North Western University

Introduction to Computer Network Using NS2 based on the domain as www.telegram.com

Sessional Final Report

Submitted to

Md. Shymon Islam

Lecturer

Department of CSE

North Western University

Submitted by:

Armin Jahan Neha

Student Id: 20201137010

Najnine Afroz Estina

Student Id: 20201144010

Rifa Tamanna

Student Id: 20201145010

Department of CSE

North Western University

Course Title: Computer Networks
Sessional Course Code: CSE-3304

Submission Date: 22/12/2022

Table of Contents

1. Overview of the project

1. Introduction to Zenmap

Zenmap work by sending packets to a user-specified host and analyzing the host's response or lack thereof. It is ideally suited for security audits. It can collect information about a scanned IP Address including number of open ports, last boot time, operating system and services running.

2. Four Different Host to Domain: Telegram

W**e** used four different host to reached www.telegram.com. Here each host generate different IP addresses.

3. Design Network Topology

Zenmap's "Topology" option provides an interactive, animated visualization of hosts connectivity. The topology view is very useful when combined with Nmap's –traceroute option to discover network path to host.

4. Prepare Excel Sheet for Network Diagram

W**e** collected IP addresses of four different network from Zenmap and stored it in a excel sheet level by level.

5. Introduction to Network Simulator 2

A network simulator is a software program that can predict the performance of a computer network or a wireless communication network. Since communication networks have become too complex for traditional analytical methods to provide an accurate understanding of system behavior, network simulators are used.

6. Source Code of NS2 Based on Network Diagram

Source code is used in NS2 for display the network flow.

7. Output Topology of Network Simulator 2

The output of NS2 shows network flow of data packets from four different host to destination.

8. Summarization of the Design Network

The design network shows direction of network flow from host to destination and network flow difference among several operators.

9. Conclusion

This project is based on the Zenmap, Excel sheet and Network Simulator 2 and it show network flow of different networks.

10. References

The source we used for collecting different information, their details are given in this section.

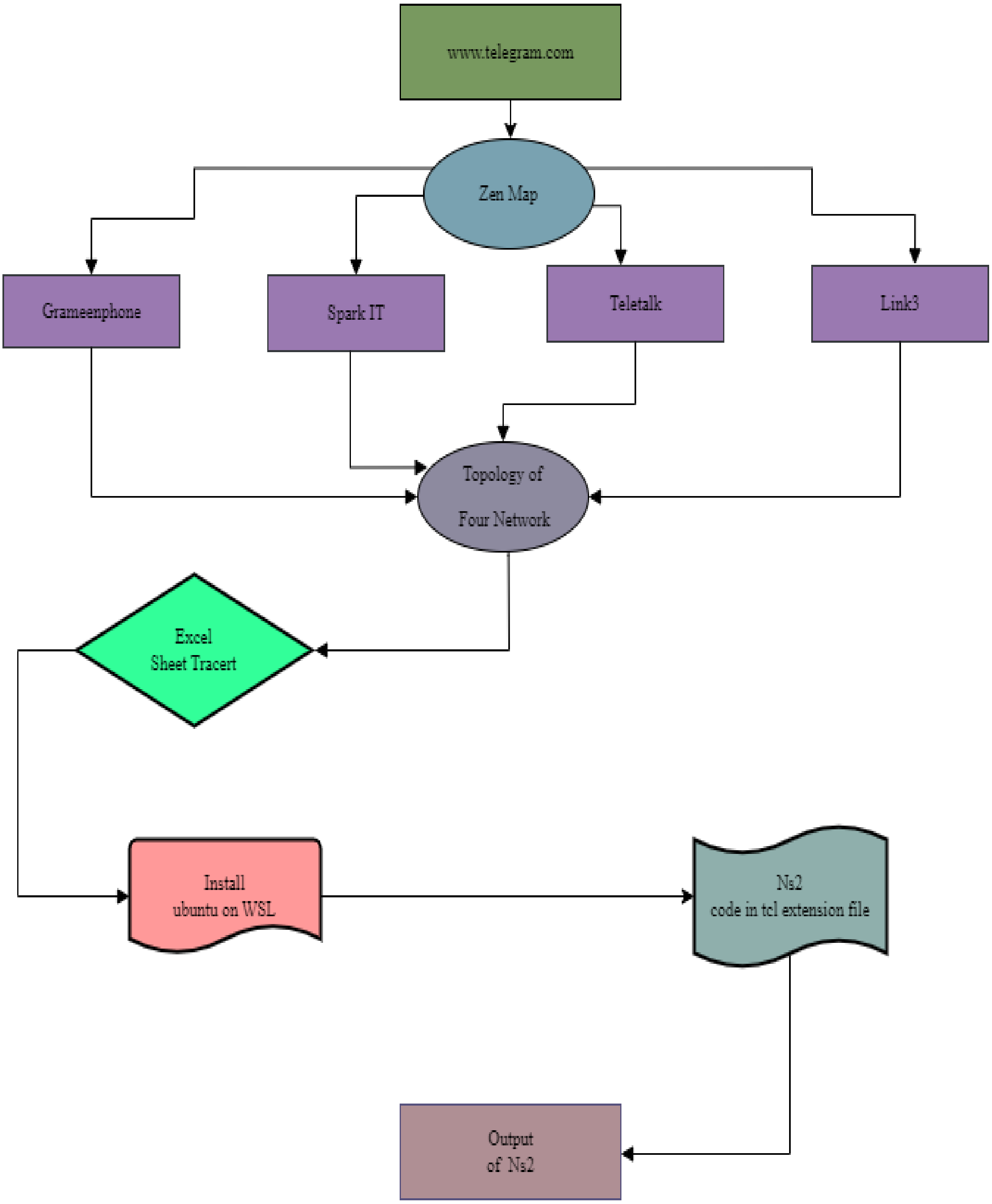


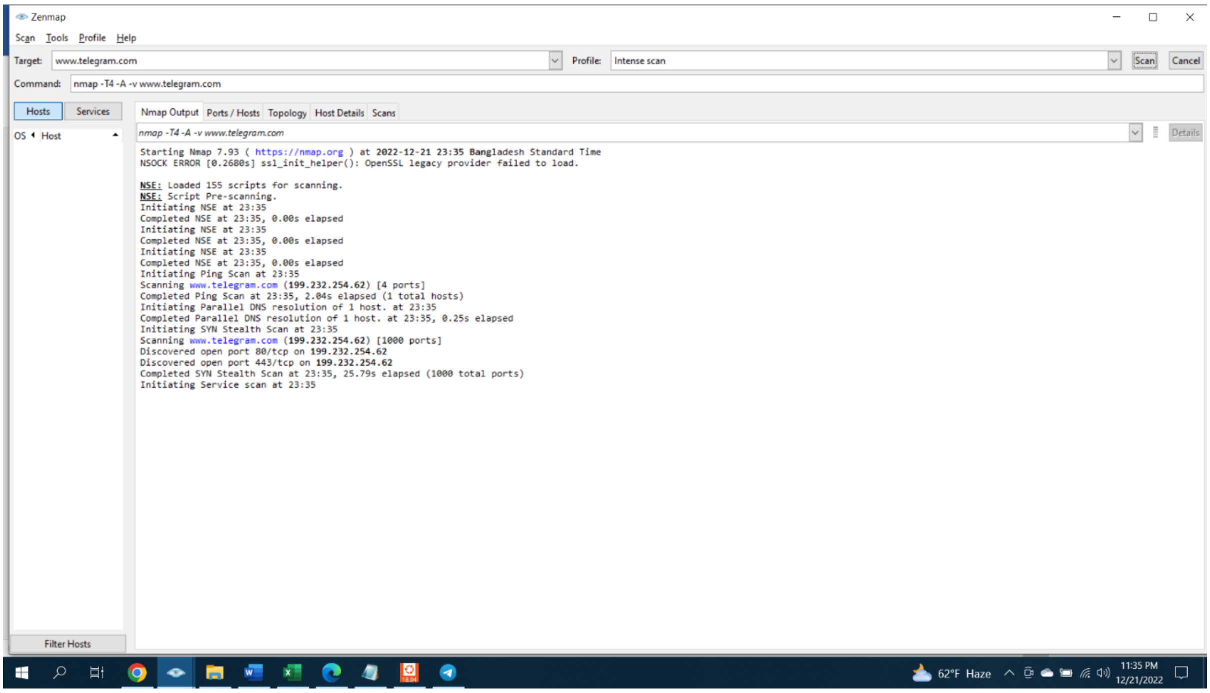
Fig 1: Bird's eye view of the project

2. Introduction to Zenmap

Zenmap is the official graphical user interface (GUI) for the Nmap Security Scanner. It is a multi-platform, free and open-source application designed to make Nmap easy for beginners to use while providing advanced features for experienced Nmap users. Frequently used scans can be saved as profiles to make them easy to run repeatedly. A command creator allows interactive creation of Nmap command lines. Scan results can be saved and viewed later. Saved scans can be compared with one another to see how they differ. The results of recent scans are stored in a searchable database.

Nmap, the acronym for network mapper, is an open-source security auditing and network scanning software. It is developed in such a way that it can quickly analyze massive networks as well as single hosts. Network administrators often use it to detect the devices that are presently working on the system, the port number to which the devices are attached, and the free ports that may be used.

Nmap becomes helpful while performing network penetration testing. It not only gives network information but also assists in discovering security issues in the system. Nmap is platform-agnostic and may be run on various standard operating systems, including Linux, Windows, macOS, and BSD. It is straightforward to use and includes both a command-line interface (CLI) and a graphical user interface (GUI).



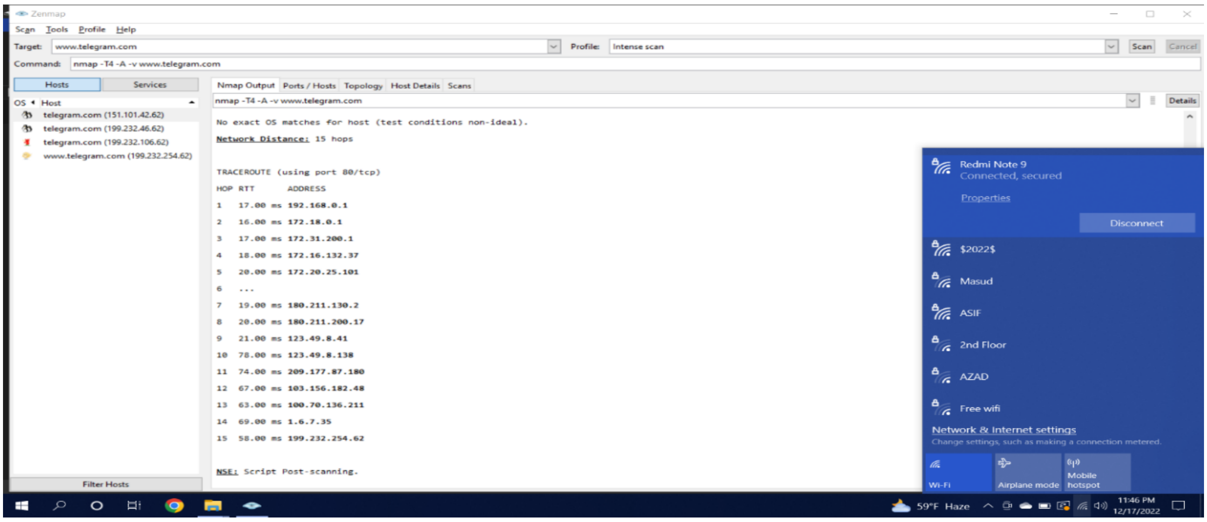
Target: www.telegram.com
Profile: Intense scan
Command: nmap -T4 -A -v www.telegram.com

```
nmap -T4 -A -v www.telegram.com
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-21 23:35 Bangladesh Standard Time
NSOCK ERROR [0.2680s] ssl_init_helper(): OpenSSL legacy provider failed to load.

NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 23:35
Completed NSE at 23:35, 0.00s elapsed
Initiating NSE at 23:35
Completed NSE at 23:35, 0.00s elapsed
Initiating NSE at 23:35
Completed NSE at 23:35, 0.00s elapsed
Initiating Ping Scan at 23:35
Scanning www.telegram.com (199.232.254.62) [4 ports]
Completed Ping Scan at 23:35, 2.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:35
Completed Parallel DNS resolution of 1 host. at 23:35, 0.25s elapsed
Initiating SYN Stealth Scan at 23:35
Scanning www.telegram.com (199.232.254.62) [1000 ports]
Discovered open port 80/tcp on 199.232.254.62
Discovered open port 443/tcp on 199.232.254.62
Completed SYN Stealth Scan at 23:35, 25.79s elapsed (1000 total ports)
Initiating Service scan at 23:35
```

Fig 2: Zen Map interface

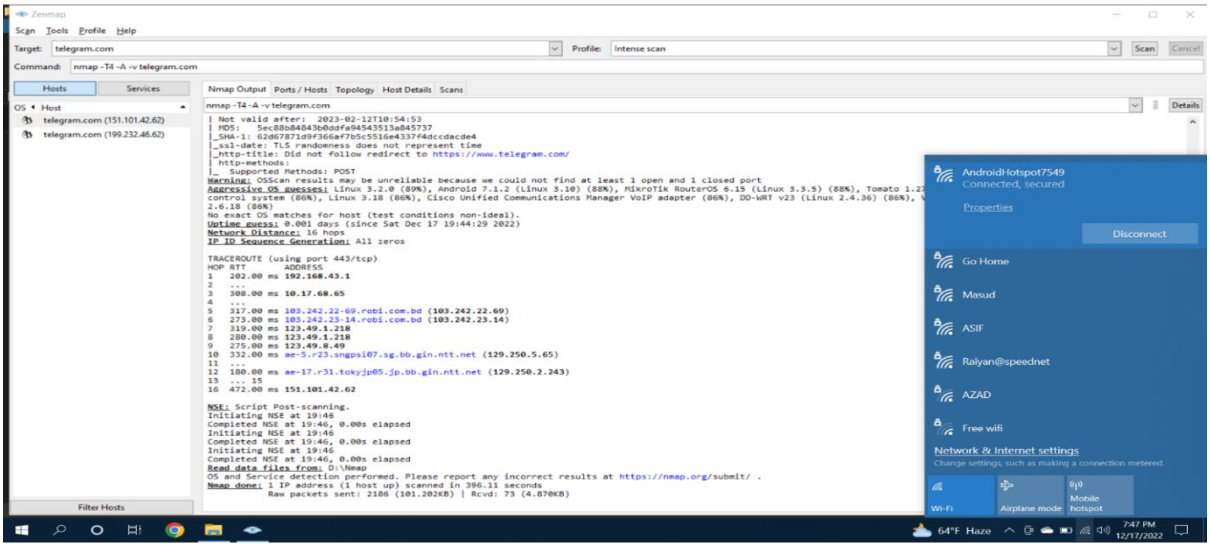
3. Different Host to Domain: Telegram

In this project we have scan www.telegram.com by four different network and find the routing. We scan this website by two cellular network and two are WIFI network. Here we can see the four different network scan instances.



```
nmap -T4 -A -v www.telegram.com

No exact OS matches for host (test conditions non-ideal).
Network Distance: 15 hops

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   17.00 ms 192.168.0.1
2   16.00 ms 172.18.0.1
3   17.00 ms 172.31.200.1
4   18.00 ms 172.16.132.37
5   20.00 ms 172.20.25.101
6   ...
7   19.00 ms 180.211.130.2
8   20.00 ms 180.211.200.17
9   21.00 ms 123.49.8.41
10  78.00 ms 123.49.8.138
11  74.00 ms 209.177.87.180
12  67.00 ms 103.156.182.48
13  63.00 ms 100.70.136.211
14  69.00 ms 1.6.7.35
15  58.00 ms 199.232.254.62

NSE: Script Post-scanning.
```

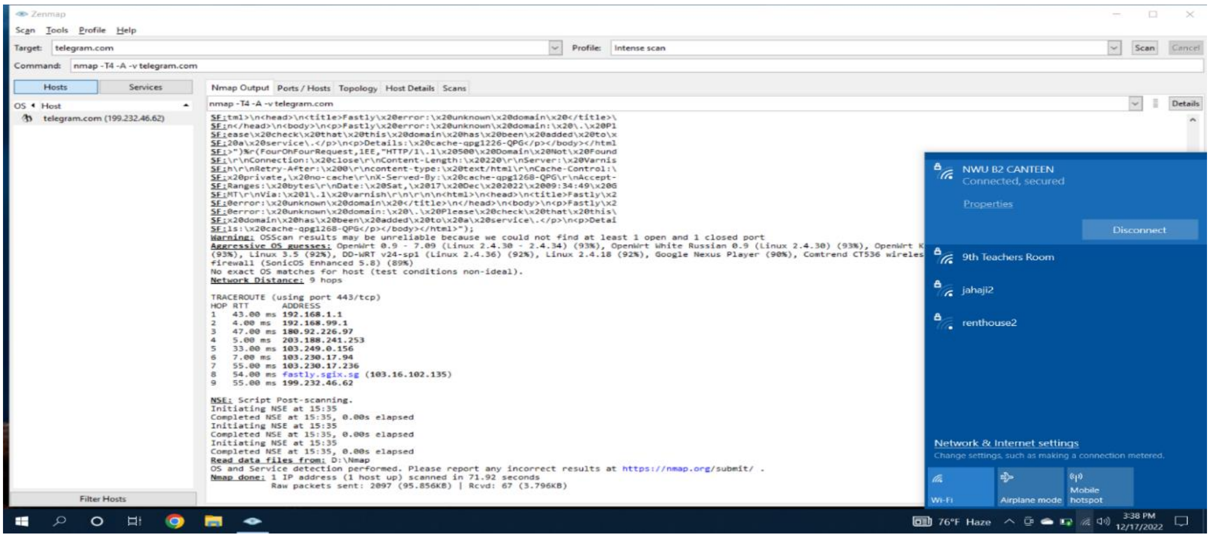
Fig 3: Grameenphone network Zen map scan output



```
nmap -T4 -A -v telegram.com
| Not valid after:  2023-02-12T10:54:53
| MD5:   5ec88b84843b0ddfa94543513a845737
|_SHA-1: 62d67871d9f366af7b5c5516e4337f4dccdacde4
|_ssl-date: TLS randomness does not represent time
|_http-title: Did not follow redirect to https://www.telegram.com/
| http-methods:
|_  Supported Methods: POST
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.2.0 (89%), Android 7.1.2 (Linux 3.10) (88%), MikroTik RouterOS 6.15 (Linux 3.3.5) (88%), Tomato 1.27
control system (86%), Linux 3.18 (86%), Cisco Unified Communications Manager VoIP adapter (86%), DD-WRT v23 (Linux 2.4.36) (86%), V
2.6.18 (86%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 0.001 days (since Sat Dec 17 19:44:29 2022)
Network Distance: 16 hops
IP ID Sequence Generation: All zeros

TRACEROUTE (using port 443/tcp)
HOP RTT       ADDRESS
1   202.00 ms 192.168.43.1
2   ...
3   308.00 ms 10.17.68.65
4   ...
5   317.00 ms 103.242.22-69.robi.com.bd (103.242.22.69)
6   273.00 ms 103.242.23-14.robi.com.bd (103.242.23.14)
7   319.00 ms 123.49.1.218
8   280.00 ms 123.49.1.218
9   275.00 ms 123.49.8.49
10  332.00 ms ae-5.r23.sngpsi07.sg.bb.gin.ntt.net (129.250.5.65)
11  ...
12  180.00 ms ae-17.r31.tokyjp05.jp.bb.gin.ntt.net (129.250.2.243)
13  ... 15
16  472.00 ms 151.101.42.62

NSE: Script Post-scanning.
Initiating NSE at 19:46
Completed NSE at 19:46, 0.00s elapsed
Initiating NSE at 19:46
Completed NSE at 19:46, 0.00s elapsed
Initiating NSE at 19:46
Completed NSE at 19:46, 0.00s elapsed
Read data files from: D:\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 396.11 seconds
           Raw packets sent: 2186 (101.202KB) | Rcvd: 73 (4.870KB)
```

Fig 4: Teletalk Zen map scan output

Here **Fig 4** is the Teletalk output. In this scan output we can see that there are 8 different IP address from host to domain.



Target: telegram.com
Profile: Intense scan
Command: nmap -T4 -A -v telegram.com

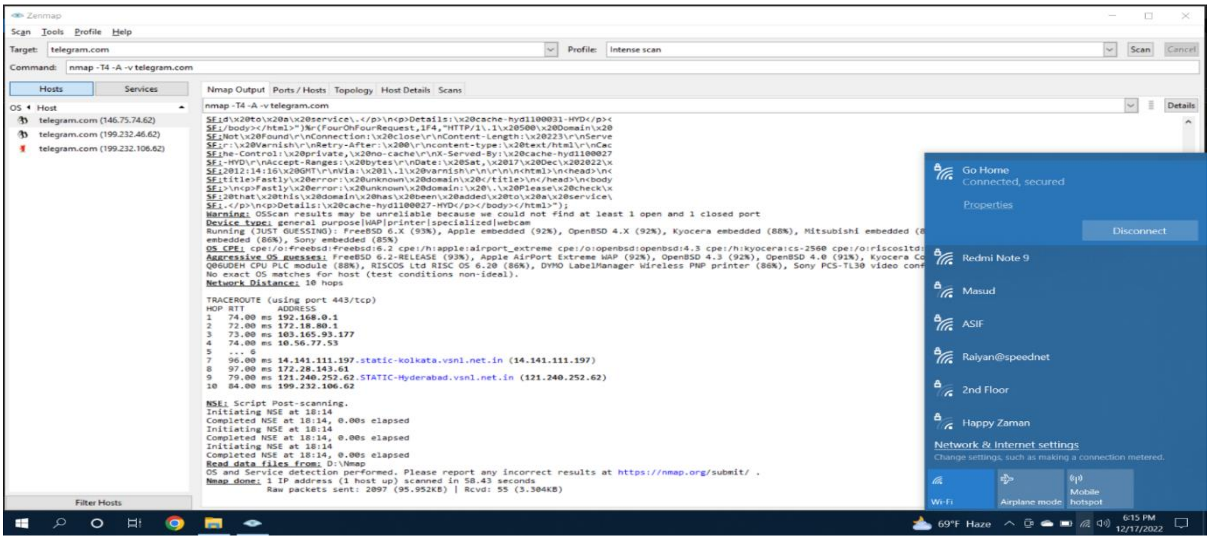
OS ◂ Host
telegram.com (199.232.46.62)

nmap -T4 -A -v telegram.com

```
SF:tml>\n<head>\n<title>Fastly\x20error:\x20unknown\x20domain\x20</title>\
SF:n</head>\n<body>\n<p>Fastly\x20error:\x20unknown\x20domain:\x20\.\x20Pl
SF:ease\x20check\x20that\x20this\x20domain\x20has\x20been\x20added\x20to\x
SF:20a\x20service\.</p>\n<p>Details:\x20cache-qpg1226-QPG</p></body></html
SF:>")%r(FourOhFourRequest,1EE,"HTTP/1\.1\x20500\x20Domain\x20Not\x20Found
SF:\r\nConnection:\x20close\r\nContent-Length:\x20220\r\nServer:\x20Varnis
SF:h\r\nRetry-After:\x200\r\ncontent-type:\x20text/html\r\nCache-Control:\
SF:x20private,\x20no-cache\r\nX-Served-By:\x20cache-qpg1268-QPG\r\nAccept-
SF:Ranges:\x20bytes\r\nDate:\x20Sat,\x2017\x20Dec\x202022\x2009:34:49\x20G
SF:MT\r\nVia:\x201\.1\x20varnish\r\n\r\n\n<html>\n<head>\n<title>Fastly\x2
SF:0error:\x20unknown\x20domain\x20</title>\n</head>\n<body>\n<p>Fastly\x2
SF:0error:\x20unknown\x20domain:\x20\.\x20Please\x20check\x20that\x20this\
SF:x20domain\x20has\x20been\x20added\x20to\x20a\x20service\.</p>\n<p>Detai
SF:ls:\x20cache-qpg1268-QPG</p></body></html>");
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: OpenWrt 0.9 - 7.09 (Linux 2.4.30 - 2.4.34) (93%), OpenWrt White Russian 0.9 (Linux 2.4.30) (93%), OpenWrt K
(93%), Linux 3.5 (92%), DD-WRT v24-sp1 (Linux 2.4.36) (92%), Linux 2.4.18 (92%), Google Nexus Player (90%), Comtrend CT536 wireles
firewall (SonicOS Enhanced 5.8) (89%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 9 hops

TRACEROUTE (using port 443/tcp)
HOP RTT      ADDRESS
1   43.00 ms 192.168.1.1
2   4.00 ms  192.168.99.1
3   47.00 ms 180.92.226.97
4   5.00 ms  203.188.241.253
5   33.00 ms 103.249.0.156
6   7.00 ms  103.230.17.94
7   55.00 ms 103.230.17.236
8   54.00 ms fastly.sgix.sg (103.16.102.135)
9   55.00 ms 199.232.46.62

NSE: Script Post-scanning.
Initiating NSE at 15:35
Completed NSE at 15:35, 0.00s elapsed
Initiating NSE at 15:35
Completed NSE at 15:35, 0.00s elapsed
Initiating NSE at 15:35
Completed NSE at 15:35, 0.00s elapsed
Read data files from: D:\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 71.92 seconds
           Raw packets sent: 2097 (95.856KB) | Rcvd: 67 (3.796KB)
```

Fig 5: SparkIT ISP network Zen map scan output

Here Fig 5 is the SparkIT ISP cellular network output. In this scan output we can see that there are 11 different IP address from host to domain.



Target: telegram.com
Profile: Intense scan
Command: nmap -T4 -A -v telegram.com

OS ◂ Host
telegram.com (146.75.74.62)
telegram.com (199.232.46.62)
telegram.com (199.232.106.62)

nmap -T4 -A -v telegram.com

```
SF:d\x20to\x20a\x20service\.</p>\n<p>Details:\x20cache-hyd1100031-HYD</p><
SF:/body></html>")%r(FourOhFourRequest,1F4,"HTTP/1\.1\x20500\x20Domain\x20
SF:Not\x20Found\r\nConnection:\x20close\r\nContent-Length:\x20223\r\nServe
SF:r:\x20Varnish\r\nRetry-After:\x200\r\ncontent-type:\x20text/html\r\nCac
SF:he-Control:\x20private,\x20no-cache\r\nX-Served-By:\x20cache-hyd1100027
SF:-HYD\r\nAccept-Ranges:\x20bytes\r\nDate:\x20Sat,\x2017\x20Dec\x202022\x
SF:2012:14:16\x20GMT\r\nVia:\x201\.1\x20varnish\r\n\r\n\n<html>\n<head>\n<
SF:title>Fastly\x20error:\x20unknown\x20domain\x20</title>\n</head>\n<body
SF:>\n<p>Fastly\x20error:\x20unknown\x20domain:\x20\.\x20Please\x20check\x
SF:20that\x20this\x20domain\x20has\x20been\x20added\x20to\x20a\x20service\
SF:.</p>\n<p>Details:\x20cache-hyd1100027-HYD</p></body></html>");
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|WAP|printer|specialized|webcam
Running (JUST GUESSING): FreeBSD 6.X (93%), Apple embedded (92%), OpenBSD 4.X (92%), Kyocera embedded (88%), Mitsubishi embedded (8
embedded (86%), Sony embedded (85%)
OS CPE: cpe:/o:freebsd:freebsd:6.2 cpe:/h:apple:airport_extreme cpe:/o:openbsd:openbsd:4.3 cpe:/h:kyocera:cs-2560 cpe:/o:riscosltd:
Aggressive OS guesses: FreeBSD 6.2-RELEASE (93%), Apple AirPort Extreme WAP (92%), OpenBSD 4.3 (92%), OpenBSD 4.0 (91%), Kyocera Co
Q06UDEH CPU PLC module (88%), RISCOS Ltd RISC OS 6.20 (86%), DYMO LabelManager Wireless PNP printer (86%), Sony PCS-TL30 video conf
No exact OS matches for host (test conditions non-ideal).
Network Distance: 10 hops

TRACEROUTE (using port 443/tcp)
HOP RTT      ADDRESS
1   74.00 ms 192.168.0.1
2   72.00 ms 172.18.80.1
3   73.00 ms 103.165.93.177
4   74.00 ms 10.56.77.53
5   ... 6
7   96.00 ms 14.141.111.197.static-kolkata.vsnl.net.in (14.141.111.197)
8   97.00 ms 172.28.143.61
9   79.00 ms 121.240.252.62.STATIC-Hyderabad.vsnl.net.in (121.240.252.62)
10  84.00 ms 199.232.106.62

NSE: Script Post-scanning.
Initiating NSE at 18:14
Completed NSE at 18:14, 0.00s elapsed
Initiating NSE at 18:14
Completed NSE at 18:14, 0.00s elapsed
Initiating NSE at 18:14
Completed NSE at 18:14, 0.00s elapsed
Read data files from: D:\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 58.43 seconds
           Raw packets sent: 2097 (95.952KB) | Rcvd: 55 (3.304KB)
```

Fig 6: Link3 ISP Zen map scan

Here **Fig 6** is the Link3 ISP output. In this scan output we can see that there are 8 different IP address from host to domain.

.

4. Design network topology

Zenmap's "Topology" option provides an interactive , animated visualization of hosts connectivity. Hosts are represented as Nodes on interface and we can use controls to zoom in and zoom out.

On clicking a host, it becomes the new center and when a new scan is launched every new host and network path will be added to the topology automatically.

The topology view is very useful when combined with Nmap's –traceroute option to discover network path to host.

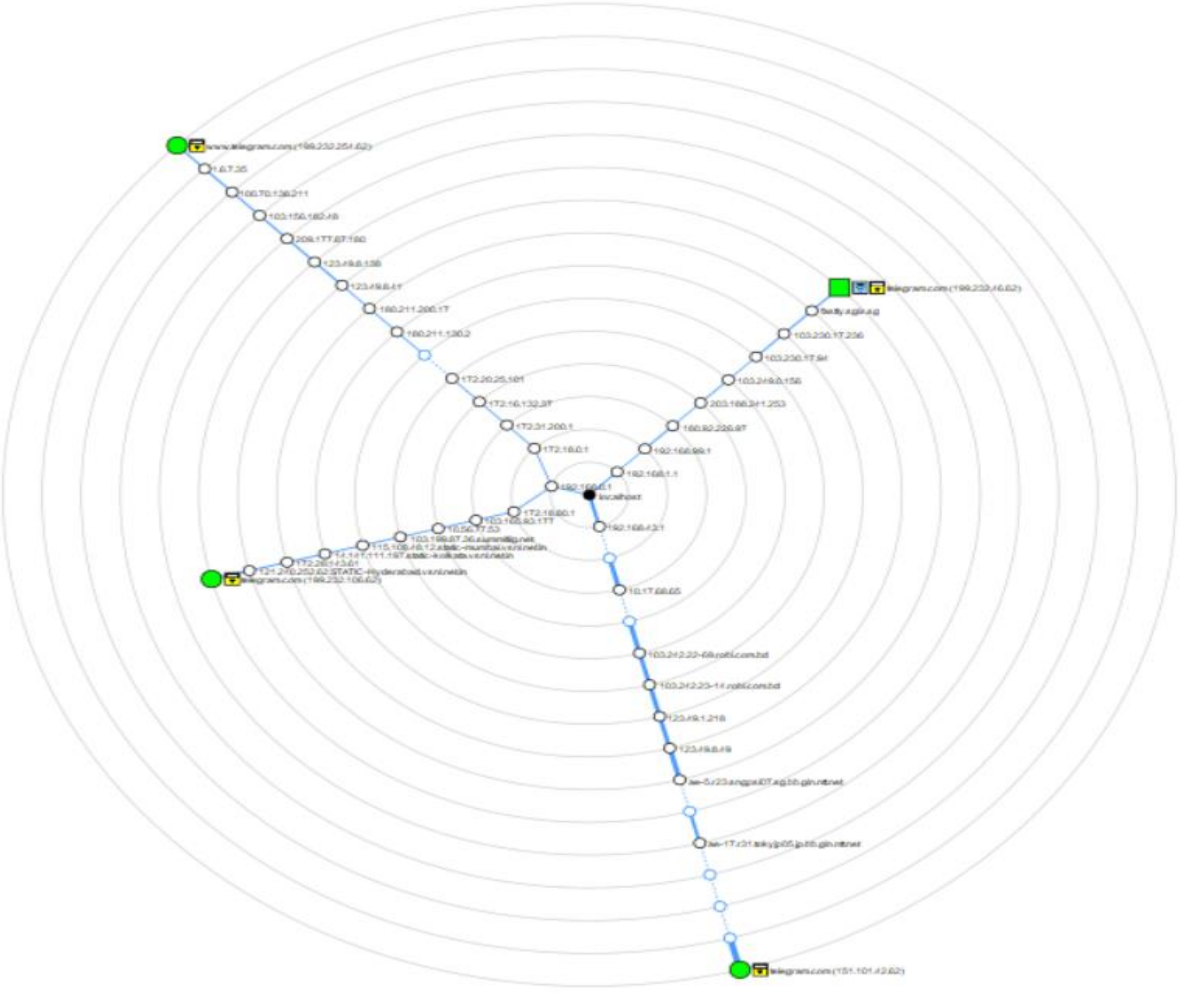
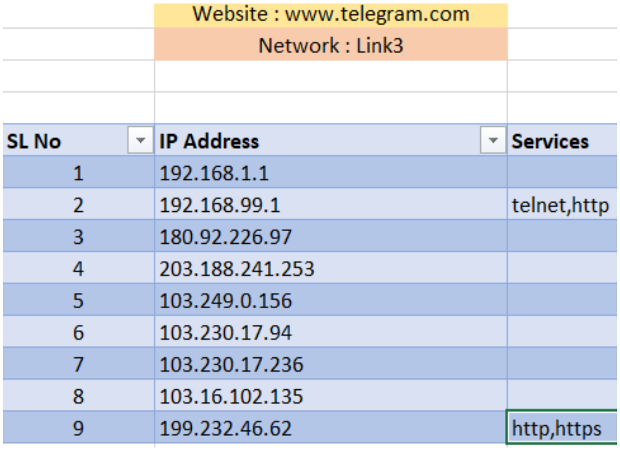

www.telegram.org (149.154.167.99)
1.6.7.35
100.70.136.211
103.156.182.49
209.177.87.180
123.49.8.138
123.49.8.41
180.211.206.17
180.211.130.2
172.20.25.181
172.16.132.37
172.31.200.1
172.18.0.1
192.168.0.1
localhost
172.18.80.1
103.165.93.177
10.56.77.53
103.199.87.36.sumnetg.net
115.108.40.12.static-mumbai.vsnl.net.in
14.141.111.197.static-kolkata.vsnl.net.in
172.28.143.61
121.240.252.82.STATIC-Hyderabad.vsnl.net.in
telegram.com (199.232.106.62)
telegram.com (199.232.46.62)
5edly.agla.g
103.230.17.236
103.230.17.94
103.249.0.156
203.188.241.253
160.92.226.87
192.168.99.1
192.168.1.1
192.168.43.1
10.17.66.65
103.242.22-69.robi.com.bd
103.242.23-14.robi.com.bd
123.49.1.218
123.49.8.49
ae-5.r23.sngpsi07.sg.bb.gin.ntt.net
ae-17.r31.tokyjp05.jp.bb.gin.ntt.net
telegram.com (151.101.42.62)

Fig 7: Network topology of Wikipedia

5. Prepare excel sheet for network diagram

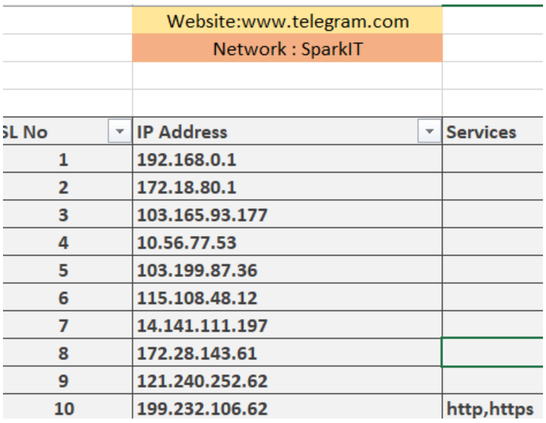
After Creating network topology by Zen map now we have to ready a excel sheet for simulation. First, we need to prepare four service table by serial by serial Ip address. Then we need to put it into a another excel sheet by serial by serial. Bellow there is a sample of excel sheet.



Website : www.telegram.com
Network : Link3

SL No	IP Address	Services
1	192.168.1.1	
2	192.168.99.1	telnet,http
3	180.92.226.97	
4	203.188.241.253	
5	103.249.0.156	
6	103.230.17.94	
7	103.230.17.236	
8	103.16.102.135	
9	199.232.46.62	http,https

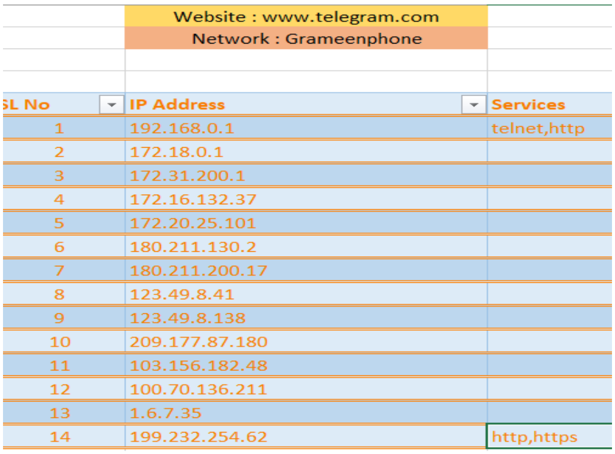
Fig 8: Service table of Link3



Website:www.telegram.com
Network : SparkIT

SL No	IP Address	Services
1	192.168.0.1	
2	172.18.80.1	
3	103.165.93.177	
4	10.56.77.53	
5	103.199.87.36	
6	115.108.48.12	
7	14.141.111.197	
8	172.28.143.61	
9	121.240.252.62	
10	199.232.106.62	http,https

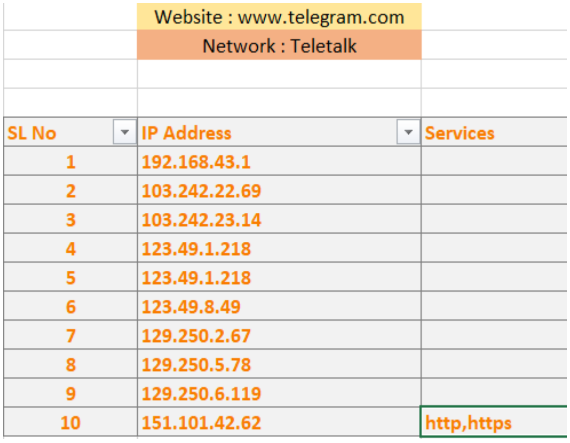
Fig 9: Service table of SparkIT



Website : www.telegram.com
Network : Grameenphone

SL No	IP Address	Services
1	192.168.0.1	telnet,http
2	172.18.0.1	
3	172.31.200.1	
4	172.16.132.37	
5	172.20.25.101	
6	180.211.130.2	
7	180.211.200.17	
8	123.49.8.41	
9	123.49.8.138	
10	209.177.87.180	
11	103.156.182.48	
12	100.70.136.211	
13	1.6.7.35	
14	199.232.254.62	http,https

Fig 10: Service table of Grameenphone



Website : www.telegram.com
Network : Teletalk

SL No	IP Address	Services
1	192.168.43.1	
2	103.242.22.69	
3	103.242.23.14	
4	123.49.1.218	
5	123.49.1.218	
6	123.49.8.49	
7	129.250.2.67	
8	129.250.5.78	
9	129.250.6.119	
10	151.101.42.62	http,https

Fig 8: Service table of Teletalk

These four figures are the serial wise Ip address table. We get this information from Zen map by scanning www.telegram.com website. Four different network goes to target address by many different Ip address. Now we will create a excel tracert for visualize the simulation process.

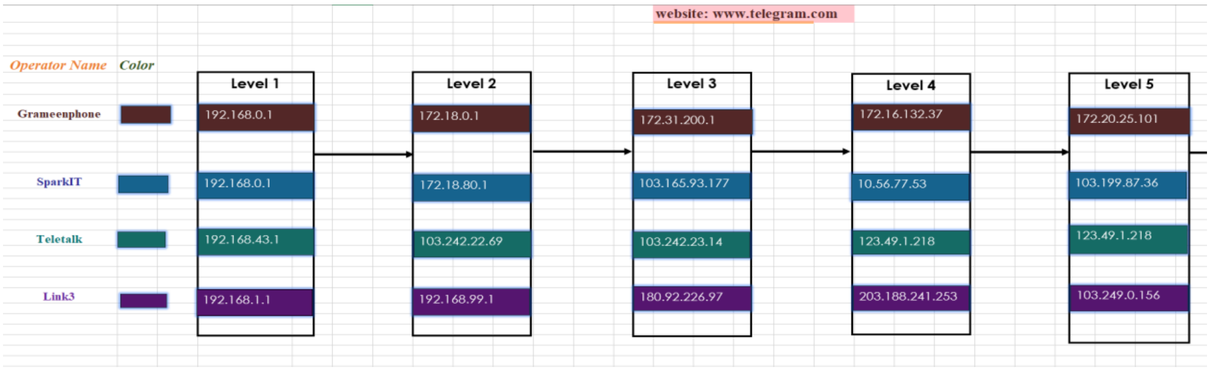


Fig 12: First-half excel sheet tracert

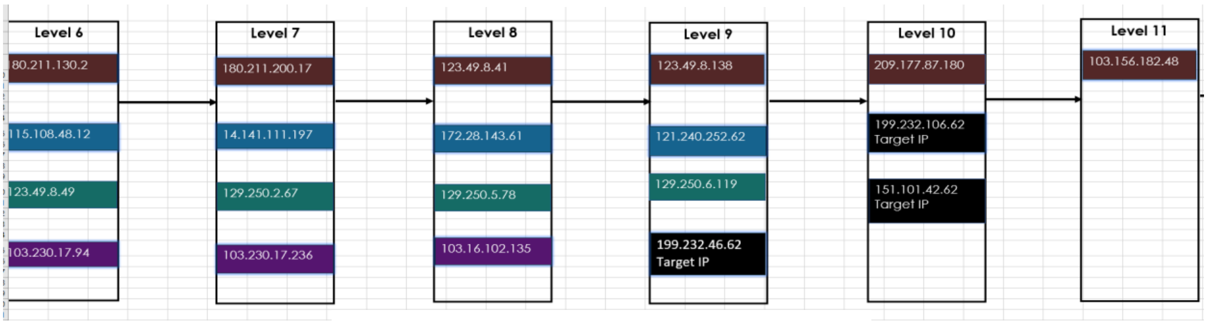


Fig 13: Middle excel sheet tracert

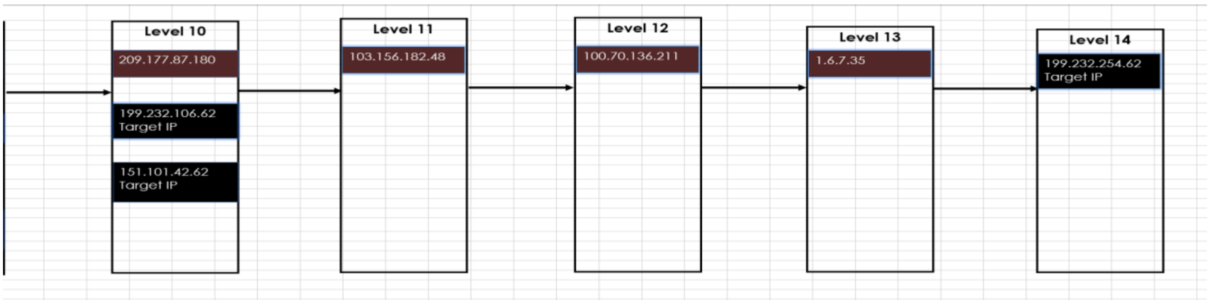


Fig 14: Last-half excel sheet tracert

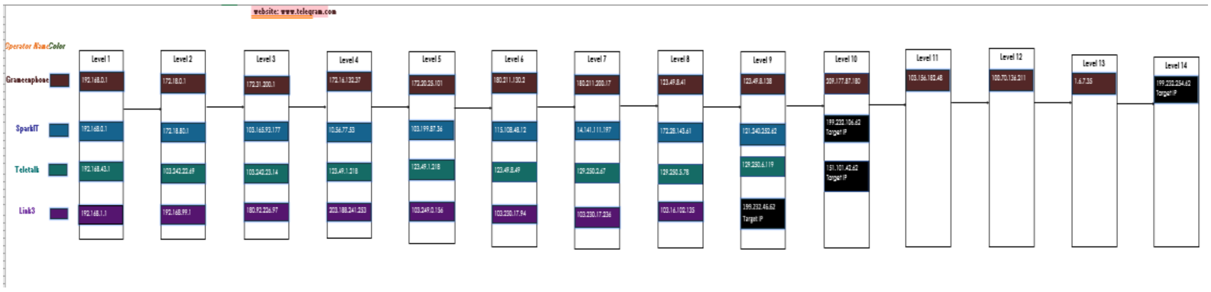


Fig 15: Full excel sheet tracert

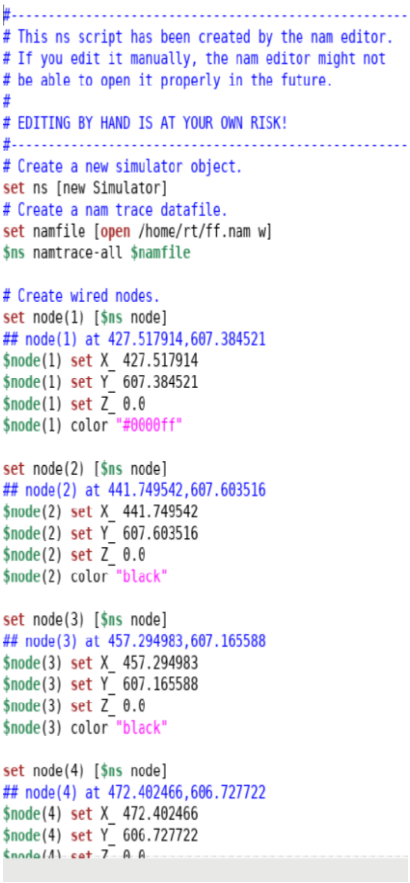
6. Introduction to NS2

Network Simulator 2 is simply a discrete event-driven network simulation tool for studying the dynamic nature of communication networks. Network Simulator 2 provides substantial support for simulation of different protocols over wired and wireless networks. It provides a highly modular platform for wired and wireless simulations supporting different network elements, protocols, traffic, and routing types.

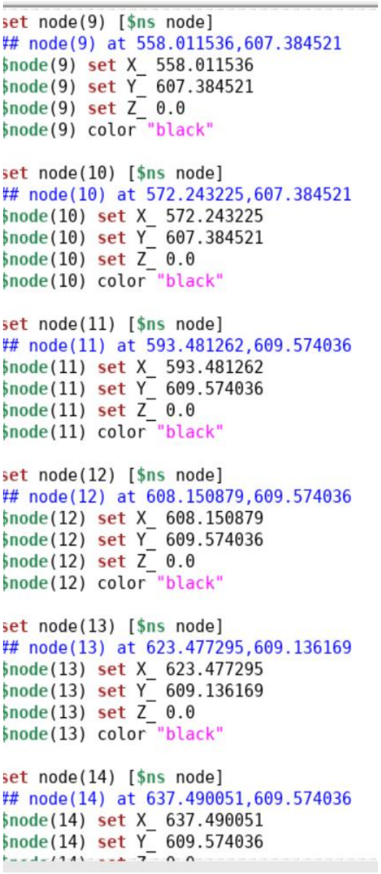
Network simulator 2 is a Linux based software. For running ns2 in windows we need to install ubuntu operating system in windows PowerShell on WSL. After installing ubuntu we need to install nam, gedit, Xlunch and NS2. After installing this software into ubuntu then the ns2 will run.

7. Source code of NS2

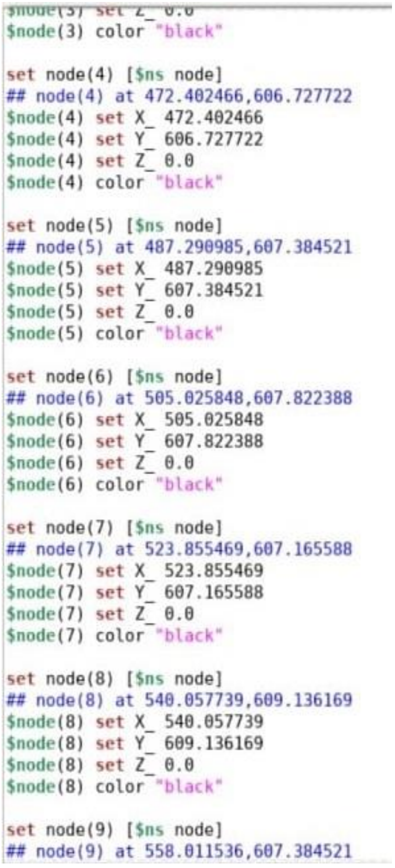
The source code of NS2 are edit and store in TCL extension file.



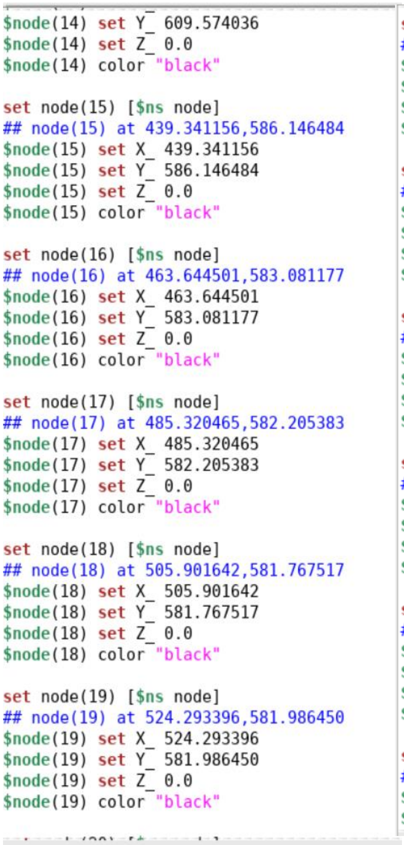
```
#-----------------------------------------------------
# This ns script has been created by the nam editor.
# If you edit it manually, the nam editor might not
# be able to open it properly in the future.
#
# EDITING BY HAND IS AT YOUR OWN RISK!
#-----------------------------------------------------
# Create a new simulator object.
set ns [new Simulator]
# Create a nam trace datafile.
set namfile [open /home/rt/ff.nam w]
$ns namtrace-all $namfile

# Create wired nodes.
set node(1) [$ns node]
## node(1) at 427.517914,607.384521
$node(1) set X_ 427.517914
$node(1) set Y_ 607.384521
$node(1) set Z_ 0.0
$node(1) color "#0000ff"

set node(2) [$ns node]
## node(2) at 441.749542,607.603516
$node(2) set X_ 441.749542
$node(2) set Y_ 607.603516
$node(2) set Z_ 0.0
$node(2) color "black"

set node(3) [$ns node]
## node(3) at 457.294983,607.165588
$node(3) set X_ 457.294983
$node(3) set Y_ 607.165588
$node(3) set Z_ 0.0
$node(3) color "black"

set node(4) [$ns node]
## node(4) at 472.402466,606.727722
$node(4) set X_ 472.402466
$node(4) set Y_ 606.727722
$node(4) set Z_ 0.0
```



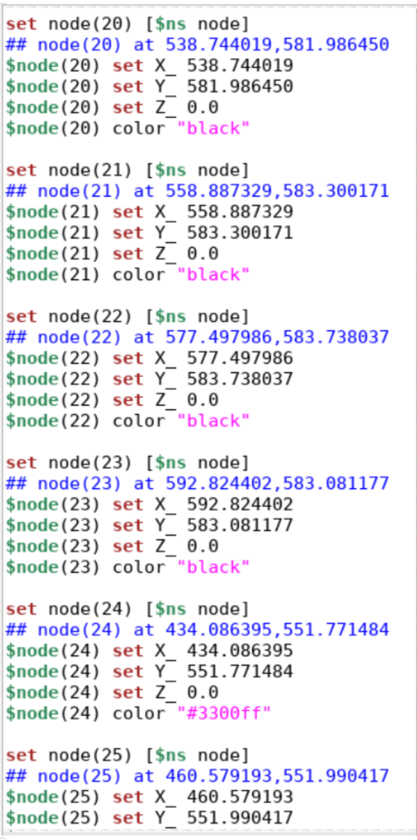
```
set node(9) [$ns node]
## node(9) at 558.011536,607.384521
$node(9) set X_ 558.011536
$node(9) set Y_ 607.384521
$node(9) set Z_ 0.0
$node(9) color "black"

set node(10) [$ns node]
## node(10) at 572.243225,607.384521
$node(10) set X_ 572.243225
$node(10) set Y_ 607.384521
$node(10) set Z_ 0.0
$node(10) color "black"

set node(11) [$ns node]
## node(11) at 593.481262,609.574036
$node(11) set X_ 593.481262
$node(11) set Y_ 609.574036
$node(11) set Z_ 0.0
$node(11) color "black"

set node(12) [$ns node]
## node(12) at 608.150879,609.574036
$node(12) set X_ 608.150879
$node(12) set Y_ 609.574036
$node(12) set Z_ 0.0
$node(12) color "black"

set node(13) [$ns node]
## node(13) at 623.477295,609.136169
$node(13) set X_ 623.477295
$node(13) set Y_ 609.136169
$node(13) set Z_ 0.0
$node(13) color "black"

set node(14) [$ns node]
## node(14) at 637.490051,609.574036
$node(14) set X_ 637.490051
$node(14) set Y_ 609.574036
$node(14) set Z_ 0.0
```



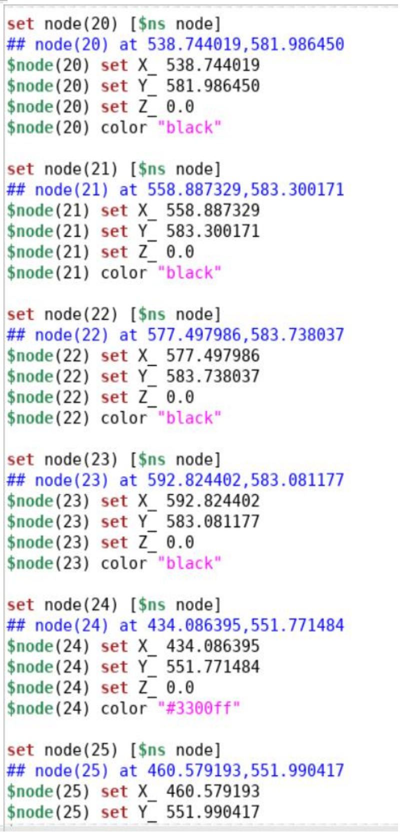
```
$node(3) set Z_ 0.0
$node(3) color "black"

set node(4) [$ns node]
## node(4) at 472.402466,606.727722
$node(4) set X_ 472.402466
$node(4) set Y_ 606.727722
$node(4) set Z_ 0.0
$node(4) color "black"

set node(5) [$ns node]
## node(5) at 487.290985,607.384521
$node(5) set X_ 487.290985
$node(5) set Y_ 607.384521
$node(5) set Z_ 0.0
$node(5) color "black"

set node(6) [$ns node]
## node(6) at 505.025848,607.822388
$node(6) set X_ 505.025848
$node(6) set Y_ 607.822388
$node(6) set Z_ 0.0
$node(6) color "black"

set node(7) [$ns node]
## node(7) at 523.855469,607.165588
$node(7) set X_ 523.855469
$node(7) set Y_ 607.165588
$node(7) set Z_ 0.0
$node(7) color "black"

set node(8) [$ns node]
## node(8) at 540.057739,609.136169
$node(8) set X_ 540.057739
$node(8) set Y_ 609.136169
$node(8) set Z_ 0.0
$node(8) color "black"

set node(9) [$ns node]
## node(9) at 558.011536,607.384521
```



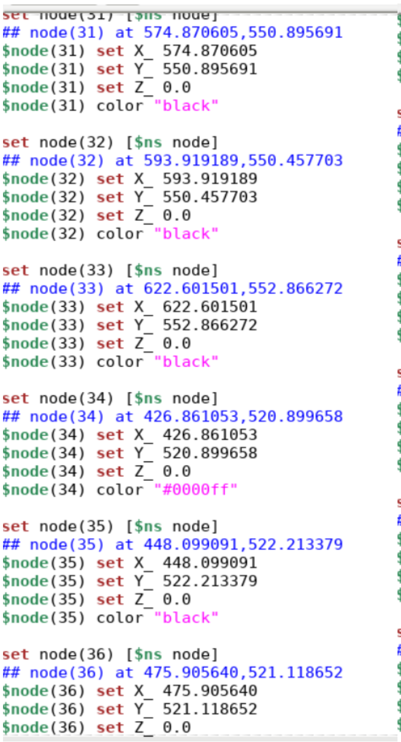
```
$node(14) set Y_ 609.574036
$node(14) set Z_ 0.0
$node(14) color "black"

set node(15) [$ns node]
## node(15) at 439.341156,586.146484
$node(15) set X_ 439.341156
$node(15) set Y_ 586.146484
$node(15) set Z_ 0.0
$node(15) color "black"

set node(16) [$ns node]
## node(16) at 463.644501,583.081177
$node(16) set X_ 463.644501
$node(16) set Y_ 583.081177
$node(16) set Z_ 0.0
$node(16) color "black"

set node(17) [$ns node]
## node(17) at 485.320465,582.205383
$node(17) set X_ 485.320465
$node(17) set Y_ 582.205383
$node(17) set Z_ 0.0
$node(17) color "black"

set node(18) [$ns node]
## node(18) at 505.901642,581.767517
$node(18) set X_ 505.901642
$node(18) set Y_ 581.767517
$node(18) set Z_ 0.0
$node(18) color "black"

set node(19) [$ns node]
## node(19) at 524.293396,581.986450
$node(19) set X_ 524.293396
$node(19) set Y_ 581.986450
$node(19) set Z_ 0.0
$node(19) color "black"

set node(20) [$ns node]
```



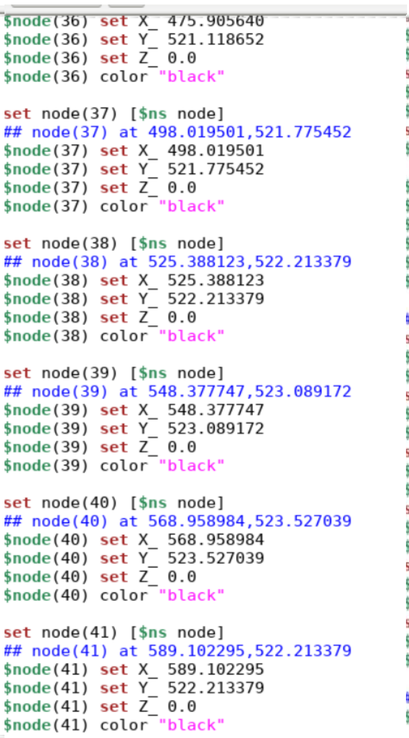
```
set node(20) [$ns node]
## node(20) at 538.744019,581.986450
$node(20) set X_ 538.744019
$node(20) set Y_ 581.986450
$node(20) set Z_ 0.0
$node(20) color "black"

set node(21) [$ns node]
## node(21) at 558.887329,583.300171
$node(21) set X_ 558.887329
$node(21) set Y_ 583.300171
$node(21) set Z_ 0.0
$node(21) color "black"

set node(22) [$ns node]
## node(22) at 577.497986,583.738037
$node(22) set X_ 577.497986
$node(22) set Y_ 583.738037
$node(22) set Z_ 0.0
$node(22) color "black"

set node(23) [$ns node]
## node(23) at 592.824402,583.081177
$node(23) set X_ 592.824402
$node(23) set Y_ 583.081177
$node(23) set Z_ 0.0
$node(23) color "black"

set node(24) [$ns node]
## node(24) at 434.086395,551.771484
$node(24) set X_ 434.086395
$node(24) set Y_ 551.771484
$node(24) set Z_ 0.0
$node(24) color "#3300ff"

set node(25) [$ns node]
## node(25) at 460.579193,551.990417
$node(25) set X_ 460.579193
$node(25) set Y_ 551.990417
```



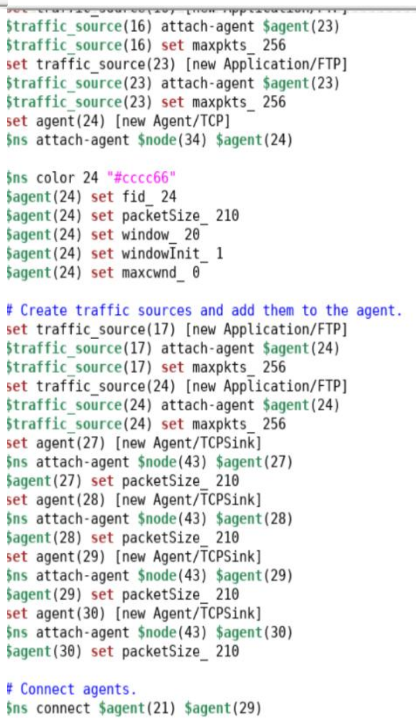
```
set node(20) [$ns node]
## node(20) at 538.744019,581.986450
$node(20) set X_ 538.744019
$node(20) set Y_ 581.986450
$node(20) set Z_ 0.0
$node(20) color "black"

set node(21) [$ns node]
## node(21) at 558.887329,583.300171
$node(21) set X_ 558.887329
$node(21) set Y_ 583.300171
$node(21) set Z_ 0.0
$node(21) color "black"

set node(22) [$ns node]
## node(22) at 577.497986,583.738037
$node(22) set X_ 577.497986
$node(22) set Y_ 583.738037
$node(22) set Z_ 0.0
$node(22) color "black"

set node(23) [$ns node]
## node(23) at 592.824402,583.081177
$node(23) set X_ 592.824402
$node(23) set Y_ 583.081177
$node(23) set Z_ 0.0
$node(23) color "black"

set node(24) [$ns node]
## node(24) at 434.086395,551.771484
$node(24) set X_ 434.086395
$node(24) set Y_ 551.771484
$node(24) set Z_ 0.0
$node(24) color "#3300ff"

set node(25) [$ns node]
## node(25) at 460.579193,551.990417
$node(25) set X_ 460.579193
$node(25) set Y_ 551.990417
```



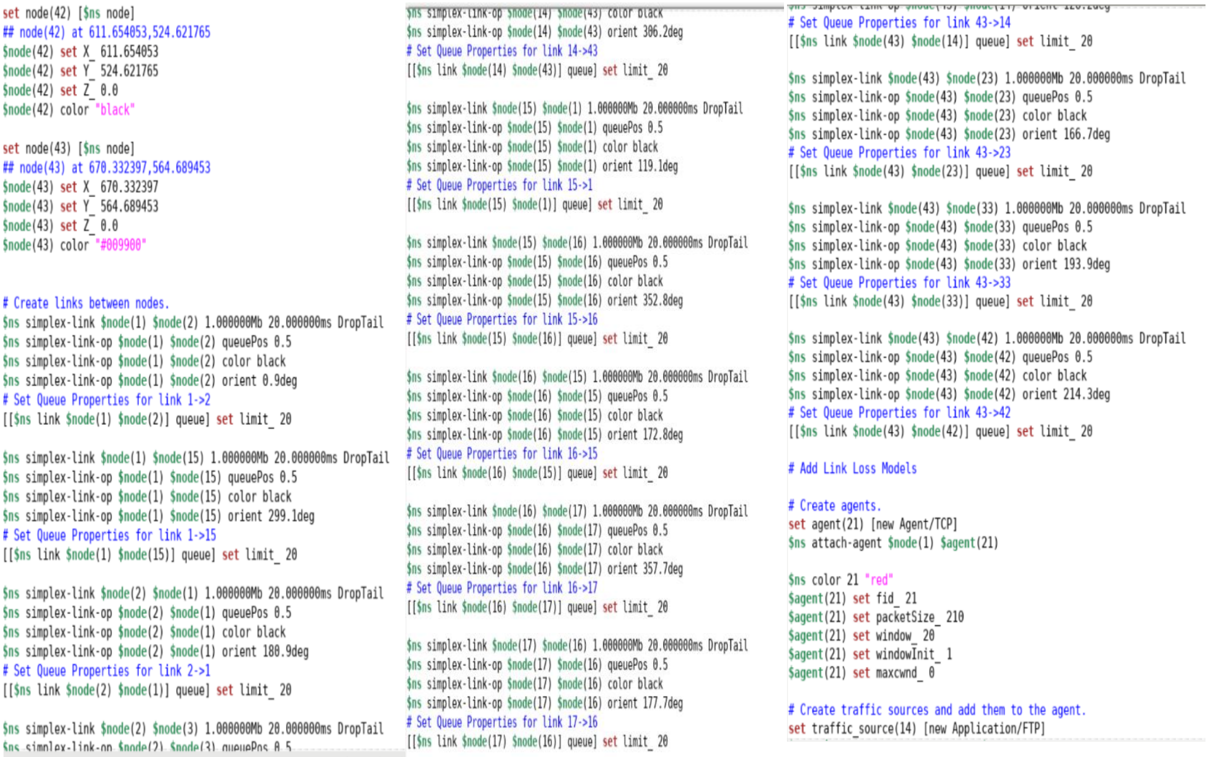
```
set node(31) [$ns node]
## node(31) at 574.870605,550.895691
$node(31) set X_ 574.870605
$node(31) set Y_ 550.895691
$node(31) set Z_ 0.0
$node(31) color "black"

set node(32) [$ns node]
## node(32) at 593.919189,550.457703
$node(32) set X_ 593.919189
$node(32) set Y_ 550.457703
$node(32) set Z_ 0.0
$node(32) color "black"

set node(33) [$ns node]
## node(33) at 622.601501,552.866272
$node(33) set X_ 622.601501
$node(33) set Y_ 552.866272
$node(33) set Z_ 0.0
$node(33) color "black"

set node(34) [$ns node]
## node(34) at 426.861053,520.899658
$node(34) set X_ 426.861053
$node(34) set Y_ 520.899658
$node(34) set Z_ 0.0
$node(34) color "#0000ff"

set node(35) [$ns node]
## node(35) at 448.099091,522.213379
$node(35) set X_ 448.099091
$node(35) set Y_ 522.213379
$node(35) set Z_ 0.0
$node(35) color "black"

set node(36) [$ns node]
## node(36) at 475.905640,521.118652
$node(36) set X_ 475.905640
$node(36) set Y_ 521.118652
$node(36) set Z_ 0.0
```



```
$node(36) set X_ 475.905640
$node(36) set Y_ 521.118652
$node(36) set Z_ 0.0
$node(36) color "black"

set node(37) [$ns node]
## node(37) at 498.019501,521.775452
$node(37) set X_ 498.019501
$node(37) set Y_ 521.775452
$node(37) set Z_ 0.0
$node(37) color "black"

set node(38) [$ns node]
## node(38) at 525.388123,522.213379
$node(38) set X_ 525.388123
$node(38) set Y_ 522.213379
$node(38) set Z_ 0.0
$node(38) color "black"

set node(39) [$ns node]
## node(39) at 548.377747,523.089172
$node(39) set X_ 548.377747
$node(39) set Y_ 523.089172
$node(39) set Z_ 0.0
$node(39) color "black"

set node(40) [$ns node]
## node(40) at 568.958984,523.527039
$node(40) set X_ 568.958984
$node(40) set Y_ 523.527039
$node(40) set Z_ 0.0
$node(40) color "black"

set node(41) [$ns node]
## node(41) at 589.102295,522.213379
$node(41) set X_ 589.102295
$node(41) set Y_ 522.213379
$node(41) set Z_ 0.0
$node(41) color "black"
```



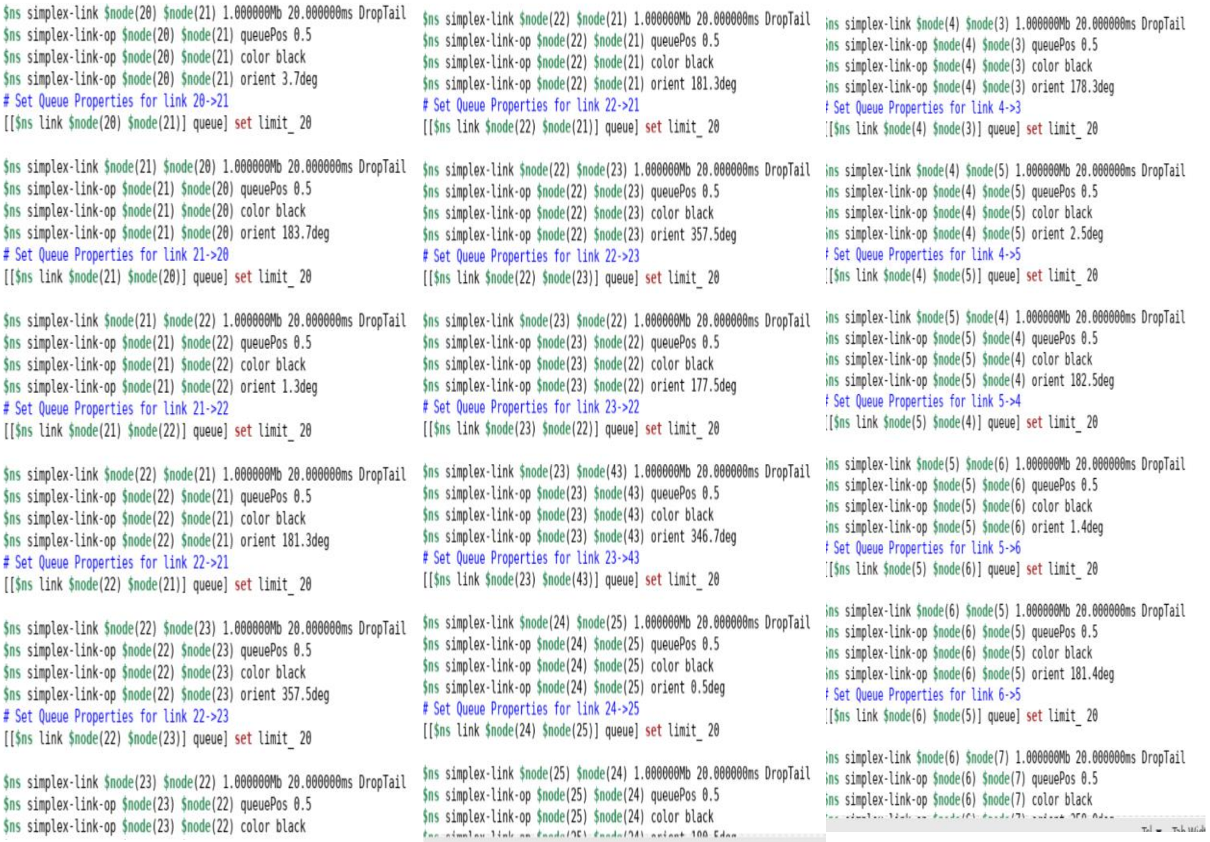
```
set traffic_source(16) [new Application/FTP]
$traffic_source(16) attach-agent $agent(23)
$traffic_source(16) set maxpkts_ 256
set traffic_source(23) [new Application/FTP]
$traffic_source(23) attach-agent $agent(23)
$traffic_source(23) set maxpkts_ 256
set agent(24) [new Agent/TCP]
$ns attach-agent $node(34) $agent(24)

$ns color 24 "#cccc66"
$agent(24) set fid_ 24
$agent(24) set packetSize_ 210
$agent(24) set window_ 20
$agent(24) set windowInit_ 1
$agent(24) set maxcwnd_ 0

# Create traffic sources and add them to the agent.
set traffic_source(17) [new Application/FTP]
$traffic_source(17) attach-agent $agent(24)
$traffic_source(17) set maxpkts_ 256
set traffic_source(24) [new Application/FTP]
$traffic_source(24) attach-agent $agent(24)
$traffic_source(24) set maxpkts_ 256
set agent(27) [new Agent/TCPSink]
$ns attach-agent $node(43) $agent(27)
$agent(27) set packetSize_ 210
set agent(28) [new Agent/TCPSink]
$ns attach-agent $node(43) $agent(28)
$agent(28) set packetSize_ 210
set agent(29) [new Agent/TCPSink]
$ns attach-agent $node(43) $agent(29)
$agent(29) set packetSize_ 210
set agent(30) [new Agent/TCPSink]
$ns attach-agent $node(43) $agent(30)
$agent(30) set packetSize_ 210

# Connect agents.
$ns connect $agent(21) $agent(29)
```



```
set node(42) [$ns node]
## node(42) at 611.654053,524.621765
$node(42) set X_ 611.654053
$node(42) set Y_ 524.621765
$node(42) set Z_ 0.0
$node(42) color "black"

set node(43) [$ns node]
## node(43) at 670.332397,564.689453
$node(43) set X_ 670.332397
$node(43) set Y_ 564.689453
$node(43) set Z_ 0.0
$node(43) color "#009900"


# Create links between nodes.
$ns simplex-link $node(1) $node(2) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(1) $node(2) queuePos 0.5
$ns simplex-link-op $node(1) $node(2) color black
$ns simplex-link-op $node(1) $node(2) orient 0.9deg
# Set Queue Properties for link 1->2
[[$ns link $node(1) $node(2)] queue] set limit_ 20

$ns simplex-link $node(1) $node(15) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(1) $node(15) queuePos 0.5
$ns simplex-link-op $node(1) $node(15) color black
$ns simplex-link-op $node(1) $node(15) orient 299.1deg
# Set Queue Properties for link 1->15
[[$ns link $node(1) $node(15)] queue] set limit_ 20

$ns simplex-link $node(2) $node(1) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(2) $node(1) queuePos 0.5
$ns simplex-link-op $node(2) $node(1) color black
$ns simplex-link-op $node(2) $node(1) orient 180.9deg
# Set Queue Properties for link 2->1
[[$ns link $node(2) $node(1)] queue] set limit_ 20

$ns simplex-link $node(2) $node(3) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(2) $node(3) queuePos 0.5
```

```
$ns simplex-link-op $node(14) $node(43) color black
$ns simplex-link-op $node(14) $node(43) orient 306.2deg
# Set Queue Properties for link 14->43
[[$ns link $node(14) $node(43)] queue] set limit_ 20

$ns simplex-link $node(15) $node(1) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(15) $node(1) queuePos 0.5
$ns simplex-link-op $node(15) $node(1) color black
$ns simplex-link-op $node(15) $node(1) orient 119.1deg
# Set Queue Properties for link 15->1
[[$ns link $node(15) $node(1)] queue] set limit_ 20

$ns simplex-link $node(15) $node(16) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(15) $node(16) queuePos 0.5
$ns simplex-link-op $node(15) $node(16) color black
$ns simplex-link-op $node(15) $node(16) orient 352.8deg
# Set Queue Properties for link 15->16
[[$ns link $node(15) $node(16)] queue] set limit_ 20

$ns simplex-link $node(16) $node(15) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(16) $node(15) queuePos 0.5
$ns simplex-link-op $node(16) $node(15) color black
$ns simplex-link-op $node(16) $node(15) orient 172.8deg
# Set Queue Properties for link 16->15
[[$ns link $node(16) $node(15)] queue] set limit_ 20

$ns simplex-link $node(16) $node(17) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(16) $node(17) queuePos 0.5
$ns simplex-link-op $node(16) $node(17) color black
$ns simplex-link-op $node(16) $node(17) orient 357.7deg
# Set Queue Properties for link 16->17
[[$ns link $node(16) $node(17)] queue] set limit_ 20

$ns simplex-link $node(17) $node(16) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(17) $node(16) queuePos 0.5
$ns simplex-link-op $node(17) $node(16) color black
$ns simplex-link-op $node(17) $node(16) orient 177.7deg
# Set Queue Properties for link 17->16
[[$ns link $node(17) $node(16)] queue] set limit_ 20
```

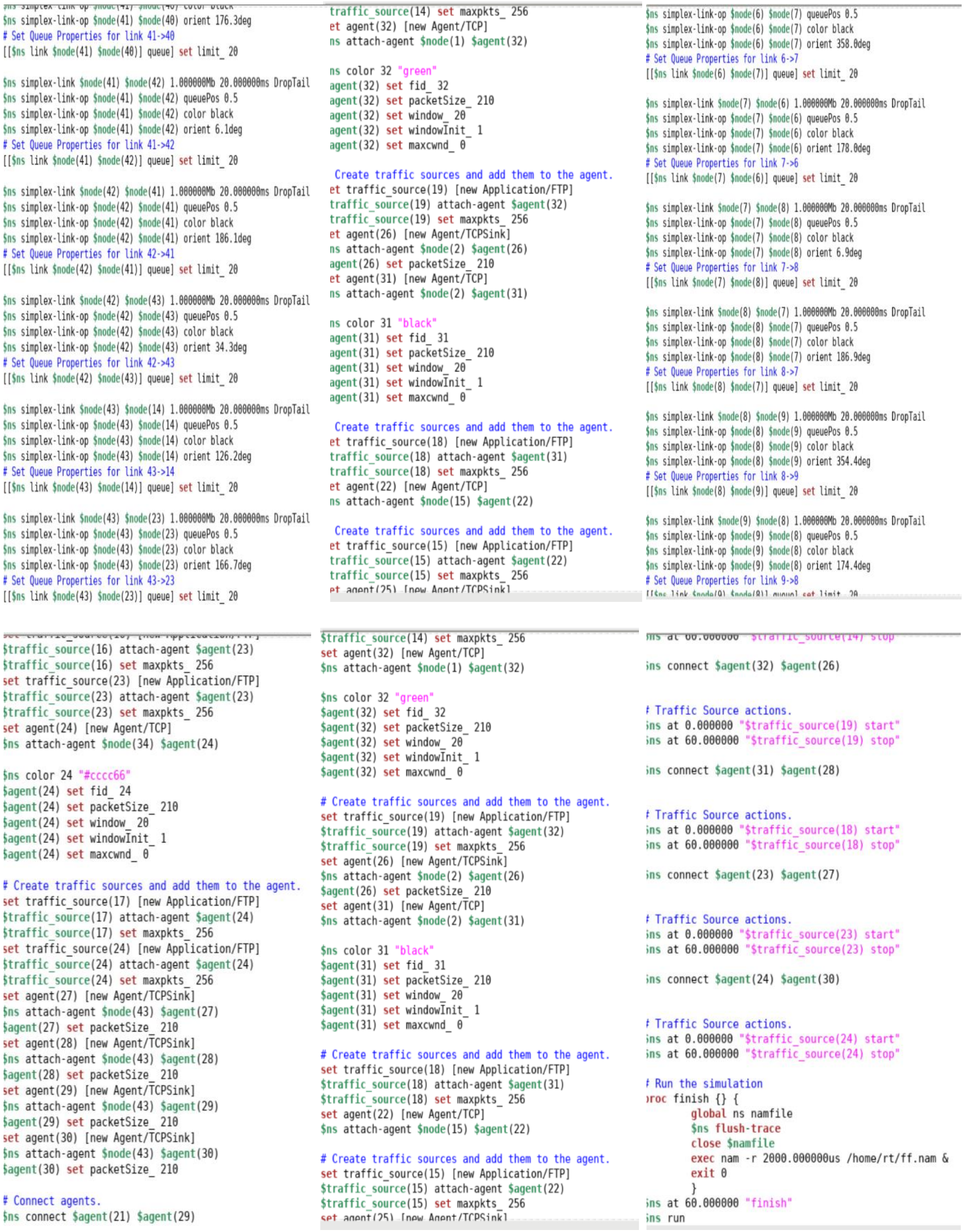
```
$ns simplex-link-op $node(43) $node(14) orient 126.2deg
# Set Queue Properties for link 43->14
[[$ns link $node(43) $node(14)] queue] set limit_ 20

$ns simplex-link $node(43) $node(23) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(43) $node(23) queuePos 0.5
$ns simplex-link-op $node(43) $node(23) color black
$ns simplex-link-op $node(43) $node(23) orient 166.7deg
# Set Queue Properties for link 43->23
[[$ns link $node(43) $node(23)] queue] set limit_ 20

$ns simplex-link $node(43) $node(33) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(43) $node(33) queuePos 0.5
$ns simplex-link-op $node(43) $node(33) color black
$ns simplex-link-op $node(43) $node(33) orient 193.9deg
# Set Queue Properties for link 43->33
[[$ns link $node(43) $node(33)] queue] set limit_ 20

$ns simplex-link $node(43) $node(42) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(43) $node(42) queuePos 0.5
$ns simplex-link-op $node(43) $node(42) color black
$ns simplex-link-op $node(43) $node(42) orient 214.3deg
# Set Queue Properties for link 43->42
[[$ns link $node(43) $node(42)] queue] set limit_ 20

# Add Link Loss Models

# Create agents.
set agent(21) [new Agent/TCP]
$ns attach-agent $node(1) $agent(21)

$ns color 21 "red"
$agent(21) set fid_ 21
$agent(21) set packetSize_ 210
$agent(21) set window_ 20
$agent(21) set windowInit_ 1
$agent(21) set maxcwnd_ 0

# Create traffic sources and add them to the agent.
set traffic_source(14) [new Application/FTP]
```



```
$ns simplex-link $node(20) $node(21) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(20) $node(21) queuePos 0.5
$ns simplex-link-op $node(20) $node(21) color black
$ns simplex-link-op $node(20) $node(21) orient 3.7deg
# Set Queue Properties for link 20->21
[[$ns link $node(20) $node(21)] queue] set limit_ 20

$ns simplex-link $node(21) $node(20) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(21) $node(20) queuePos 0.5
$ns simplex-link-op $node(21) $node(20) color black
$ns simplex-link-op $node(21) $node(20) orient 183.7deg
# Set Queue Properties for link 21->20
[[$ns link $node(21) $node(20)] queue] set limit_ 20

$ns simplex-link $node(21) $node(22) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(21) $node(22) queuePos 0.5
$ns simplex-link-op $node(21) $node(22) color black
$ns simplex-link-op $node(21) $node(22) orient 1.3deg
# Set Queue Properties for link 21->22
[[$ns link $node(21) $node(22)] queue] set limit_ 20

$ns simplex-link $node(22) $node(21) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(22) $node(21) queuePos 0.5
$ns simplex-link-op $node(22) $node(21) color black
$ns simplex-link-op $node(22) $node(21) orient 181.3deg
# Set Queue Properties for link 22->21
[[$ns link $node(22) $node(21)] queue] set limit_ 20

$ns simplex-link $node(22) $node(23) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(22) $node(23) queuePos 0.5
$ns simplex-link-op $node(22) $node(23) color black
$ns simplex-link-op $node(22) $node(23) orient 357.5deg
# Set Queue Properties for link 22->23
[[$ns link $node(22) $node(23)] queue] set limit_ 20

$ns simplex-link $node(23) $node(22) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(23) $node(22) queuePos 0.5
$ns simplex-link-op $node(23) $node(22) color black
```

```
$ns simplex-link $node(22) $node(21) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(22) $node(21) queuePos 0.5
$ns simplex-link-op $node(22) $node(21) color black
$ns simplex-link-op $node(22) $node(21) orient 181.3deg
# Set Queue Properties for link 22->21
[[$ns link $node(22) $node(21)] queue] set limit_ 20

$ns simplex-link $node(22) $node(23) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(22) $node(23) queuePos 0.5
$ns simplex-link-op $node(22) $node(23) color black
$ns simplex-link-op $node(22) $node(23) orient 357.5deg
# Set Queue Properties for link 22->23
[[$ns link $node(22) $node(23)] queue] set limit_ 20

$ns simplex-link $node(23) $node(22) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(23) $node(22) queuePos 0.5
$ns simplex-link-op $node(23) $node(22) color black
$ns simplex-link-op $node(23) $node(22) orient 177.5deg
# Set Queue Properties for link 23->22
[[$ns link $node(23) $node(22)] queue] set limit_ 20

$ns simplex-link $node(23) $node(43) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(23) $node(43) queuePos 0.5
$ns simplex-link-op $node(23) $node(43) color black
$ns simplex-link-op $node(23) $node(43) orient 346.7deg
# Set Queue Properties for link 23->43
[[$ns link $node(23) $node(43)] queue] set limit_ 20

$ns simplex-link $node(24) $node(25) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(24) $node(25) queuePos 0.5
$ns simplex-link-op $node(24) $node(25) color black
$ns simplex-link-op $node(24) $node(25) orient 0.5deg
# Set Queue Properties for link 24->25
[[$ns link $node(24) $node(25)] queue] set limit_ 20

$ns simplex-link $node(25) $node(24) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(25) $node(24) queuePos 0.5
$ns simplex-link-op $node(25) $node(24) color black
$ns simplex-link-op $node(25) $node(24) orient 180.5deg
```

```
ns simplex-link $node(4) $node(3) 1.000000Mb 20.000000ms DropTail
ns simplex-link-op $node(4) $node(3) queuePos 0.5
ns simplex-link-op $node(4) $node(3) color black
ns simplex-link-op $node(4) $node(3) orient 178.3deg
 Set Queue Properties for link 4->3
[$ns link $node(4) $node(3)] queue] set limit_ 20

ns simplex-link $node(4) $node(5) 1.000000Mb 20.000000ms DropTail
ns simplex-link-op $node(4) $node(5) queuePos 0.5
ns simplex-link-op $node(4) $node(5) color black
ns simplex-link-op $node(4) $node(5) orient 2.5deg
 Set Queue Properties for link 4->5
[$ns link $node(4) $node(5)] queue] set limit_ 20

ns simplex-link $node(5) $node(4) 1.000000Mb 20.000000ms DropTail
ns simplex-link-op $node(5) $node(4) queuePos 0.5
ns simplex-link-op $node(5) $node(4) color black
ns simplex-link-op $node(5) $node(4) orient 182.5deg
 Set Queue Properties for link 5->4
[$ns link $node(5) $node(4)] queue] set limit_ 20

ns simplex-link $node(5) $node(6) 1.000000Mb 20.000000ms DropTail
ns simplex-link-op $node(5) $node(6) queuePos 0.5
ns simplex-link-op $node(5) $node(6) color black
ns simplex-link-op $node(5) $node(6) orient 1.4deg
 Set Queue Properties for link 5->6
[$ns link $node(5) $node(6)] queue] set limit_ 20

ns simplex-link $node(6) $node(5) 1.000000Mb 20.000000ms DropTail
ns simplex-link-op $node(6) $node(5) queuePos 0.5
ns simplex-link-op $node(6) $node(5) color black
ns simplex-link-op $node(6) $node(5) orient 181.4deg
 Set Queue Properties for link 6->5
[$ns link $node(6) $node(5)] queue] set limit_ 20

ns simplex-link $node(6) $node(7) 1.000000Mb 20.000000ms DropTail
ns simplex-link-op $node(6) $node(7) queuePos 0.5
ns simplex-link-op $node(6) $node(7) color black
```



```
$ns simplex-link-op $node(41) $node(40) color black
$ns simplex-link-op $node(41) $node(40) orient 176.3deg
# Set Queue Properties for link 41->40
[[$ns link $node(41) $node(40)] queue] set limit_ 20

$ns simplex-link $node(41) $node(42) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(41) $node(42) queuePos 0.5
$ns simplex-link-op $node(41) $node(42) color black
$ns simplex-link-op $node(41) $node(42) orient 6.1deg
# Set Queue Properties for link 41->42
[[$ns link $node(41) $node(42)] queue] set limit_ 20

$ns simplex-link $node(42) $node(41) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(42) $node(41) queuePos 0.5
$ns simplex-link-op $node(42) $node(41) color black
$ns simplex-link-op $node(42) $node(41) orient 186.1deg
# Set Queue Properties for link 42->41
[[$ns link $node(42) $node(41)] queue] set limit_ 20

$ns simplex-link $node(42) $node(43) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(42) $node(43) queuePos 0.5
$ns simplex-link-op $node(42) $node(43) color black
$ns simplex-link-op $node(42) $node(43) orient 34.3deg
# Set Queue Properties for link 42->43
[[$ns link $node(42) $node(43)] queue] set limit_ 20

$ns simplex-link $node(43) $node(14) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(43) $node(14) queuePos 0.5
$ns simplex-link-op $node(43) $node(14) color black
$ns simplex-link-op $node(43) $node(14) orient 126.2deg
# Set Queue Properties for link 43->14
[[$ns link $node(43) $node(14)] queue] set limit_ 20

$ns simplex-link $node(43) $node(23) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(43) $node(23) queuePos 0.5
$ns simplex-link-op $node(43) $node(23) color black
$ns simplex-link-op $node(43) $node(23) orient 166.7deg
# Set Queue Properties for link 43->23
[[$ns link $node(43) $node(23)] queue] set limit_ 20
```

```
traffic_source(14) set maxpkts_ 256
et agent(32) [new Agent/TCP]
ns attach-agent $node(1) $agent(32)

ns color 32 "green"
agent(32) set fid_ 32
agent(32) set packetSize_ 210
agent(32) set window_ 20
agent(32) set windowInit_ 1
agent(32) set maxcwnd_ 0

 Create traffic sources and add them to the agent.
et traffic_source(19) [new Application/FTP]
traffic_source(19) attach-agent $agent(32)
traffic_source(19) set maxpkts_ 256
et agent(26) [new Agent/TCPSink]
ns attach-agent $node(2) $agent(26)
agent(26) set packetSize_ 210
et agent(31) [new Agent/TCP]
ns attach-agent $node(2) $agent(31)

ns color 31 "black"
agent(31) set fid_ 31
agent(31) set packetSize_ 210
agent(31) set window_ 20
agent(31) set windowInit_ 1
agent(31) set maxcwnd_ 0

 Create traffic sources and add them to the agent.
et traffic_source(18) [new Application/FTP]
traffic_source(18) attach-agent $agent(31)
traffic_source(18) set maxpkts_ 256
et agent(22) [new Agent/TCP]
ns attach-agent $node(15) $agent(22)

 Create traffic sources and add them to the agent.
et traffic_source(15) [new Application/FTP]
traffic_source(15) attach-agent $agent(22)
traffic_source(15) set maxpkts_ 256
et agent(25) [new Agent/TCPSink]
```

```
$ns simplex-link-op $node(6) $node(7) queuePos 0.5
$ns simplex-link-op $node(6) $node(7) color black
$ns simplex-link-op $node(6) $node(7) orient 358.0deg
# Set Queue Properties for link 6->7
[[$ns link $node(6) $node(7)] queue] set limit_ 20

$ns simplex-link $node(7) $node(6) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(7) $node(6) queuePos 0.5
$ns simplex-link-op $node(7) $node(6) color black
$ns simplex-link-op $node(7) $node(6) orient 178.0deg
# Set Queue Properties for link 7->6
[[$ns link $node(7) $node(6)] queue] set limit_ 20

$ns simplex-link $node(7) $node(8) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(7) $node(8) queuePos 0.5
$ns simplex-link-op $node(7) $node(8) color black
$ns simplex-link-op $node(7) $node(8) orient 6.9deg
# Set Queue Properties for link 7->8
[[$ns link $node(7) $node(8)] queue] set limit_ 20

$ns simplex-link $node(8) $node(7) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(8) $node(7) queuePos 0.5
$ns simplex-link-op $node(8) $node(7) color black
$ns simplex-link-op $node(8) $node(7) orient 186.9deg
# Set Queue Properties for link 8->7
[[$ns link $node(8) $node(7)] queue] set limit_ 20

$ns simplex-link $node(8) $node(9) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(8) $node(9) queuePos 0.5
$ns simplex-link-op $node(8) $node(9) color black
$ns simplex-link-op $node(8) $node(9) orient 354.4deg
# Set Queue Properties for link 8->9
[[$ns link $node(8) $node(9)] queue] set limit_ 20

$ns simplex-link $node(9) $node(8) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(9) $node(8) queuePos 0.5
$ns simplex-link-op $node(9) $node(8) color black
$ns simplex-link-op $node(9) $node(8) orient 174.4deg
# Set Queue Properties for link 9->8
[[$ns link $node(9) $node(8)] queue] set limit_ 20
```

```
set traffic_source(16) [new Application/FTP]
$traffic_source(16) attach-agent $agent(23)
$traffic_source(16) set maxpkts_ 256
set traffic_source(23) [new Application/FTP]
$traffic_source(23) attach-agent $agent(23)
$traffic_source(23) set maxpkts_ 256
set agent(24) [new Agent/TCP]
$ns attach-agent $node(34) $agent(24)

$ns color 24 "#cccc66"
$agent(24) set fid_ 24
$agent(24) set packetSize_ 210
$agent(24) set window_ 20
$agent(24) set windowInit_ 1
$agent(24) set maxcwnd_ 0

# Create traffic sources and add them to the agent.
set traffic_source(17) [new Application/FTP]
$traffic_source(17) attach-agent $agent(24)
$traffic_source(17) set maxpkts_ 256
set traffic_source(24) [new Application/FTP]
$traffic_source(24) attach-agent $agent(24)
$traffic_source(24) set maxpkts_ 256
set agent(27) [new Agent/TCPSink]
$ns attach-agent $node(43) $agent(27)
$agent(27) set packetSize_ 210
set agent(28) [new Agent/TCPSink]
$ns attach-agent $node(43) $agent(28)
$agent(28) set packetSize_ 210
set agent(29) [new Agent/TCPSink]
$ns attach-agent $node(43) $agent(29)
$agent(29) set packetSize_ 210
set agent(30) [new Agent/TCPSink]
$ns attach-agent $node(43) $agent(30)
$agent(30) set packetSize_ 210

# Connect agents.
$ns connect $agent(21) $agent(29)
```

```
$traffic_source(14) set maxpkts_ 256
set agent(32) [new Agent/TCP]
$ns attach-agent $node(1) $agent(32)

$ns color 32 "green"
$agent(32) set fid_ 32
$agent(32) set packetSize_ 210
$agent(32) set window_ 20
$agent(32) set windowInit_ 1
$agent(32) set maxcwnd_ 0

# Create traffic sources and add them to the agent.
set traffic_source(19) [new Application/FTP]
$traffic_source(19) attach-agent $agent(32)
$traffic_source(19) set maxpkts_ 256
set agent(26) [new Agent/TCPSink]
$ns attach-agent $node(2) $agent(26)
$agent(26) set packetSize_ 210
set agent(31) [new Agent/TCP]
$ns attach-agent $node(2) $agent(31)

$ns color 31 "black"
$agent(31) set fid_ 31
$agent(31) set packetSize_ 210
$agent(31) set window_ 20
$agent(31) set windowInit_ 1
$agent(31) set maxcwnd_ 0

# Create traffic sources and add them to the agent.
set traffic_source(18) [new Application/FTP]
$traffic_source(18) attach-agent $agent(31)
$traffic_source(18) set maxpkts_ 256
set agent(22) [new Agent/TCP]
$ns attach-agent $node(15) $agent(22)

# Create traffic sources and add them to the agent.
set traffic_source(15) [new Application/FTP]
$traffic_source(15) attach-agent $agent(22)
$traffic_source(15) set maxpkts_ 256
set agent(25) [new Agent/TCPSink]
```

```
$ns at 60.000000 "$traffic_source(14) stop"

$ns connect $agent(32) $agent(26)

# Traffic Source actions.
$ns at 0.000000 "$traffic_source(19) start"
$ns at 60.000000 "$traffic_source(19) stop"

$ns connect $agent(31) $agent(28)

# Traffic Source actions.
$ns at 0.000000 "$traffic_source(18) start"
$ns at 60.000000 "$traffic_source(18) stop"

$ns connect $agent(23) $agent(27)

# Traffic Source actions.
$ns at 0.000000 "$traffic_source(23) start"
$ns at 60.000000 "$traffic_source(23) stop"

$ns connect $agent(24) $agent(30)

# Traffic Source actions.
$ns at 0.000000 "$traffic_source(24) start"
$ns at 60.000000 "$traffic_source(24) stop"

# Run the simulation
proc finish {} {
        global ns namfile
        $ns flush-trace
        close $namfile
        exec nam -r 2000.000000us /home/rt/ff.nam &
        exit 0
        }
$ns at 60.000000 "finish"
$ns run
```

Fig 15: Source code of NS2

8. Output topology of NS2

After completing the coding into TCL file then running the code then this graphical inter face will be shown.

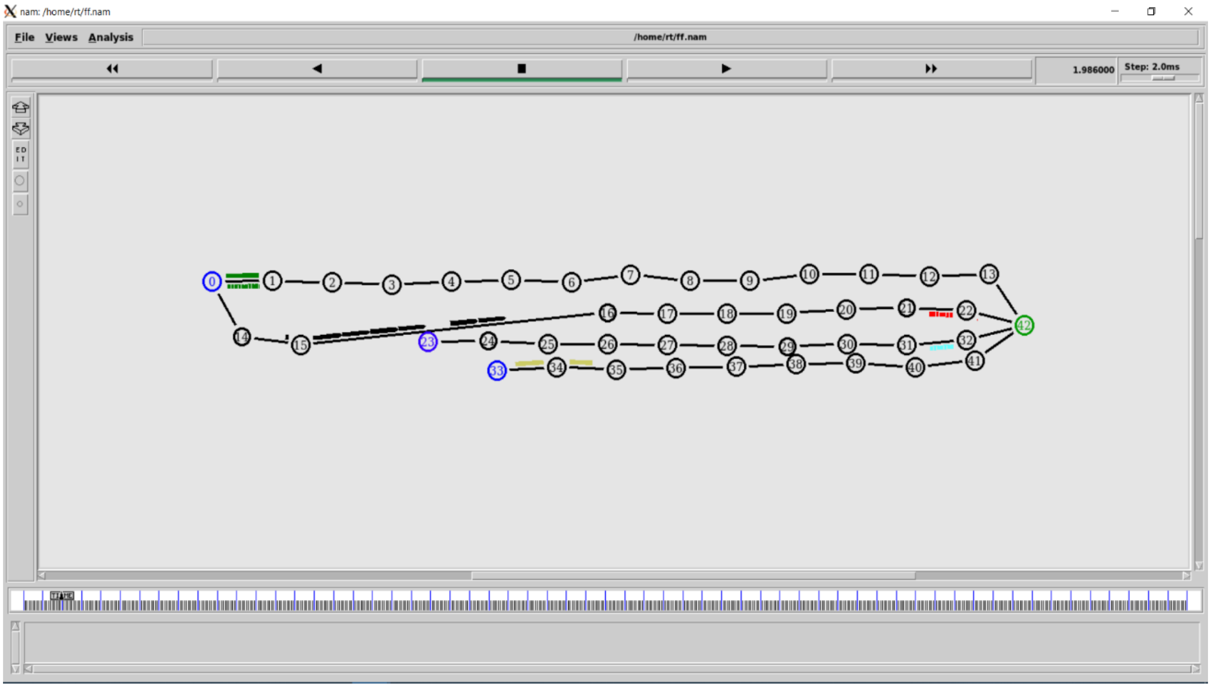


Fig 16: Fast output of NS2

Fig 16 is the starting position of the NS2. Here we can see that the data packet is very small and they are started to going to the target point. The data packet is going serially by one by one.

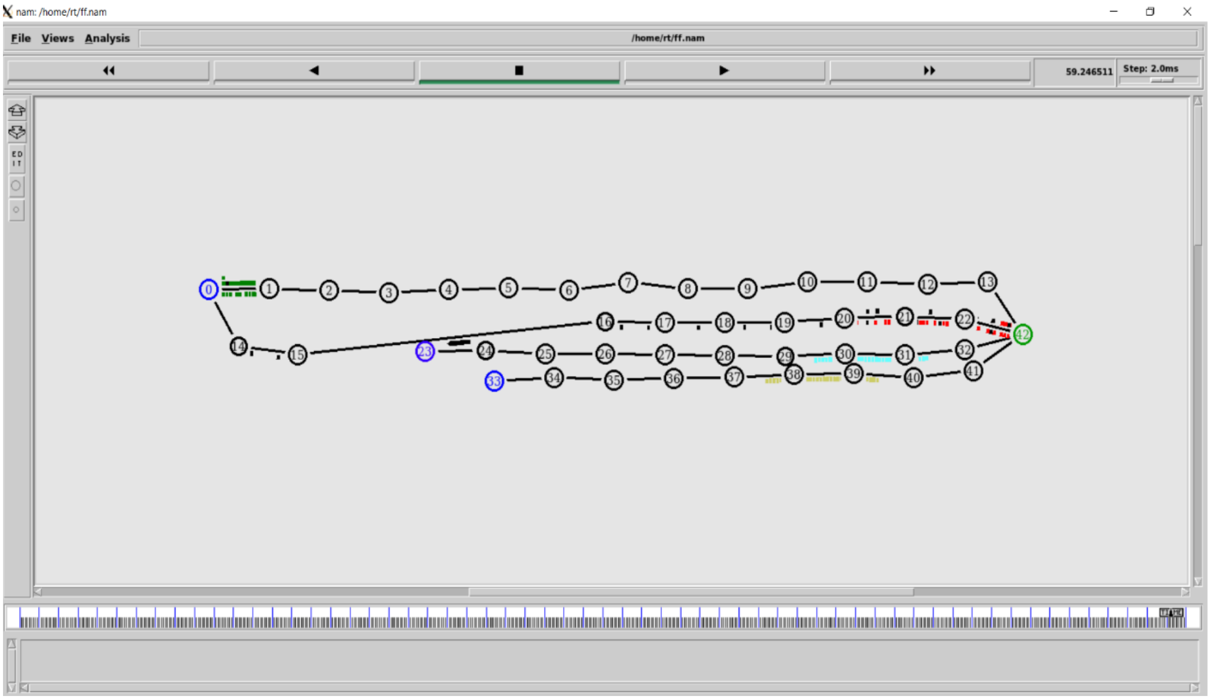


Fig 17: Second output of NS2

Fig 17 is the middle position of the ns2. In this position we can see that the data packets are going to the target point and the target point are sending back the data. It called hand shaking method.

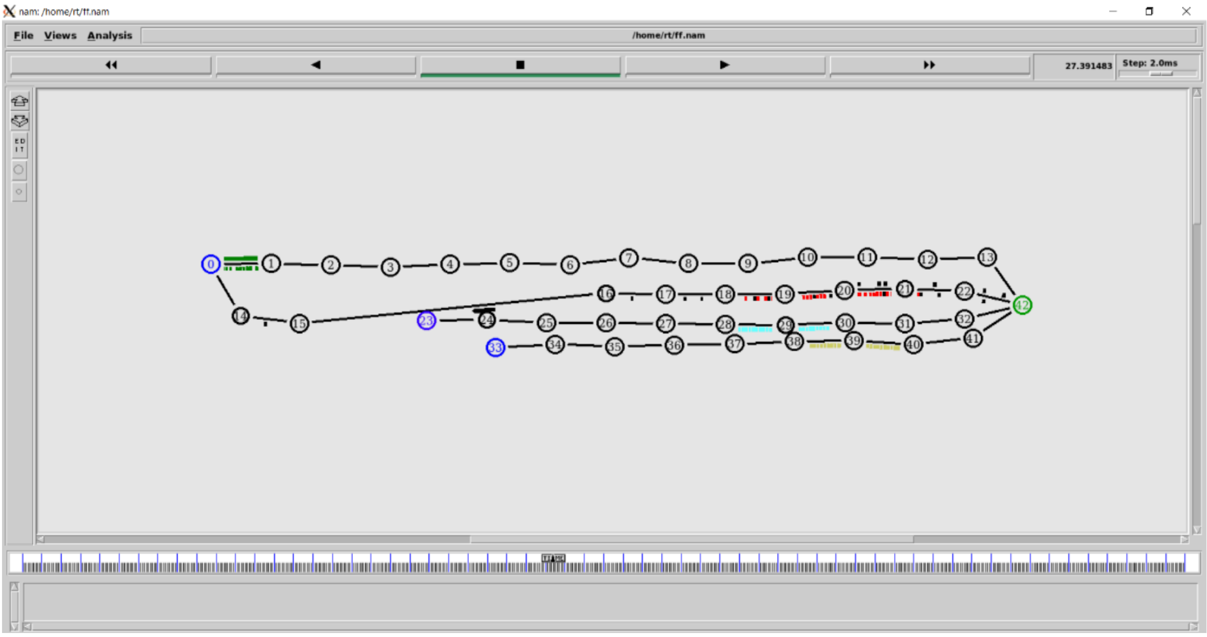


Fig 18: Third output of NS2

Fig 18 is the last position of the ns2. In this position we can see that the data packets are going to the target point but the data packets are very smaller and they are return into starting point.

9. Summarization of the designed network

In Zenmap frequently used scans can be saved as profiles to make them easy to run repeatedly. It arranges its display to show all ports on a host or all hosts running a specific service.

♣ After scan 4 different host to the domain telegram.com we find 4 different topology design.

♣ After completed all 4 scans in Zenmap collect all IP addresses of the 4 different network and store in a excel sheet and prepare a final sheet of all 4 networks organized level by level.

♣ Network Simulator 2 (NS2) provides substantial support for simulation of different protocols over wired and wireless networks. It provides a highly modular platform for wired and wireless simulations supporting different network elements, protocols, traffic, and routing types.

♣ Source code is used in NS2 to demonstrate network flow. Finally, four different source use ns2 to pass data packet to reach the destination telegram.

10. Conclusion

From this project, we learnt the basic operation of NS2 network simulator. How it works and also had a study of different routing protocols. The main learning of our project is the practical understanding of zenmap, xming, ns2, excel. It is also a good experience to work on ubuntu.

11. References

[1] https://ipwithease.com/introduction-to-zenmap-nmap network-mapper/

[2] https://link.springer.com/chapter/10.1007/978-1-4614 1406-3_2

[3] https://nmap.org/book/zenmap-results.html