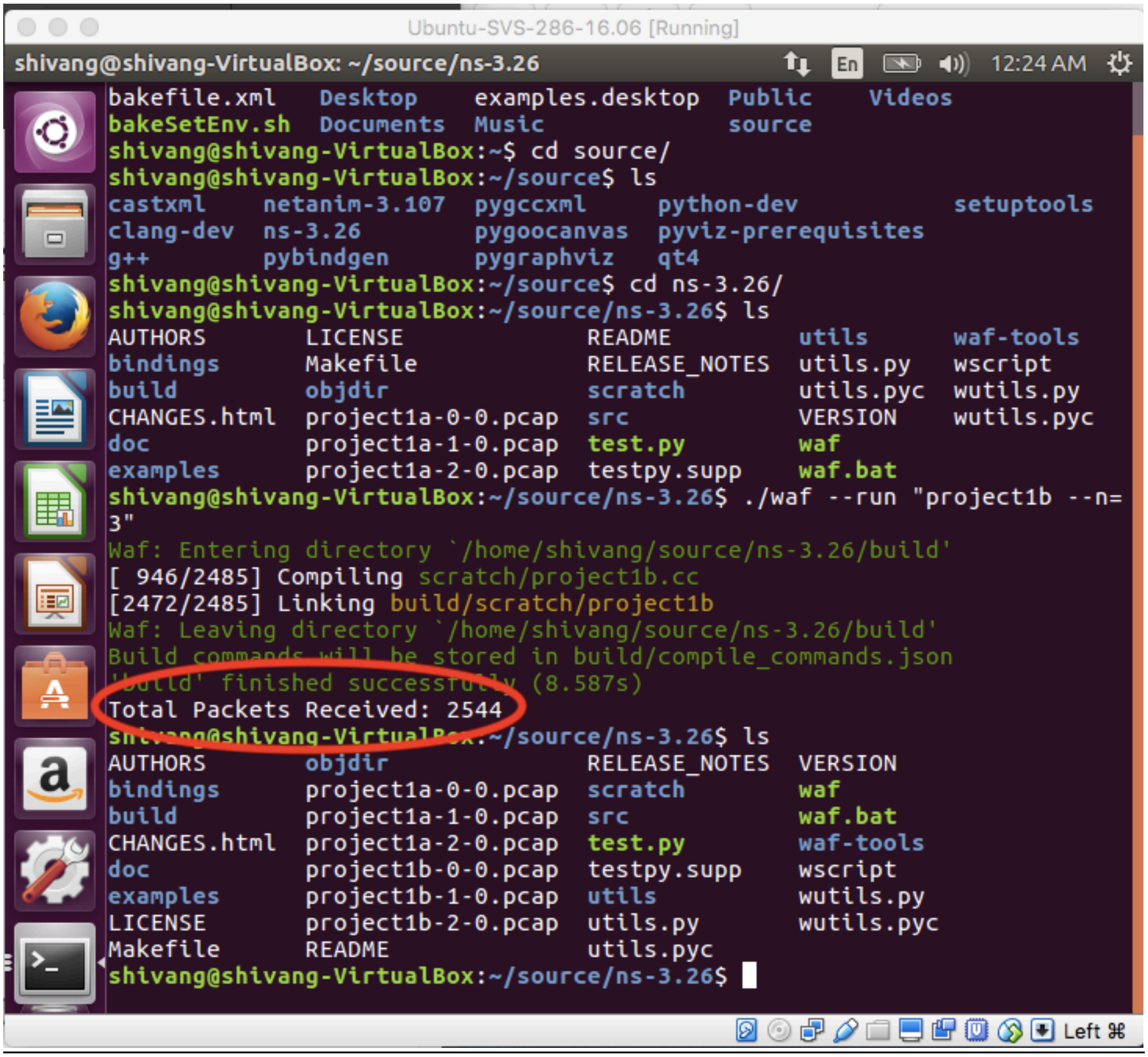
Part-4- RTS/CTS

Ans.1. In the code file given to us, I changed line number 79. In the "79th line", the RTS/CTS value was set to 2200, but we have set the value to be 220, since it was quite less when compared to the threshold value.



```
  YansWifiChannelHelper wifiChannel;
  wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
  wifiChannel.AddPropagationLoss ("ns3::FriisPropagationLossModel");


  wifiPhy.SetChannel (wifiChannel.Create ());

  wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager",
                                "DataMode",StringValue ("DsssRate2Mbps"),
                                "ControlMode",StringValue ("DsssRate1Mbps"));

/////Fragmentation and RTS/CTS properties
  Config::SetDefault ("ns3::WifiRemoteStationManager::FragmentationThreshold", StringValue
("2200"));
  Config::SetDefault ("ns3::WifiRemoteStationManager::RtsCtsThreshold", StringValue
("220"));


/////Set it to adhoc mode
  WifiMacHelper wifiMac;


/////Add the sensing of traffic
  wifiMac.SetType ("ns3::AdhocWifiMac");
  NetDeviceContainer devices = wifi.Install (wifiPhy, wifiMac, nodes);
```

Cause **-** I have modified the value of RTS/CTS value from 2200 to 220. This is done since if the packet value exceeds RTS/CTS value, then the RTS/CTS handshake initiates; the packet size was kept to be the same as 1000 bytes.


Ubuntu-SVS-286-16.06 [Running]
shivang@shivang-VirtualBox: ~/source/ns-3.26
En
12:24 AM
Left ⌘

```
bakefile.xml   Desktop    examples.desktop  Public    Videos
bakeSetEnv.sh  Documents  Music             source
shivang@shivang-VirtualBox:~$ cd source/
shivang@shivang-VirtualBox:~/source$ ls
castxml     netanim-3.107  pygccxml     python-dev          setuptools
clang-dev   ns-3.26        pygoocanvas  pyviz-prerequisites
g++         pybindgen      pygraphviz   qt4
shivang@shivang-VirtualBox:~/source$ cd ns-3.26/
shivang@shivang-VirtualBox:~/source/ns-3.26$ ls
AUTHORS       LICENSE             README         utils      waf-tools
bindings      Makefile            RELEASE_NOTES  utils.py   wscript
build         objdir              scratch        utils.pyc  wutils.py
CHANGES.html  project1a-0-0.pcap  src            VERSION    wutils.pyc
doc           project1a-1-0.pcap  test.py        waf
examples      project1a-2-0.pcap  testpy.supp    waf.bat
shivang@shivang-VirtualBox:~/source/ns-3.26$ ./waf --run "project1b --n=
3"
Waf: Entering directory `/home/shivang/source/ns-3.26/build'
[ 946/2485] Compiling scratch/project1b.cc
[2472/2485] Linking build/scratch/project1b
Waf: Leaving directory `/home/shivang/source/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (8.587s)
Total Packets Received: 2544
shivang@shivang-VirtualBox:~/source/ns-3.26$ ls
AUTHORS       objdir              RELEASE_NOTES  VERSION
bindings      project1a-0-0.pcap  scratch        waf
build         project1a-1-0.pcap  src            waf.bat
CHANGES.html  project1a-2-0.pcap  test.py        waf-tools
doc           project1b-0-0.pcap  testpy.supp    wscript
examples      project1b-1-0.pcap  utils          wutils.py
LICENSE       project1b-2-0.pcap  utils.py       wutils.pyc
Makefile      README              utils.pyc
shivang@shivang-VirtualBox:~/source/ns-3.26$ █
```

Ans.2. The total number of packets received in this network this time are 2544. In Part 3, the number of packets which I received were 1340. The number of packets have increased as we have enabled the RTS/CTS values in the network.

Ans.3. When RTS/CTS are enabled, the number of packets have increased, because it reduces the frame collisions ***among hidden stations***. Once we enable the RTS/CTS, it stops from sending a data frame until the station completes a RTS/CTS handshaking with the other station.

Also, this generally occurs because the system sends these packets with their overheads to cover up for the collision and interference.

We can understand this by a simple example; that if any station A sends a packet without noticing that another station B is already sending the packet, a collision will happen. But if RTS/CTS are enabled, collision won't happen. Thus, the usage of RTS/CTS increases the performance of the system and reduces the collisions, if hidden stations are there. After all, the main goal of RTS/CTS is to improve the performance.

Ans.4. The throughput of the network is –

Throughput = Total Size/Simulation Time
= (Packet Size * Total number of packets)/Simulation Time
= (1064*8 * 2544) / (14.994346) * 10^{-6}
= 1.44417 Mbps

The throughput in Part-2 was 1.68225; while the throughput in this part is 1.44417, which is a decrement of 0.23808.

Ans.5. There is a direct effect of enabling RTS/CTS on Throughput value. If we consider that we do not have a hidden station, then all the usage of RTS/CTS increases the amount of overhead, which reduces the value of Throughput. So, by this fact we can see that it is obvious that ***RTS/CTS*** enabling is ***inversely proportional to Throughput***; which we can see in our calculation of Throughput.

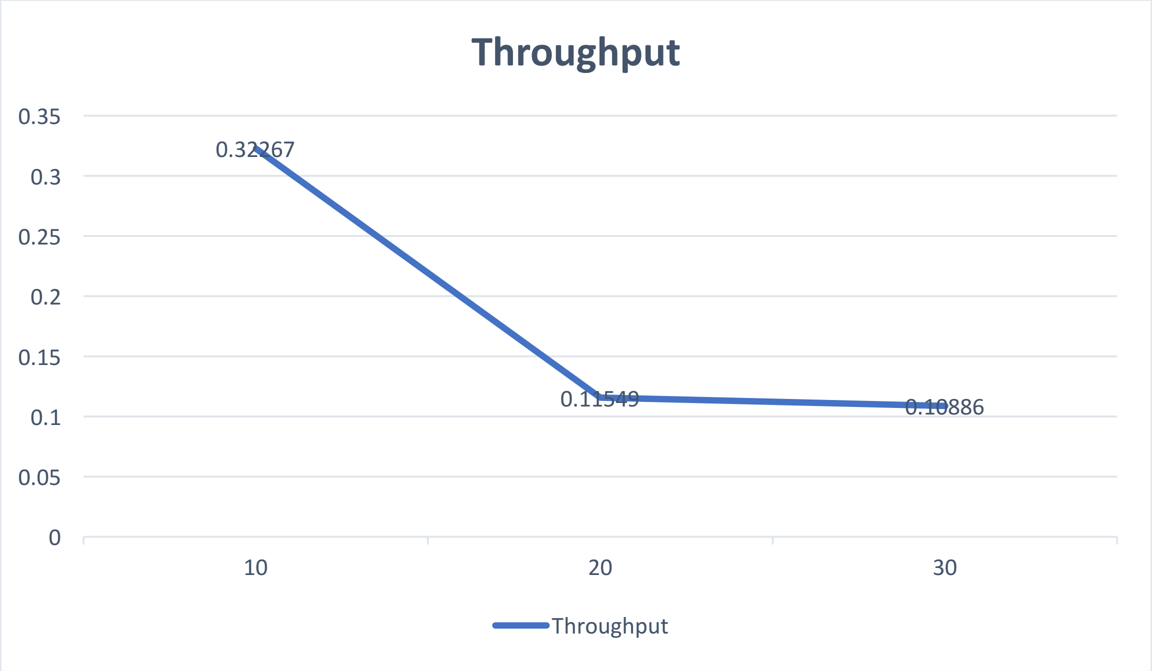
Part-5 –Throughput as a Function of Number of Users

Ans.1. When RTS/CTS is disabled –

When RTS/CTS is disabled, then the overall throughput is calculated and are also plotted in the following graphs (with #of users on X-axis and Throughput value on the Y-axis) –

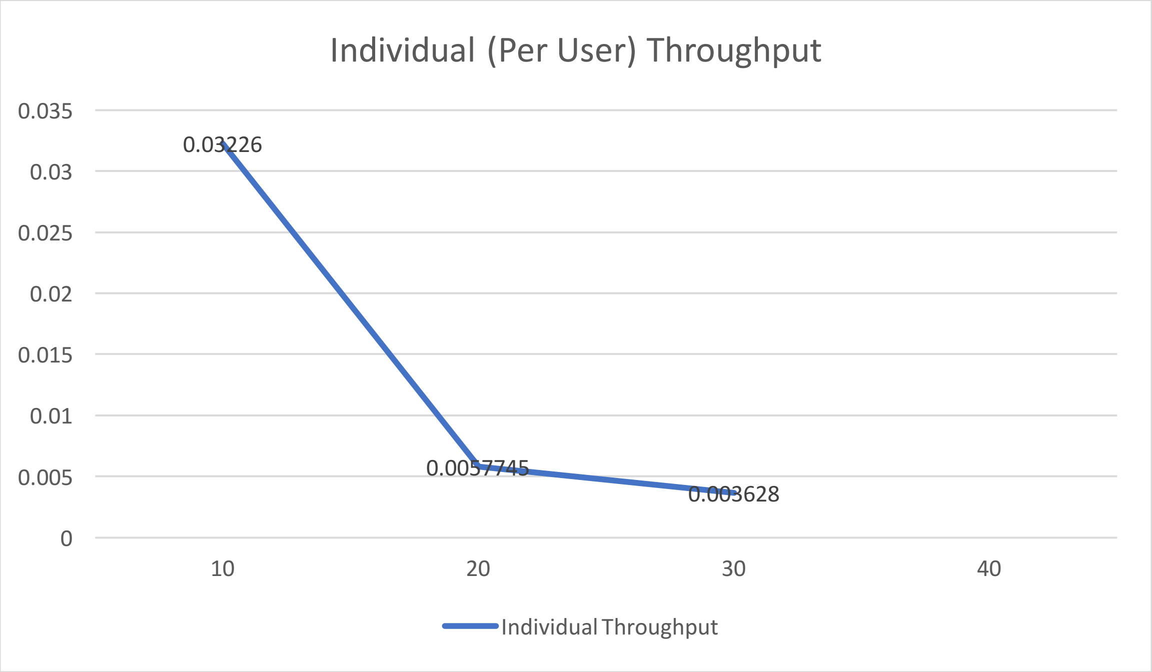
Overall Throughput, when:

# of nodes	Throughput calculation	Throughput (in Mbps)
n=10	$568*1064*8/14.98343*10^{-6}$	= 0.32267
n=20	$201*1064*8/14.8131*10^{-6}$	= 0.11549
n=30	$189*1064*8/14.777559*10^{-6}$	= 0.10886



Per User Throughput –

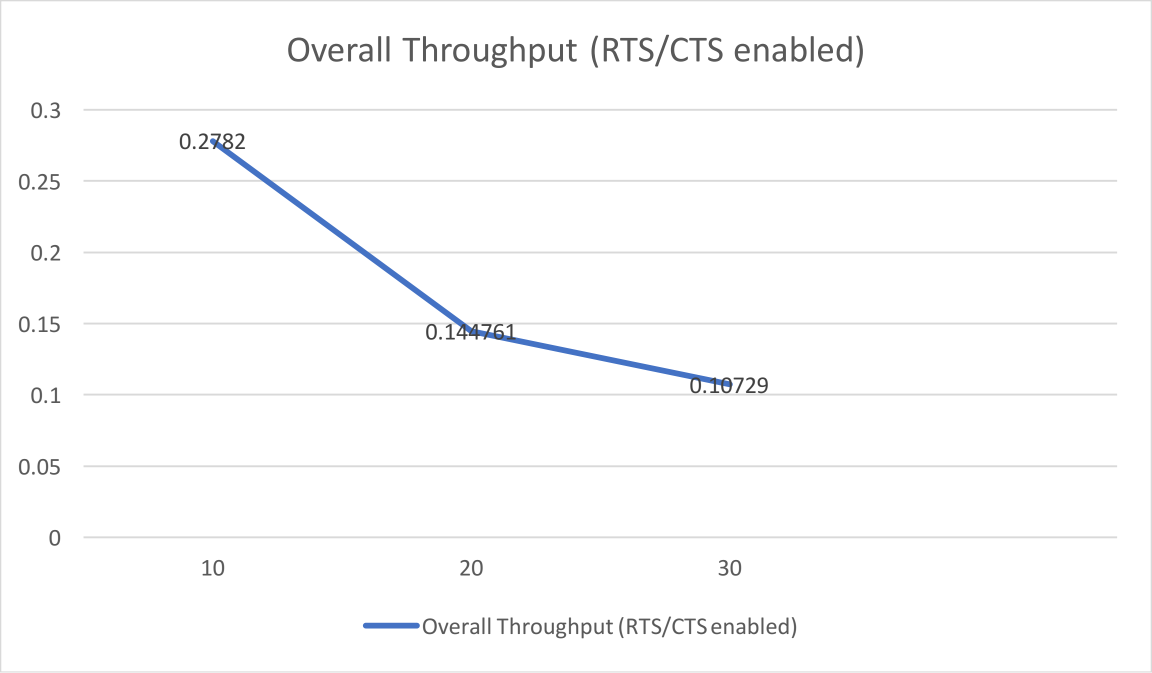
# of nodes	Throughput calculation	Throughput (in Mbps)
n=10	$(568*1064*8/14.98343*10^{-6})/10$ = 0.32267/10	= 0.032267
n=20	$(201*1064*8/14.8131*10^{-6})/20$ = 0.11549/20	= 0.0057745
n=30	$(189*1064*8/14.777559*10^{-6})/30$ = 0.10886	= 0.003628



Ans.2. When RTS/CTS is enabled –

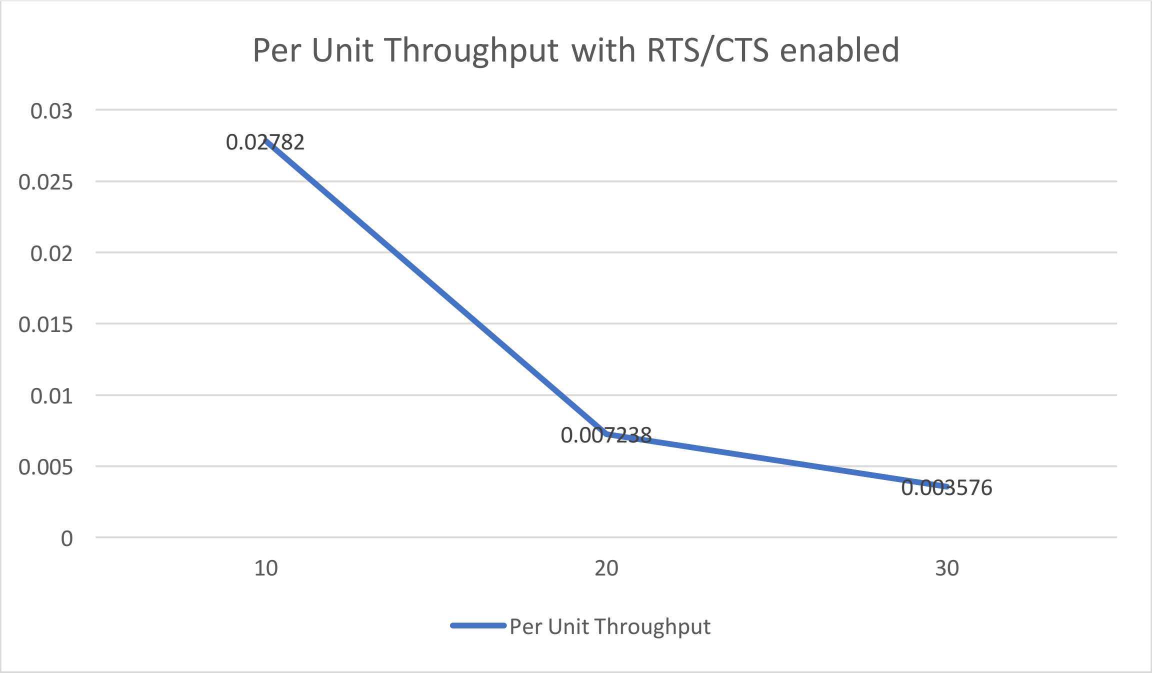
Overall Throughput, when:

# of nodes	Throughput calculation	Throughput (in Mbps)
n=10	490*1064*8/14.992285*10^{-6}	= 0.27820
n=20	255*1064*8/14.994012*10^{-6}	= 0.144761
n=30	189*1064*8/14.993462*10^{-6}	= 0.10729



Per User Throughput –

# of nodes	Throughput calculation	Throughput (in Mbps)
n=10	$(490*1064*8/14.992285*10^{-6})/10$ = 0.27820/10	= 0.02782
n=20	$(255*1064*8/14.994012*10^{-6})/20$ = 0.144761/20	= 0.0072380
n=30	$(189*1064*8/14.993462*10^{-6})/30$ = 0.10729/30	= 0.003576


Per Unit Throughput with RTS/CTS enabled
Per Unit Throughput