

North Western University

Department of Computer Science and Engineering

Sessional Final Report

Title: Introduction to computer network using NS2
based on the domain as www.w3schools.com

Submitted To:

Md. Shymon Islam

Lecturer

Department of CSE

North Western University

Submitted By :

Md abu siddik

Id:20201131010

Department of CSE

North Western University

Mrittika Banerjee

Id : 20201140010

Department of CSE

North Western University

Tahmina Akter

Id : 20201148010

Department of CSE

North Western University

Date of submission: December 22,2022

Course Code: CSE-3304

Course Title: Computer Network Sessional

Table of contents

1.Overview of the project	3
2.Introduction to Zenmap	4
3.Different host to domain	5
4.Design Network Topology	7
5.Prepare excel sheet for network diagram	8
6.Introduction to NS2	11
8.Output topology of NS2	16
10.Conclusion	18
11.Reference	18

1. Overview of the project

This is a networking project. Here we can see the pass of network bus from one node to another node. Here we use zenmap and ns2(network simulator 2) for our project work. Firstly we install Ubuntu virtually in our windows operation system. We connect 4 different networks with a laptop and connect with the w3schools. W3schools is our targeted ip. And then we get scan the targeted ip with zenmap and take the tropology graph. After that we make an excel sheet with this tropology graph. After that we install and open ns2. We write code and make the same node as we see in tropology graphs. Then we connect the nodes and run that. After run we can see the pass of network from network bus. We can also detect the fault of passing data or data loss from this graph. This is the main purpose of this projects.

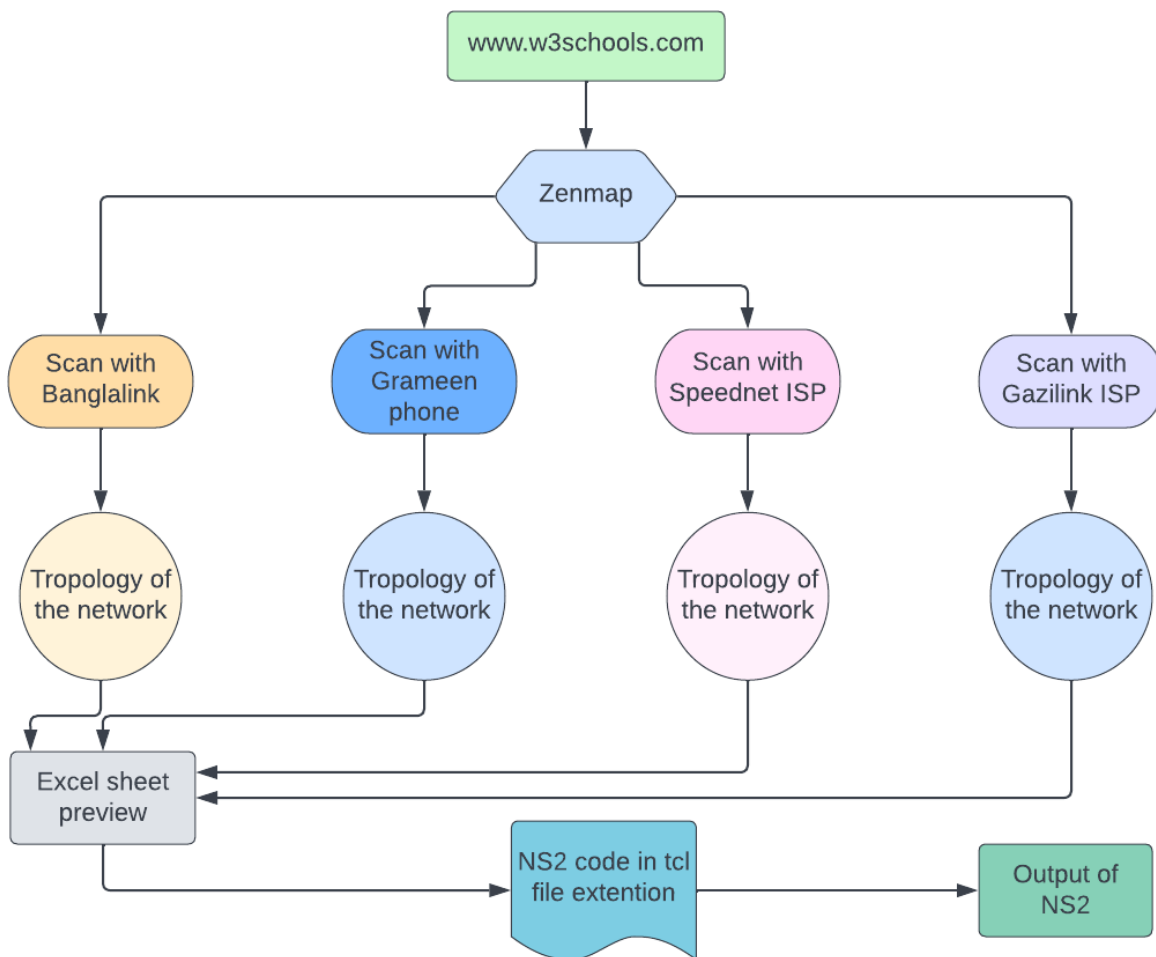


Fig 1 : Bird's eye view of the project

2.Introduction to Zenmap

Zenmap is a software that is a graphical user interface for security scanner. This is used for scanning a network and see the details about the network. It also can create a graph of tropology via which we can see how many networks are connected behind the networks. Zenmap has many features. Some of them are Target, Profile, Scan, Open scan, Namp Output, Posts / Hosts Tropology, Host Details etc. In Target field, the domain name which we selected for experiment needs to be written for scan. In Profile we can set different kind of network profile. After all of that we have to press scan and the we can see the Namp Output section. By Namp Output the host details can be shown. After scanning a domainwe can see the details about thet by this section. We can see the Script Pre-scanning and scanning ports of the domain. We can see the SYN Stealth Scan Timing, network distance and many other things. We can also see a overview of tropology by click on the tropology. Host details show the details about host. It also show the accuracy, host status, host address, hostname etc in Host Details section. Zenmap is too much important for see and investigate about a network. We can see any type of fault in network by zenmap.

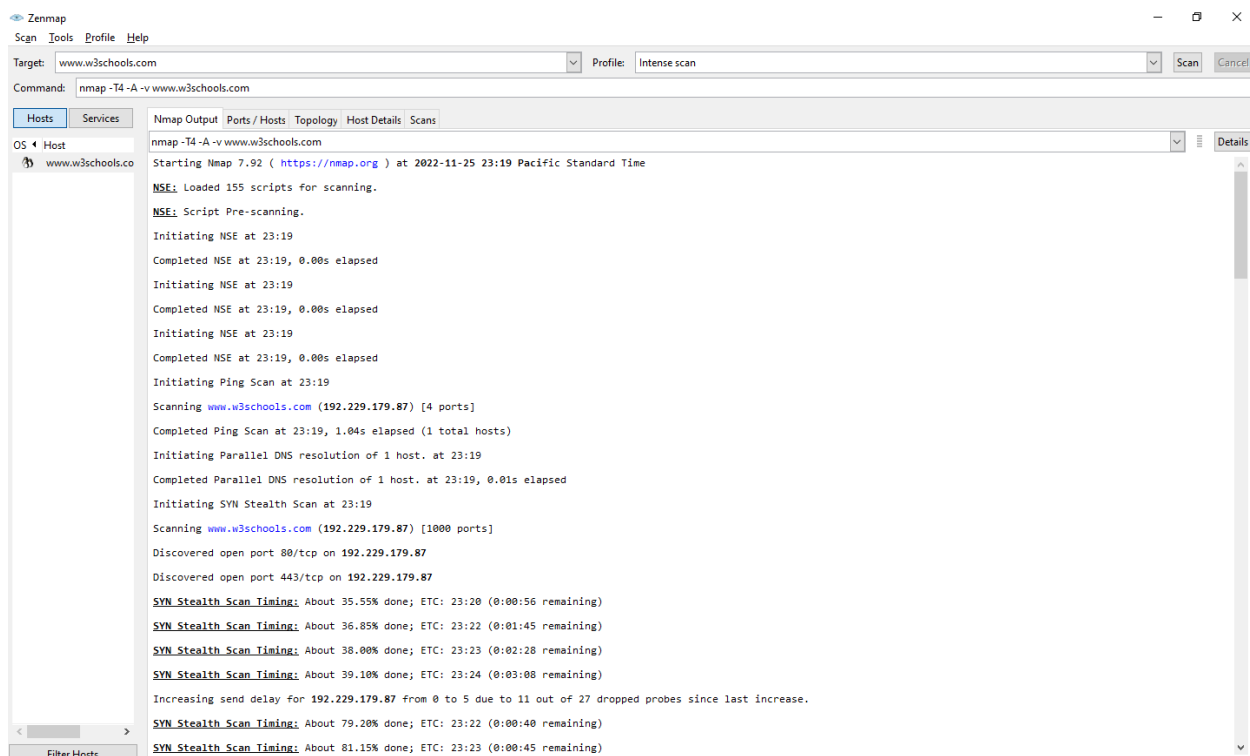


Fig 2 : Zenmap interface

3. Different host to domain

The name of our domain is w3schools. We use w3schools for our targeted domain. We connect with www.w3schools.com domain by different networks and check the data about the networks.

Hosts :

We connect the network with speednet broadband connection. Then we check details about that network

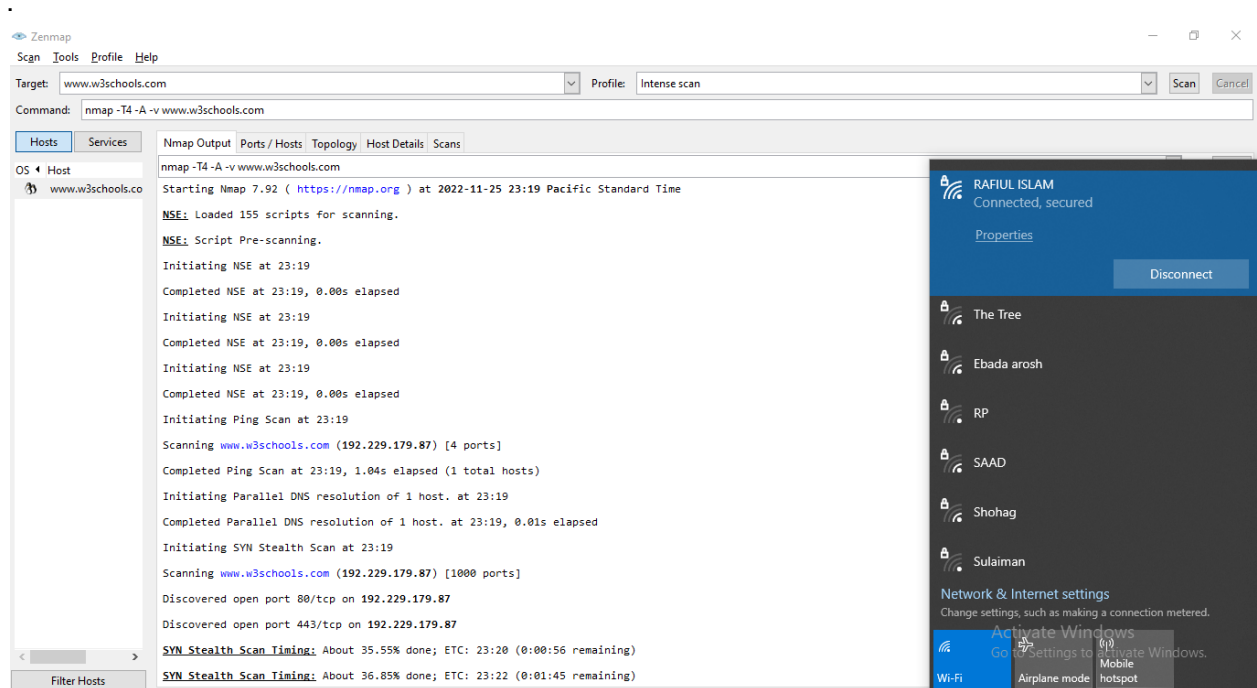


Fig 3 : Speednet network Zenmap scan output

Fig 3 is the speednet network connection which name is edited by the user. This is not the provider name. They just use there user name.

Then we connect with banglalink network. As the same way we check the network details and tropology graphs.

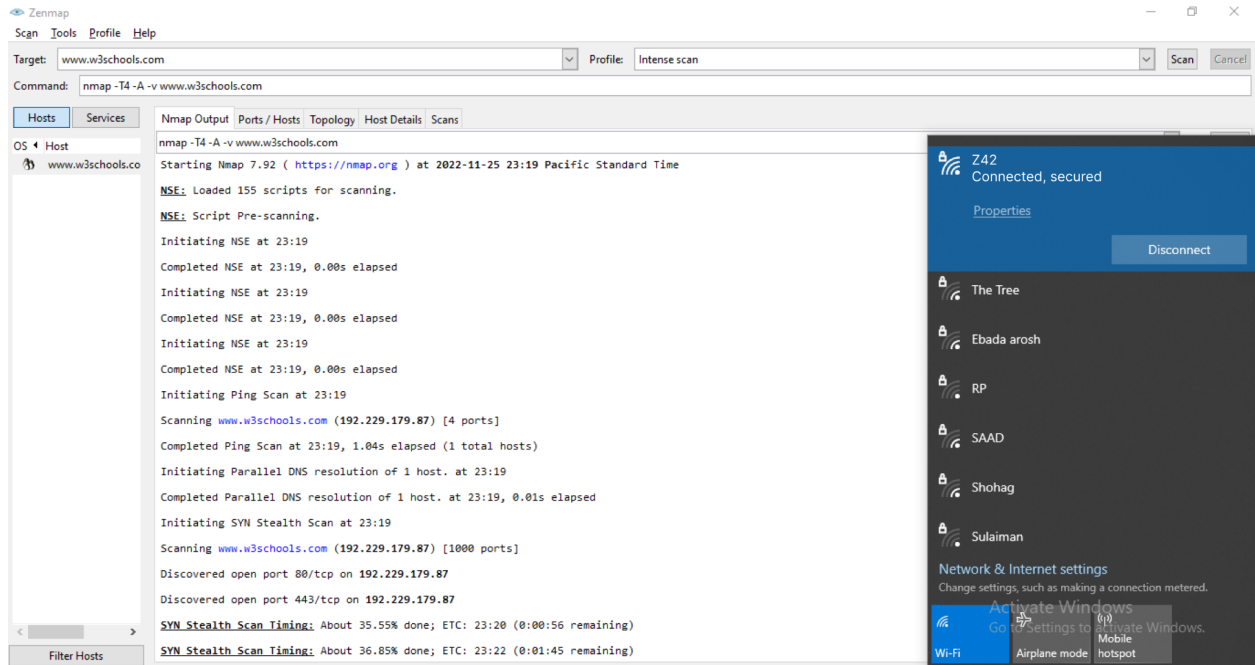


Fig 4 : Banglalink network connected by phone scan by Zenmap output

Here fig 3 shows the banglalink network. We connect this network by phone hotspot. For this, here's showing only the phone name.

After that we connect with grameenphone.

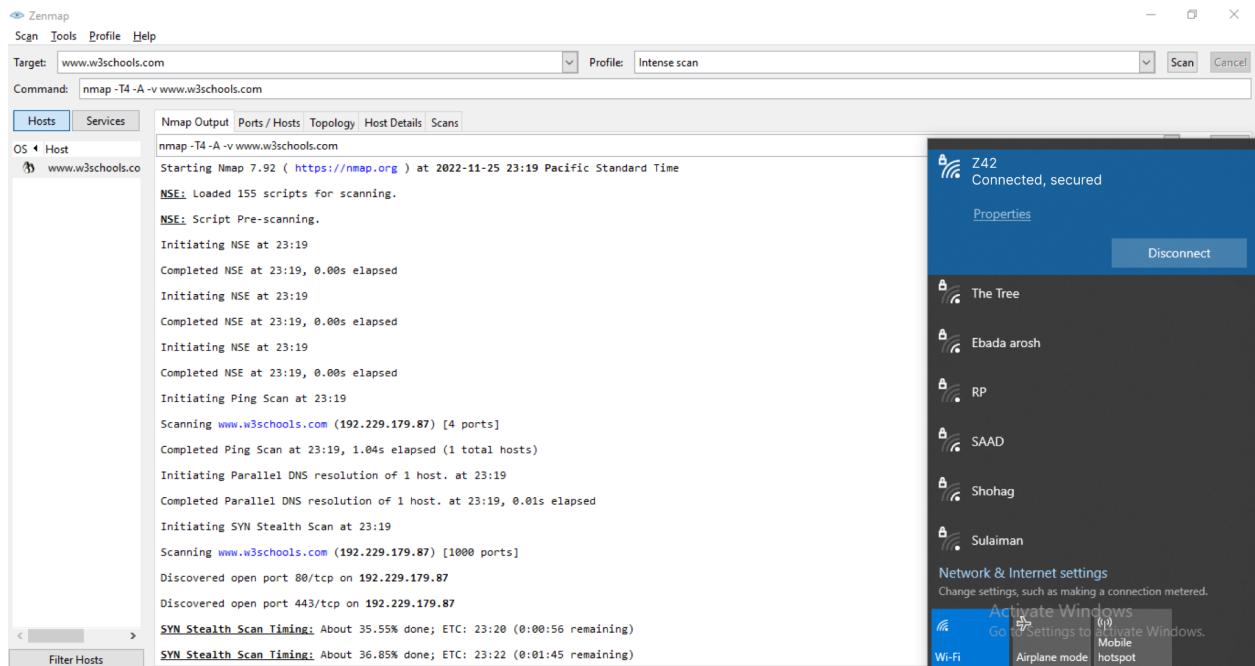


Fig 5 : Grmeen phone network connected by phone scan by Zenmap output

Here fig 3 shows the grameen phone network. We connect this network by phone hotspot. For this, here's showing only the phone name.

We connect with gazilink broadband connection. We also scan the networks as usual and check the details with topology graphs.

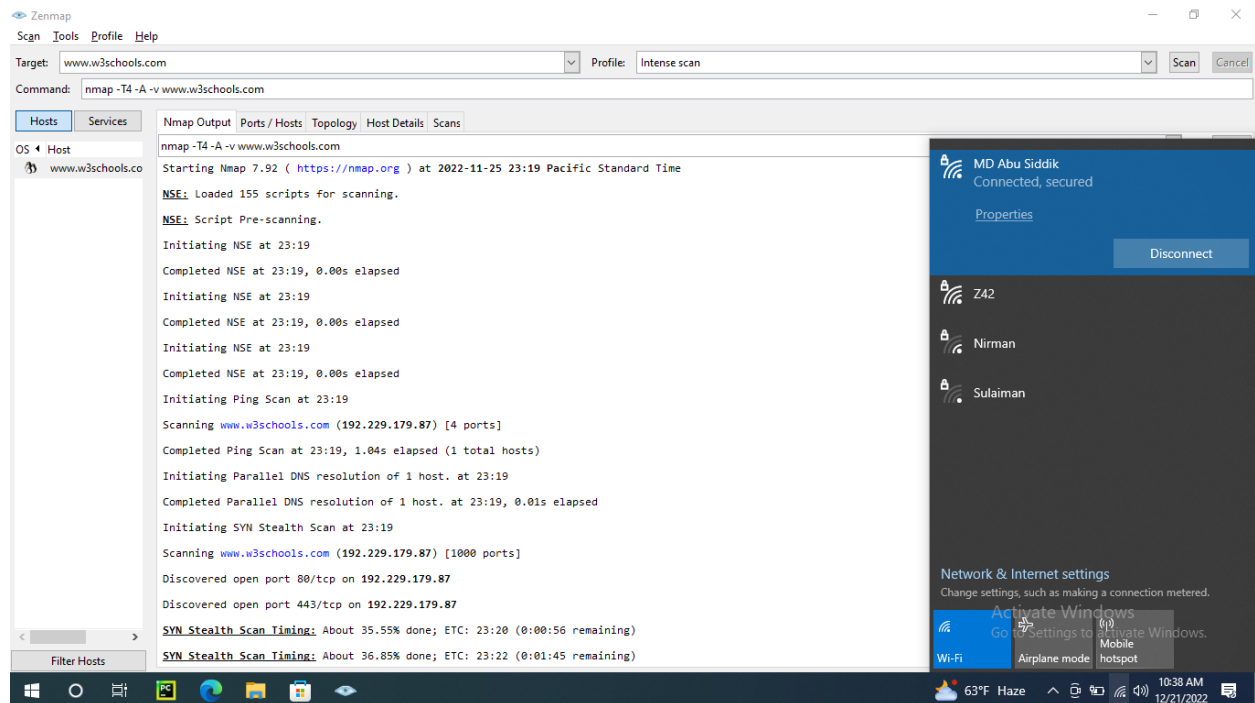


Fig 6 : Gazilink ISP Zenmap scan output

Here Fig 6 shows the Gazilink ISP output after scanning.

4.Design Network Topology

Network topology is the overall overview of a network by a graph. By this graph we can see all details about the network in a graph. Network topology can be physical and also can be logical. We all able to see the all network nodes behind the networks. Usually we're not able to see all the nodes. For this we use this kind of topology design for see the details about a networks.

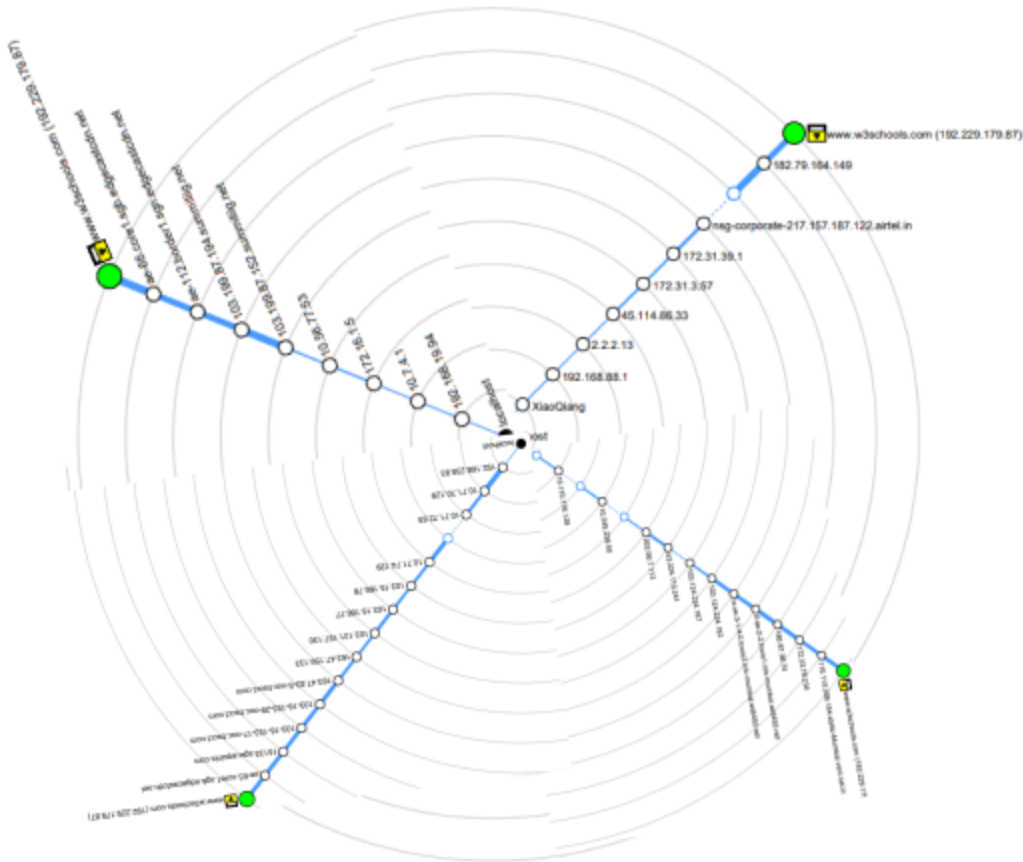


Fig 7 : Network topology of w3schools

From Fig 7 we can see this is a network topology. Here the host IP is 4 different network. But the domain or targeted IP are same.

5.Prepare excel sheet for network diagram

After analysis and scan the network topology by zenmap we have to prepare an excel sheet with this all simulation which is happen in zenmap and network topology. From the topology we can see every node have different IP addresses. Here the excel sheet sample is given below:

		Website : www.w3schools.com	
		Network : Banglalink	
SL No	IP Address	Services	
1	192.168.238.83	domain,http	
2	192.168.0.1	telnet,http	
3	10.174.21.1		
4	10.20.245.217		
5	203.76.104.166	ssh,msrpc,netbios-ssn,microsoft-ds	
6	203.76.118.249	ssh,msrpc,netbios-ssn,microsoft-ds	
7	203.76.118.253	ssh,telnet,msrpc,netbios-ssn,microsoft-ds	
8	103.9.104.105	msrpc,netbios-ssn,microsoft-ds	
9	103.9.104.65		
10	103.9.104.97		
11	123.255.90.58		
12	212.188.2.105		
13	212.188.28.214		
14	192.229.179.87		

Fig 8 : Service table of Banglalink

		Website: www.w3schools.com	
		Network : SpeedNet	
SL No	IP Address	Services	
1	192.168.1.1		
2	172.17.5.1		
3	103.79.183.210	bgp,pptp,cisco-sccp	
4	103.26.246.241	smtp,kerberos-sec,pop3,bgp,ms-lsa,http-alt,https-alt,filenet-tms	
5	157.119.185.222		
6	149.6.149.255	h323q931,cisco-sccp,dc,sip	
7	154.54.56.77	h323q931,cisco-sccp,dc,sip	
8	192.229.179.87	h323q931,cisco-sccp,dc,sip	

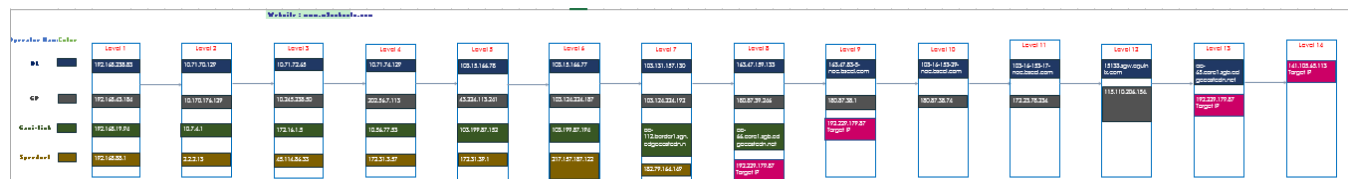
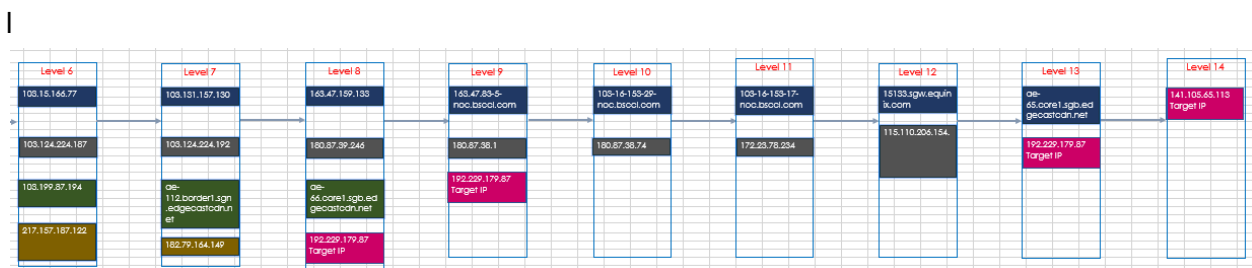
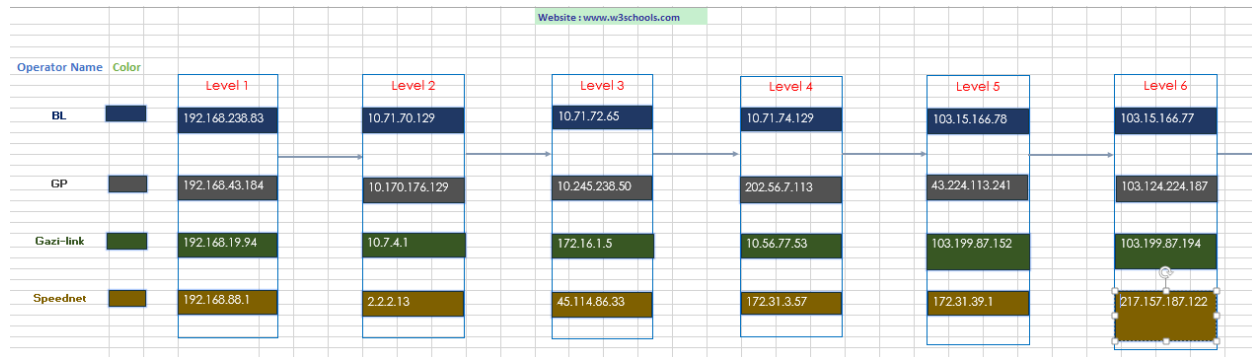
Fig 9 : Service table of speednet

	Website : www.w3schools.com	
	Network : Grameen Phone	
SL No	IP Address	Services
1	192.168.0.1	telnet,http
2	172.18.1.10	bgp,pptp,cisco-sccp
3	103.80.71.65	
4	103.26.246.101	echo,submission,LSA-or-nterm
5	157.119.185.85	
6	180.87.36.84	
7	180.87.36.83	
8	180.87.98.37	
9	129.250.8.241	
10	129.250.2.123	
11	129.250.3.101	
12	129.250.2.67	
13	192.229.179.87	

Fig 10 : Service table of Grameen Phone

	Website : www.w3schools.com	
	Network : Gazilink	
SL No	IP Address	Services
1	192.168.43.1	
2	10.99.249.22	
3	123.49.0.1	echo,ftp,ssh,telnet,smtps,ftps,nfsd-status,h323q931,cisco-sccp,dc,sip
4	93.186.133.137	echo,ftp,ssh,telnet,hosts2-ns,bgp,shell,ldp,h323q931,cisco-sccp,dc,sip
5	195.22.213.218	ftp,ssh,telnet,bgp,shell,ldp,h323q931,cisco-sccp,dc,sip
6	195.22.213.173	
7	95.211.113.154	
8	103.199.87.194	
9	192.229.179.87	http,https

Fig 11 : Service table of Gazilink



NS2 is a linux based software. For this the installation process of a windows user is quite different. We need to install ubuntu virtually on WSL. After installing linux we install the ns2 and started to code.

7.Source code of NS2

```
#-----
# 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/siddik/networking.nam w]
$ns namtrace-all $namfile

# Create wired nodes.
set node(41) [$ns node]
## node(41) at 784.719543,496.695831
$node(41) set X_ 784.719543
$node(41) set Y_ 496.695831
$node(41) set Z_ 0.0
$node(41) color "green"

set node(40) [$ns node]
## node(40) at 719.959900,489.384277
$node(40) set X_ 719.959900
$node(40) set Y_ 489.384277
$node(40) set Z_ 0.0
$node(40) color "green"

set node(39) [$ns node]
## node(39) at 625.431702,484.161652
$node(39) set X_ 625.431702
$node(39) set Y_ 484.161652
$node(39) set Z_ 0.0
$node(39) color "green"

set node(38) [$ns node]
## node(38) at 549.182434,481.550446
$node(38) set X_ 549.182434
$node(38) set Y_ 481.550446
$node(38) set Z_ 0.0
$node(38) color "green"

set node(37) [$ns node]
## node(37) at 457.265564,482.072601
$node(37) set X_ 457.265564
$node(37) set Y_ 482.072601
$node(37) set Z_ 0.0
$node(37) color "green"

set node(36) [$ns node]
## node(36) at 387.283356,479.983704
$node(36) set X_ 387.283356
$node(36) set Y_ 479.983704
$node(36) set Z_ 0.0
$node(36) color "green"

set node(35) [$ns node]
## node(35) at 328.268524,482.072601
$node(35) set X_ 328.268524
$node(35) set Y_ 482.072601
$node(35) set Z_ 0.0
$node(35) color "green"

set node(34) [$ns node]
## node(34) at 273.431732,484.683990
$node(34) set X_ 273.431732
$node(34) set Y_ 484.683990
$node(34) set Z_ 0.0
$node(34) color "green"
```

```

## node(33) at 801.431702,546.310120
$node(33) set X_ 801.431702
$node(33) set Y_ 546.310120
$node(33) set Z_ 0.0
$node(33) color "black"

set node(32) [$ns node]
## node(32) at 753.384277,542.654297
$node(32) set X_ 753.384277
$node(32) set Y_ 542.654297
$node(32) set Z_ 0.0
$node(32) color "black"

set node(31) [$ns node]
## node(31) at 695.413879,542.132080
$node(31) set X_ 695.413879
$node(31) set Y_ 542.132080
$node(31) set Z_ 0.0
$node(31) color "black"

set node(30) [$ns node]
## node(30) at 636.399048,545.265747
$node(30) set X_ 636.399048
$node(30) set Y_ 545.265747
$node(30) set Z_ 0.0
$node(30) color "black"

set node(29) [$ns node]
## node(29) at 594.618652,543.698730
$node(29) set X_ 594.618652
$node(29) set Y_ 543.698730
$node(29) set Z_ 0.0
$node(29) color "black"

set node(28) [$ns node]
## node(28) at 556.494019,546.310120
$node(28) set X_ 556.494019
$node(28) set Y_ 546.310120
$node(28) set Z_ 0.0

set node(27) [$ns node]
## node(27) at 511.580078,551.532654
$node(27) set X_ 511.580078
$node(27) set Y_ 551.532654
$node(27) set Z_ 0.0
$node(27) color "black"

set node(26) [$ns node]
## node(26) at 464.577179,548.399231
$node(26) set X_ 464.577179
$node(26) set Y_ 548.399231
$node(26) set Z_ 0.0
$node(26) color "black"

set node(25) [$ns node]
## node(25) at 426.452393,547.354614
$node(25) set X_ 426.452393
$node(25) set Y_ 547.354614
$node(25) set Z_ 0.0
$node(25) color "black"

set node(24) [$ns node]
## node(24) at 369.526672,545.265747
$node(24) set X_ 369.526672
$node(24) set Y_ 545.265747
$node(24) set Z_ 0.0
$node(24) color "black"

set node(23) [$ns node]
## node(23) at 310.511871,547.354614
$node(23) set X_ 310.511871
$node(23) set Y_ 547.354614
$node(23) set Z_ 0.0
$node(23) color "black"

set node(22) [$ns node]
## node(22) at 260.897614,548.399231
$node(22) set X_ 260.897614

```

```
## node(21) at 219.639450,547.354614
$node(21) set X_ 219.639450
$node(21) set Y_ 547.354614
$node(21) set Z_ 0.0
$node(21) color "black"

set node(20) [$ns node]
## node(20) at 802.998474,598.535645
$node(20) set X_ 802.998474
$node(20) set Y_ 598.535645
$node(20) set Z_ 0.0
$node(20) color "red"

set node(19) [$ns node]
## node(19) at 757.040039,618.381287
$node(19) set X_ 757.040039
$node(19) set Y_ 618.381287
$node(19) set Z_ 0.0
$node(19) color "red"

set node(18) [$ns node]
## node(18) at 704.814392,617.336792
$node(18) set X_ 704.814392
$node(18) set Y_ 617.336792
$node(18) set Z_ 0.0
$node(18) color "red"

set node(17) [$ns node]
## node(17) at 655.722473,614.725525
$node(17) set X_ 655.722473
$node(17) set Y_ 614.725525
$node(17) set Z_ 0.0
$node(17) color "red"

set node(16) [$ns node]
## node(16) at 610.808533,613.681030
$node(16) set X_ 610.808533
$node(16) set Y_ 613.681030
$node(16) set Z_ 0.0
```

```
$node(15) set Y_ 614.725525
$node(15) set Z_ 0.0
$node(15) color "red"

set node(14) [$ns node]
## node(14) at 500.090515,618.381287
$node(14) set X_ 500.090515
$node(14) set Y_ 618.381287
$node(14) set Z_ 0.0
$node(14) color "red"

set node(13) [$ns node]
## node(13) at 442.120270,616.814575
$node(13) set X_ 442.120270
$node(13) set Y_ 616.814575
$node(13) set Z_ 0.0
$node(13) color "red"

set node(12) [$ns node]
## node(12) at 380.494049,615.770325
$node(12) set X_ 380.494049
$node(12) set Y_ 615.770325
$node(12) set Z_ 0.0
$node(12) color "red"

set node(11) [$ns node]
## node(11) at 329.313049,615.770325
$node(11) set X_ 329.313049
$node(11) set Y_ 615.770325
$node(11) set Z_ 0.0
$node(11) color "red"

set node(10) [$ns node]
## node(10) at 284.399078,614.203247
$node(10) set X_ 284.399078
$node(10) set Y_ 614.203247
$node(10) set Z_ 0.0
$node(10) color "red"
```

```

# Create links between nodes.
$ns simplex-link $node(41) $node(8) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(41) $node(8) queuePos 0.5
$ns simplex-link-op $node(41) $node(8) color black
$ns simplex-link-op $node(41) $node(8) orient 51.2deg
# Set Queue Properties for link 41->8
[[ $ns link $node(41) $node(8) ] queue] set limit_ 20

$ns simplex-link $node(41) $node(40) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(41) $node(40) queuePos 0.5
$ns simplex-link-op $node(41) $node(40) color black
$ns simplex-link-op $node(41) $node(40) orient 186.4deg
# Set Queue Properties for link 41->40
[[ $ns link $node(41) $node(40) ] queue] set limit_ 20

$ns simplex-link $node(40) $node(41) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(40) $node(41) queuePos 0.5
$ns simplex-link-op $node(40) $node(41) color black
$ns simplex-link-op $node(40) $node(41) orient 6.4deg
# Set Queue Properties for link 40->41
[[ $ns link $node(40) $node(41) ] queue] set limit_ 20

$ns simplex-link $node(40) $node(39) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(40) $node(39) queuePos 0.5
$ns simplex-link-op $node(40) $node(39) color black
$ns simplex-link-op $node(40) $node(39) orient 183.2deg
# Set Queue Properties for link 40->39
[[ $ns link $node(40) $node(39) ] queue] set limit_ 20

$ns simplex-link $node(39) $node(40) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(39) $node(40) queuePos 0.5
$ns simplex-link-op $node(39) $node(40) color black
$ns simplex-link-op $node(39) $node(40) orient 3.2deg
# Set Queue Properties for link 39->40
[[ $ns link $node(39) $node(40) ] queue] set limit_ 20

$ns simplex-link $node(39) $node(38) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(39) $node(38) queuePos 0.5

```



```
# Set Queue Properties for link 27->28
[[ $ns link $node(27) $node(28) ] queue] set limit_ 20

$ns simplex-link $node(27) $node(26) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(27) $node(26) queuePos 0.5
$ns simplex-link-op $node(27) $node(26) color black
$ns simplex-link-op $node(27) $node(26) orient 183.8deg
# Set Queue Properties for link 27->26
[[ $ns link $node(27) $node(26) ] queue] set limit_ 20

$ns simplex-link $node(26) $node(27) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(26) $node(27) queuePos 0.5
$ns simplex-link-op $node(26) $node(27) color black
$ns simplex-link-op $node(26) $node(27) orient 3.8deg
# Set Queue Properties for link 26->27
[[ $ns link $node(26) $node(27) ] queue] set limit_ 20

$ns simplex-link $node(26) $node(25) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(26) $node(25) queuePos 0.5
$ns simplex-link-op $node(26) $node(25) color black
$ns simplex-link-op $node(26) $node(25) orient 181.6deg
# Set Queue Properties for link 26->25
[[ $ns link $node(26) $node(25) ] queue] set limit_ 20

$ns simplex-link $node(25) $node(26) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(25) $node(26) queuePos 0.5
$ns simplex-link-op $node(25) $node(26) color black
$ns simplex-link-op $node(25) $node(26) orient 1.6deg
# Set Queue Properties for link 25->26
[[ $ns link $node(25) $node(26) ] 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 182.1deg
# Set Queue Properties for link 25->24
[[ $ns link $node(25) $node(24) ] queue] set limit_ 20
```



```
$ns simplex-link $node(11) $node(12) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(11) $node(12) queuePos 0.5
$ns simplex-link-op $node(11) $node(12) color black
$ns simplex-link-op $node(11) $node(12) orient 0.0deg
# Set Queue Properties for link 11->12
[[ $ns link $node(11) $node(12)] queue] set limit_ 20

$ns simplex-link $node(11) $node(10) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(11) $node(10) queuePos 0.5
$ns simplex-link-op $node(11) $node(10) color black
$ns simplex-link-op $node(11) $node(10) orient 182.0deg
# Set Queue Properties for link 11->10
[[ $ns link $node(11) $node(10)] queue] set limit_ 20

$ns simplex-link $node(10) $node(11) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(10) $node(11) queuePos 0.5
$ns simplex-link-op $node(10) $node(11) color black
$ns simplex-link-op $node(10) $node(11) orient 2.0deg
# Set Queue Properties for link 10->11
[[ $ns link $node(10) $node(11)] queue] set limit_ 20

$ns simplex-link $node(10) $node(9) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(10) $node(9) queuePos 0.5
$ns simplex-link-op $node(10) $node(9) color black
$ns simplex-link-op $node(10) $node(9) orient 181.4deg
# Set Queue Properties for link 10->9
[[ $ns link $node(10) $node(9)] queue] set limit_ 20

$ns simplex-link $node(9) $node(10) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(9) $node(10) queuePos 0.5
$ns simplex-link-op $node(9) $node(10) color black
$ns simplex-link-op $node(9) $node(10) orient 1.4deg
# Set Queue Properties for link 9->10
[[ $ns link $node(9) $node(10)] queue] set limit_ 20

$ns simplex-link $node(8) $node(41) 1.000000Mb 20.000000ms DropTail
$ns simplex-link-op $node(8) $node(41) queuePos 0.5
$ns simplex-link-op $node(8) $node(41) color black
$ns simplex-link-op $node(8) $node(41) orient 231.2deg
```

Add Link Loss Models

```
# Create agents.
set agent(4) [new Agent/TCP]
$ns attach-agent $node(34) $agent(4)

$ns color 4 "green"
$agent(4) set fid_ 4
$agent(4) set packetSize_ 210
$agent(4) set window_ 20
$agent(4) set windowInit_ 1
$agent(4) set maxcwnd_ 0

# Create traffic sources and add them to the agent.
set traffic_source(4) [new Application/FTP]
$traffic_source(4) attach-agent $agent(4)
$traffic_source(4) set maxpkts_ 256
set agent(3) [new Agent/TCP]
$ns attach-agent $node(21) $agent(3)

$ns color 3 "black"
$agent(3) set fid_ 3
$agent(3) set packetSize_ 210
$agent(3) set window_ 20
$agent(3) set windowInit_ 1
$agent(3) set maxcwnd_ 0

# Create traffic sources and add them to the agent.
set traffic_source(3) [new Application/FTP]
$traffic_source(3) attach-agent $agent(3)
$traffic_source(3) set maxpkts_ 256
set agent(2) [new Agent/TCP]
$ns attach-agent $node(9) $agent(2)

$ns color 2 "red"
$agent(2) set fid_ 2
$agent(2) set packetSize_ 210
$agent(2) set window_ 20
$agent(2) set windowInit_ 1
$agent(2) set maxcwnd_ 0
```

```

# Create traffic sources and add them to the agent.
set traffic_source(2) [new Application/FTP]
$traffic_source(2) attach-agent $agent(2)
$traffic_source(2) set maxpkts_ 256
set agent(8) [new Agent/TCPSink]
$ns attach-agent $node(8) $agent(8)
$agent(8) set packetSize_ 210
set agent(7) [new Agent/TCPSink]
$ns attach-agent $node(8) $agent(7)
$agent(7) set packetSize_ 210
set agent(6) [new Agent/TCPSink]
$ns attach-agent $node(8) $agent(6)
$agent(6) set packetSize_ 210
set agent(5) [new Agent/TCPSink]
$ns attach-agent $node(8) $agent(5)
$agent(5) set packetSize_ 210
set agent(1) [new Agent/TCP]
$ns attach-agent $node(1) $agent(1)

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

# Create traffic sources and add them to the agent.
set traffic_source(1) [new Application/FTP]
$traffic_source(1) attach-agent $agent(1)
$traffic_source(1) set maxpkts_ 256

# Connect agents.
$ns connect $agent(4) $agent(8)

# Traffic Source actions.
$ns at 0.000000 "$traffic_source(4) start"
$ns at 60.000000 "$traffic_source(4) stop"

```

```

# Connect agents.
$ns connect $agent(4) $agent(8)

# Traffic Source actions.
$ns at 0.000000 "$traffic_source(4) start"
$ns at 60.000000 "$traffic_source(4) stop"

$ns connect $agent(3) $agent(7)

# Traffic Source actions.
$ns at 0.000000 "$traffic_source(3) start"
$ns at 60.000000 "$traffic_source(3) stop"

$ns connect $agent(2) $agent(6)

# Traffic Source actions.
$ns at 0.000000 "$traffic_source(2) start"
$ns at 60.000000 "$traffic_source(2) stop"

$ns connect $agent(1) $agent(5)

# Traffic Source actions.
$ns at 0.000000 "$traffic_source(1) start"
$ns at 60.000000 "$traffic_source(1) stop"

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

```

Fig 15 : Source code of NS2

8. Output topology of NS2

After complete the code ins ns2 it's time for run and see the output in graphical interface.

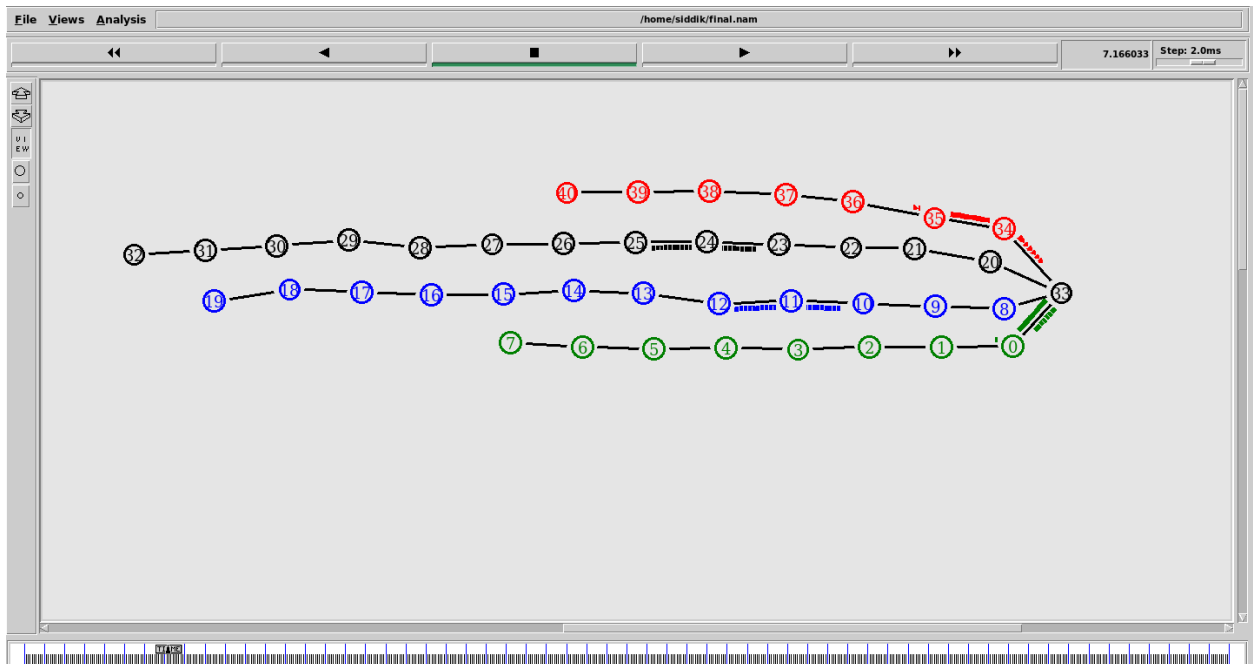


Fig 16 : First output of NS2

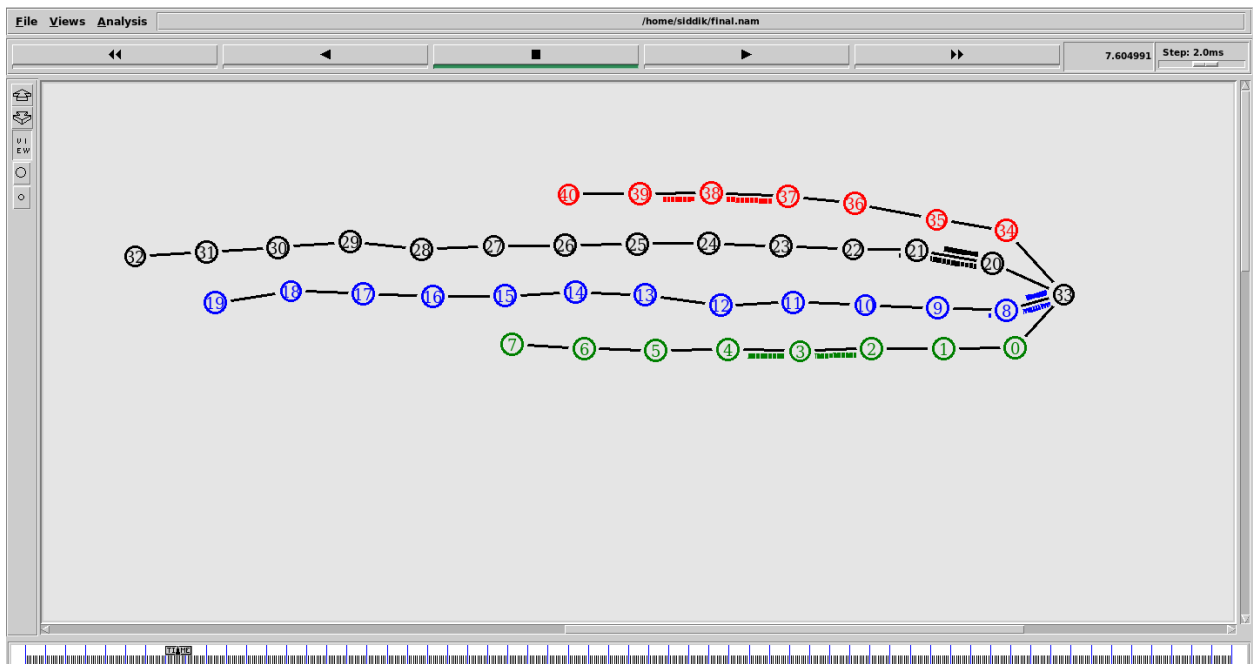


Fig 17 : Second output of NS2

10.Conclusion

Networking is a must things in modern life. In this digital world, nothing is out of networking. So gain knowledge about the networking is a must things for us. In this lab we can learn some very important things about networking. We're able to work with zenmap and ns2 by this lab which is a very useful and popular software. We also learn about wsl and ubuntu operating system from this lab which is very Nice and good experience for us. We gain so many skills from this lab about networking.

Zenmap is used for may large purpose. But we atleast learn the about that and we also use this software for this lab. Zenmap is a most popular and useful software for networking. We work with that. Network Simulator 2 is also a professional software for networking. We are able to use this professional software through this lab. Both tool are free and open sources. So we can use that in future.

11.Reference

[1] <https://beta.openai.com/playground>