JAIN DEEMED-TO-BE UNIVERSITY

FACULTY OF ENGINEERING AND TECHNOLOGY

Bachelor of Technology
in
Computer Science and Engineering

Lab Manual
for
Network Simulator Lab
(18CS57S)

Faculty of Engineering & Technology
Global Campus
45th km NH – 209, Jakkasandra Post, Kanakapura Rd, Bangalore
www.set.jainuniversity.ac.in
Fax STD Code:- 080 Fax:- 2757 7199

CONTENTS

Faculty in-charge(s) **Head of the Department**

Institute Vision and Mission

Vision:

To be a leading technical institution that offers a transformative education to create leaders and innovators with ethical values to contribute for the sustainable development and economic growth of our nation.

Mission:

M1: To impart high standard of engineering education through innovative teaching and research to meet the changing needs of the modern society.

M2: To provide outcome-based education that transforms the students to understand and solve the societal, industrial problems through engineering and technology.

M3: To collaborate with other leading technical institutions and organization to develop globally competitive technocrats and entrepreneurs.

School Vision and Mission

Vision :

To be the Nation's Leading Research and Teaching School of Computer Science & Engineering.

Mission:

M1: To create, share and apply the knowledge in Computer Science, including interdisciplinary areas.

M2: To educate students to be successful, ethical, and effective problem-solvers and life-long learners.

M3: To make students ready to respond swiftly to the challenges of the 21st century.

Department Vision and Mission

Vision:

To emerge as a model Centre for education and research in the area of Computer Science and Engineering through Knowledge acquisition, dissemination and generation to meet societal demands.

Mission:

M1: To impart the quality education in cutting edge technologies, teaching & learning ambience in Computer Science and Engineering.

M2: To establish a center of excellence in collaboration with industries, research laboratories and other agencies to meet the changing needs of society.

M3: To provide an environment conducive to develop innovation, team-spirit and Entrepreneurship.

M4: To practice and promote high standards of professional ethics and transparency.

Program Educational Objectives (PEOs)

A few years after graduation, the Graduates of Computer Science and Engineering will be able

PEO1: To excel as professionals in the area of Computer Science and Engineering with an inclination towards continuous learning.

PEO2: To involve in interdisciplinary innovative and creative research work to solve societal needs and adopt themselves to rapidly evolving technologies.

PEO3: To develop entrepreneurial skills with leadership capabilities

PEO4: To exhibit professional ethics among the graduates to transform them as responsible citizens.

Program Specific Outcomes (PSO)

PSO 1: Design and develop network, web-based, cloud-based computational systems

PSO 2: Design efficient algorithms, understand software practices and implement code with optimization

Program Outcomes

Engineering Graduate's attributes

Sl.No.	Program Outcomes
1.	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2.	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3.	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4.	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of

	data, and synthesis of the information to provide valid conclusions.
5.	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6.	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7.	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8.	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9.	**Individual and teamwork**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10.	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11.	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12.	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Course Outcome Statements and Course Articulation Matrix

Course Outcome Statements

After the completion of the course, the students are able to

18CS41L.1	Demonstrate basic of networks and Sub netting and its Implementation using simulation tools.
18CS41L.2	Implement TCP and UDP one way /two way communication using NS2.
18CS41L.3	Demonstrate devices connection and implement star and bus topology using Simulator.
18CS41L.4	Demonstrate Installation and configuration of NetAnim.
18CS41L.5	Implement FTP using TCP bulk transfer and building a hybrid topology to calculating network performance using Simulator
18CS41L.6	Analyze network traces using wire shark software

CO – PO Mapping

CO/PO: Mapping												
(H/M/L indicates strength of correlation) 3-High, 2-Medium, 1-Low												
Course Outcome (COs)	**Programme Outcome (POs)**											
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**
CO-1	3	2		3	2							
CO-2	3	2		3	2							
CO-3	3	3	3	3	2				3		3	3
CO-4	3	3	3	3	2							
CO-5	3	3		3	2							
CO-6	3	3	3	3	2				3		3	3
Average	3	2		3	2							

Rubrics for Evaluation (CA and Semester End Assessment)

Assessment and Evaluation

CA (Continuous Assessment) : Every experiment is evaluated for100 marks

Test1(Lab Internal-1) : Conducted for 100 marks (Open Ended Experiment) in the middle of the semester
Test1 (Lab Internal-1) : Conducted for 100 marks (Open Ended Experiment) towards end of the semester
Semester End Test : Conducted for 100 marks at the end of the semester

Rubrics for CA Marks :

Rubrics	Marks Allocated (100)	High	Medium	Low
Procedure	20	For given concept, algorithm and pseudo code is designed	For given concept, partial algorithm and pseudo code is designed	For given concept, low preparation before implementation but understood the concept
		20	10	5
Conduction	40	For given concept, implementation successfully done.	For given concept, implementation partially done.	For given concept, implementation still in process towards the goal.
		40	30	20
Calculation, Results, Graph	15	Desired output achieved and validation is processed.	Desired output partially achieved.	Desired output is yet to achieve.
		15	10	5
Viva/Oral	15	Student answered all the viva voce questions which include analytical skills	Student answered to all viva voce questions	Student answered to partial viva voce questions
		15	10	5
Record Writing	10	completed record was submitted on time	Record was submitted late Late submission	Record was submitted but incomplete
		10	8	5

Rubrics used for continuous evaluation lab internals

Rubrics	Marks Allocated (100)	High	Medium	Low
Write up	30	Student is able to analyze the problem statement, design the algorithm and pseudo code with expected logic and approach.	Student is able to analyze the problem statement, but partially design the algorithm and pseudo code with expected logic and approach.	Student is able to partially analyze the problem statement, but partially design the algorithm and pseudo code with expected logic and approach.
		30	20	10
Execution / Output	50	For given concept, implementation successfully done.	For given concept, implementation partially done.	For given concept, implementation still in process towards the goal.
		50	40	35
Viva Voce	20	Student answered all the viva voce questions which includes analytical skills	Student answered to all viva voce questions	Student answered to partial viva voce questions
		20	16	10

Rubrics for Semester End Assessment (Total 100 marks)

Experiment write Up : 35 marks

Results : 35 marks

Viva-Voce : 30 marks

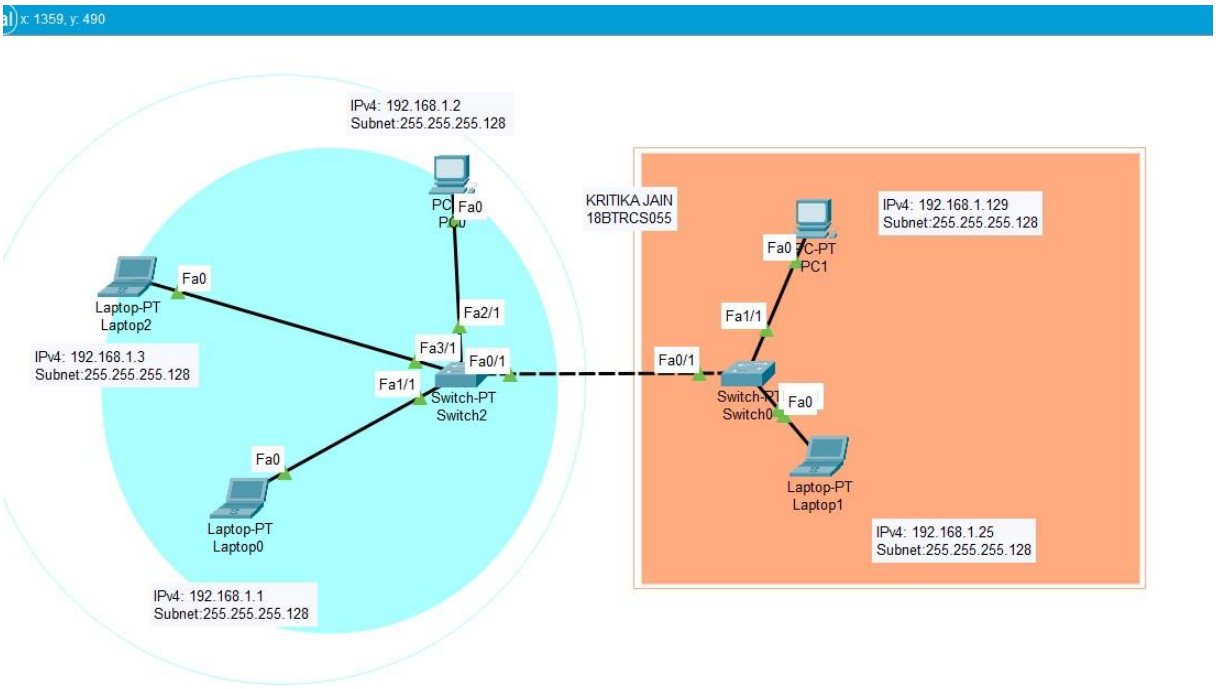
List of Experiment with CO-Mapping

SL No	Title of the Experiment	Course Outcomes
1.	Initialize the Network and Configure a Subnet	
2.	Sketching the Network Topology and Networking Components and devices: LAN Adapters, Hubs, Switches, Routers	
3.	Implementation of file and printer sharing and designing and implementing Class A, B, C Networks	
4.	Subnet planning and its implementation	
5.	TCP one way / two way communication and UPD one way /two way communication	
6.	Program in NS3 to connect two nodes	
7.	Program in NS3 for connecting three nodes considering one node as a central node.	
8.	Program in NS3 to implement star topology and bus topology	
9.	Program in NS3 for connecting multiple routers and nodes and building a hybrid topology.	
10.	Installation and configuration of NetAnim	
11.	Program in NS3 to implement FTP using TCP bulk transfer.	
12.	Program in NS3 for connecting multiple routers and nodes and building a hybrid topology and then calculating network performance	
13.	To analyse network traces using wireshark software	

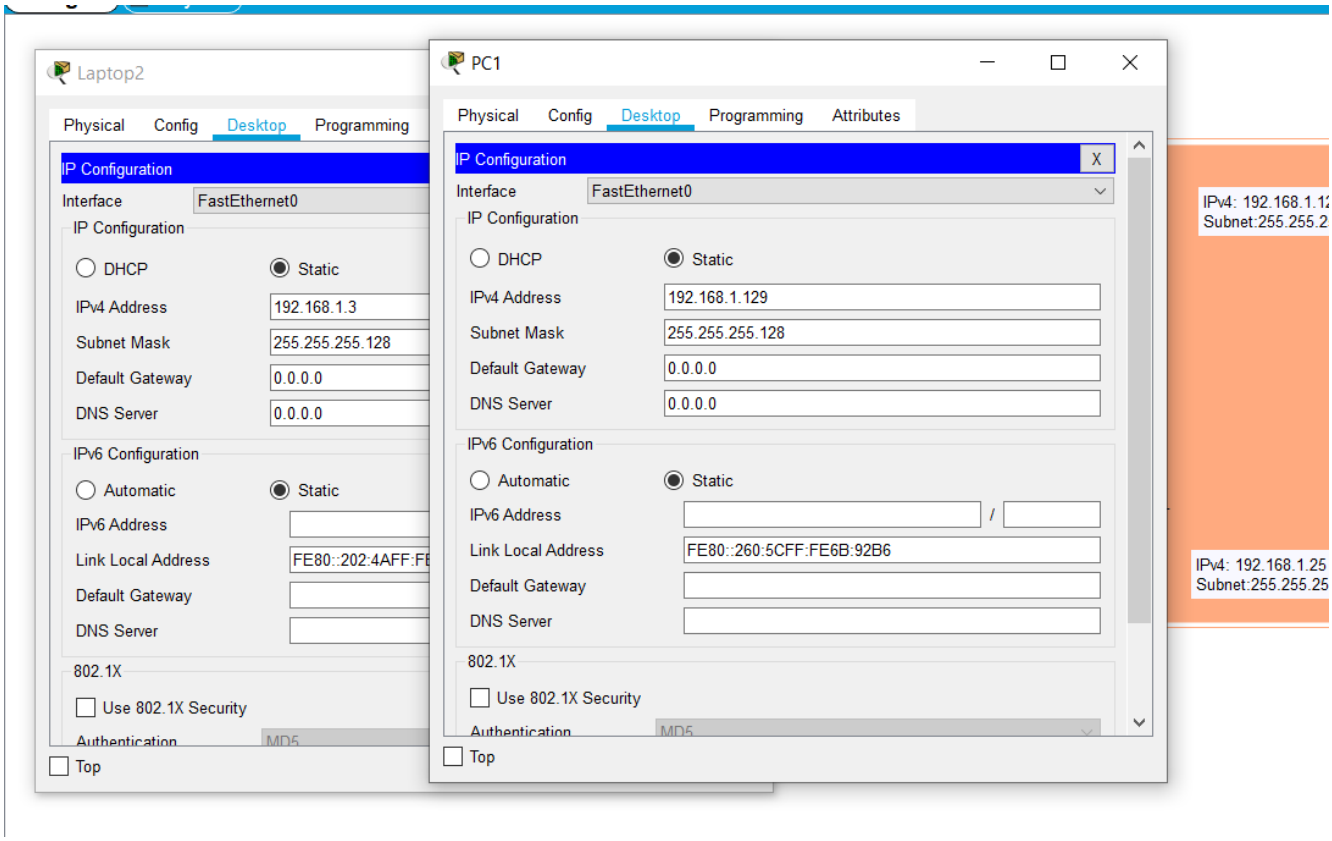
1. Initialize the network and configure a subnet.

TASK 1: Creating the network topology by dragging all the components in the logical workspace.

TOPOLOGY:

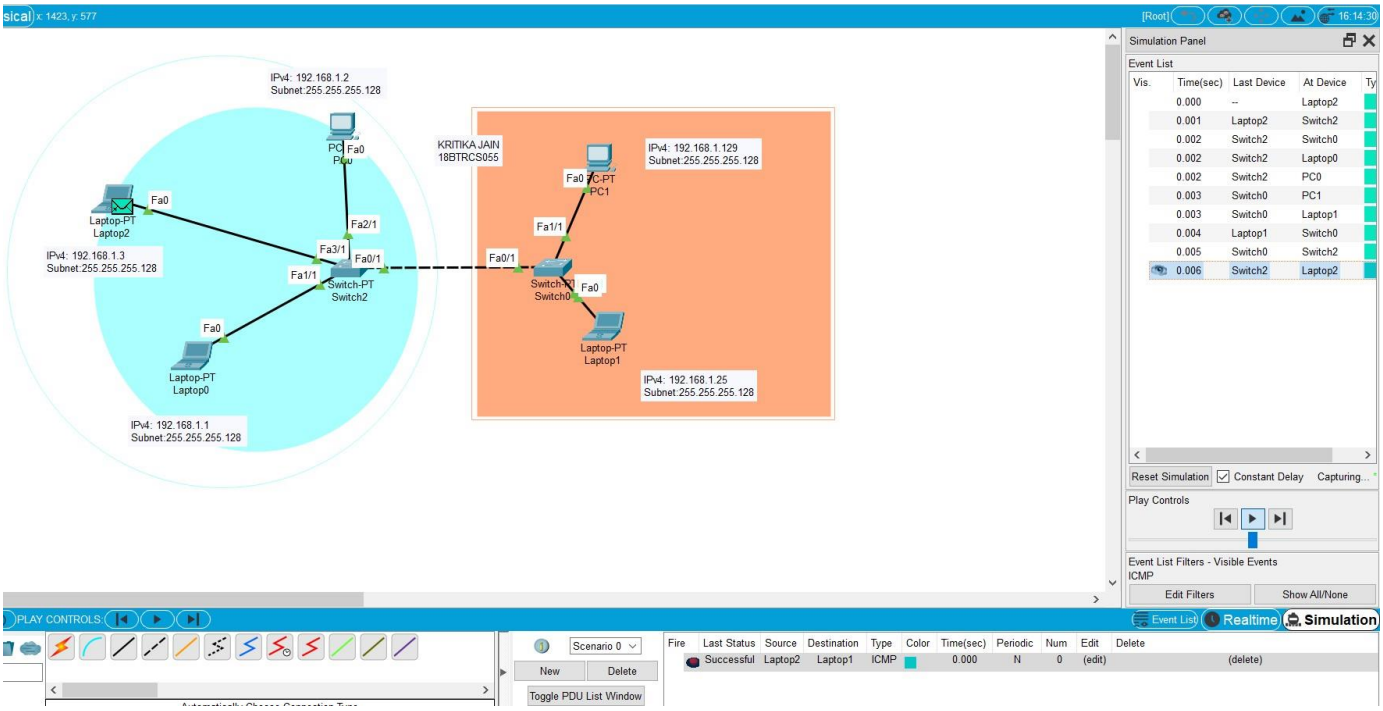


TASK 2: Configuring the network components by assigning IP addresses to all.



TASK 3: Checking the connections via ping.

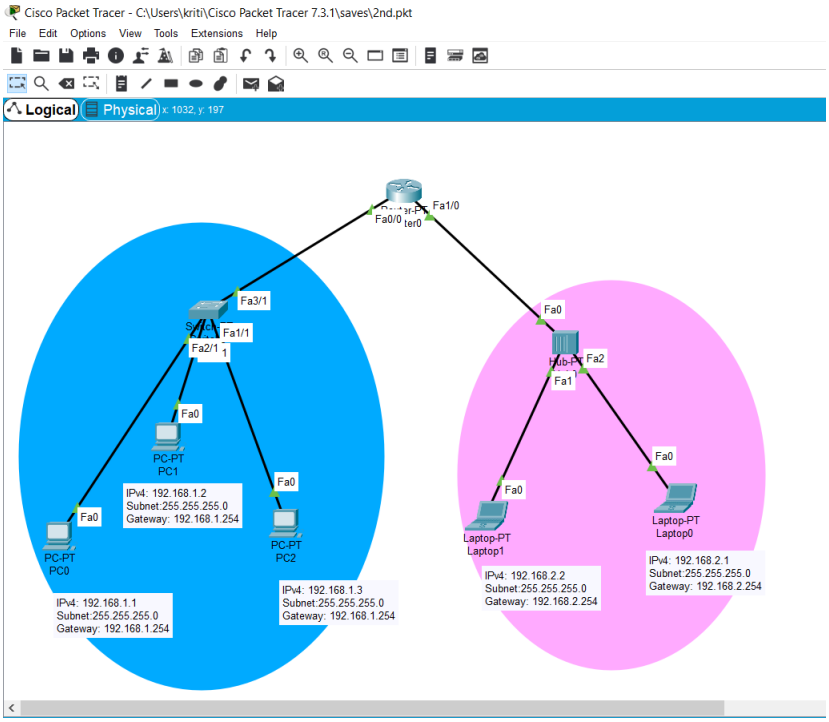
TASK 4: Sending and receiving a packet and observing in simulation mode.



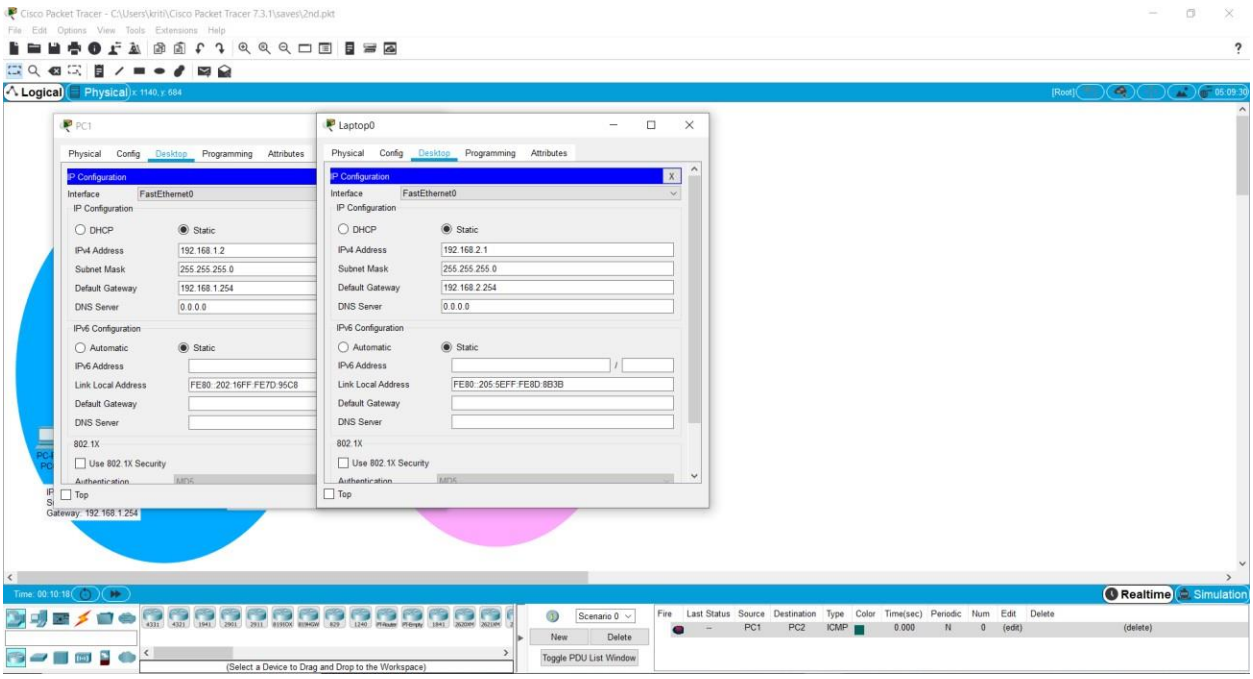
RESULT: The two networks were successfully configured with 255.255.255.128 subnet mask. Packet exchange simulation was also done successfully.

2. Sketching the network topology and networking components and devices: Lan Adapters, Hubs, Switches, Routers,etc.

TASK 1: Creating the network topology by dragging all the components in the logical workspace.

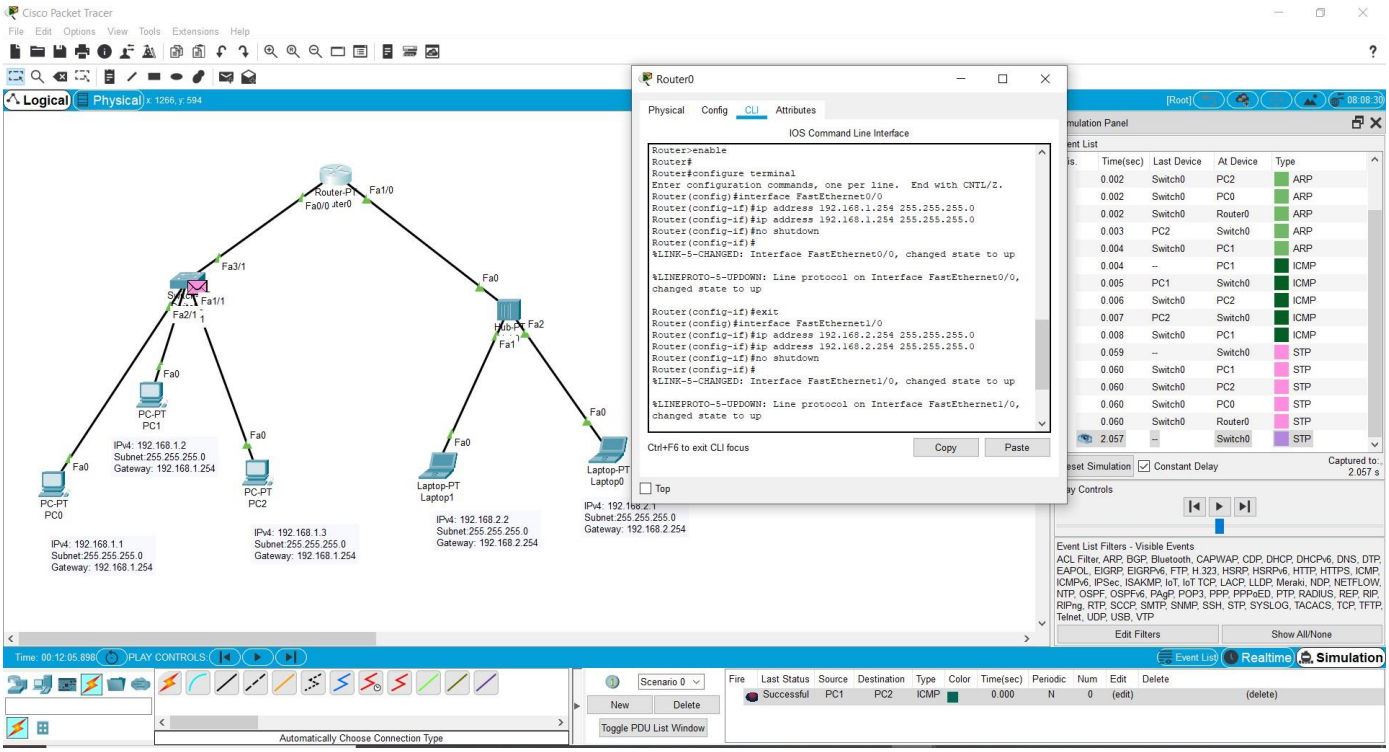


TASK 2: Configuring the network components by assigning IP addresses to all.



TASK 3: Configuring the router using CLI commands.

TASK 4: Sending and receiving a packet and observing in simulation mode.

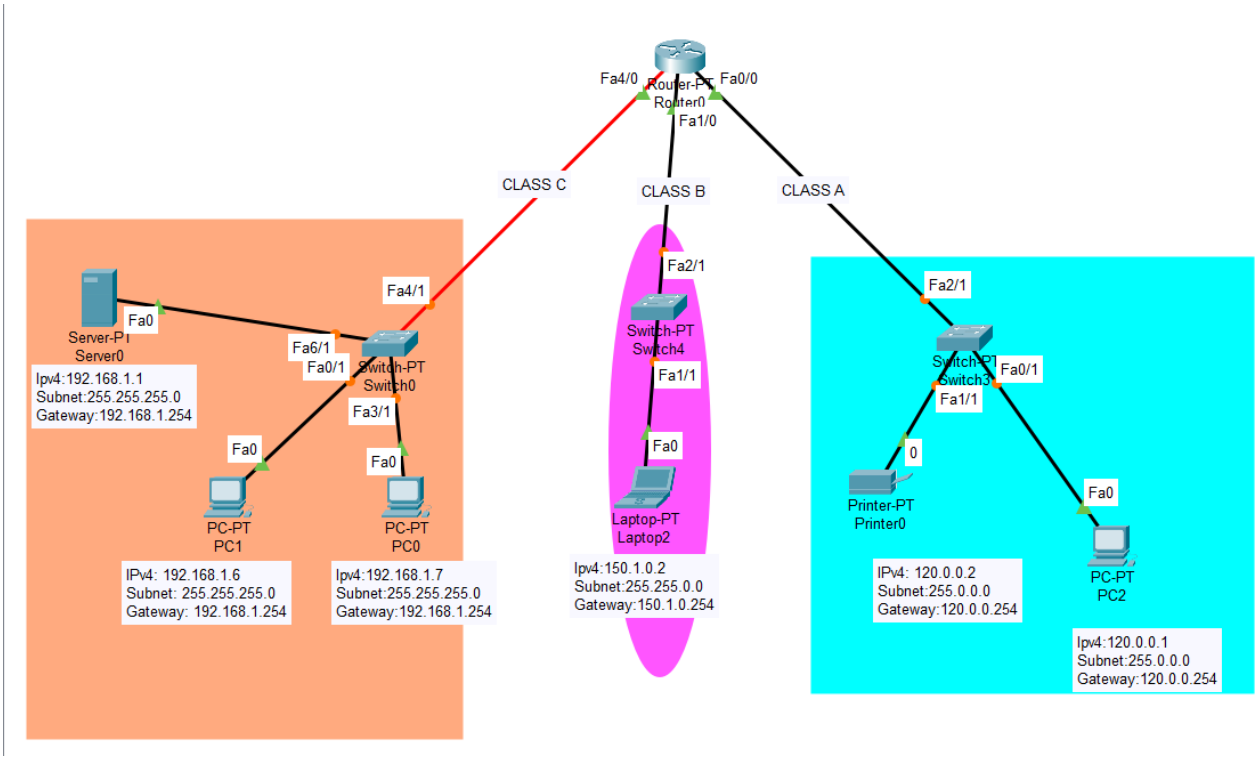


Result: The two networks were successfully configured with subnet masks and the ip addresses which were assigned accordingly. Packet exchange simulation was also done successfully.

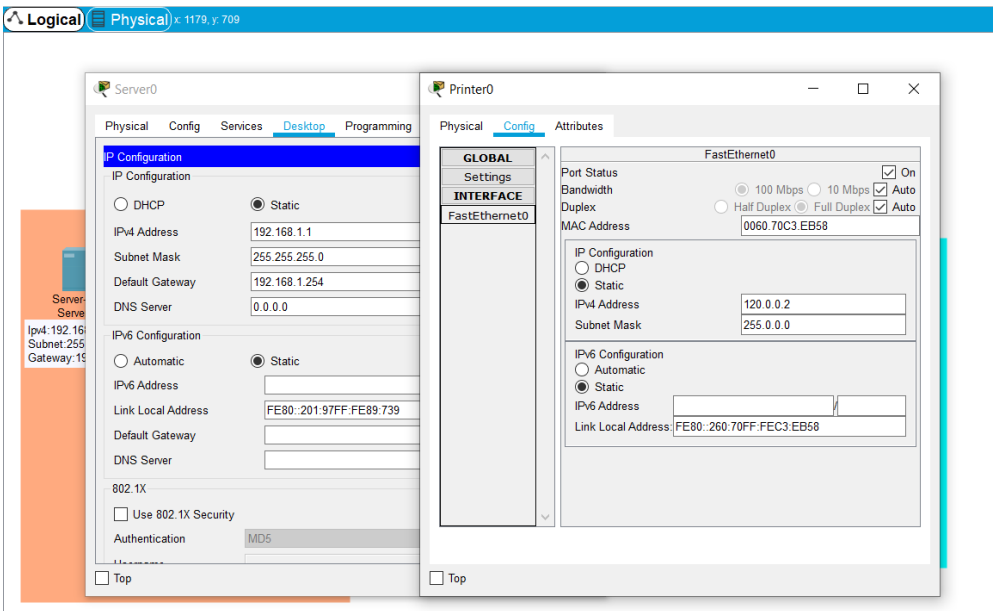
3. Implementation of file sharing and printer sharing and designing and implementing class A,B,C Networks.

TASK 1: Creating the network topology by dragging all the components in the logical workspace.

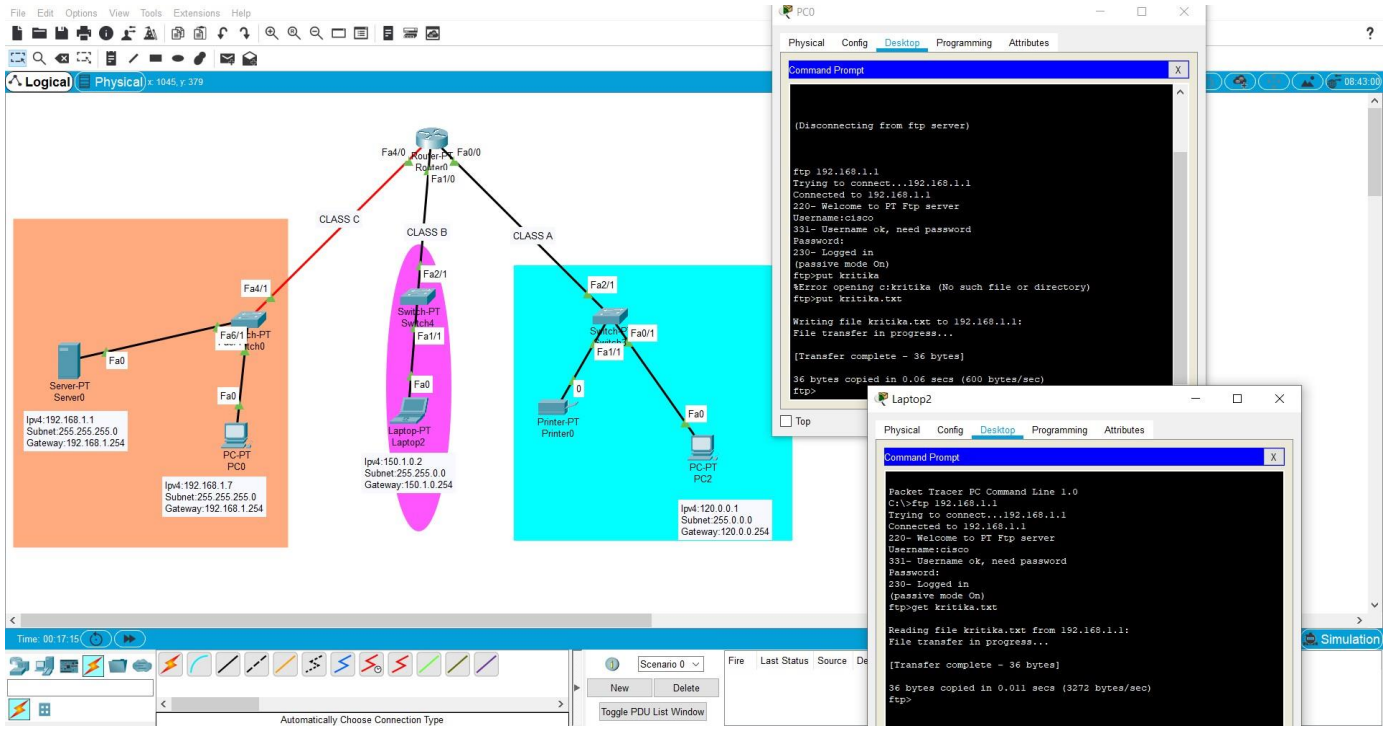
TOPOLOGY:



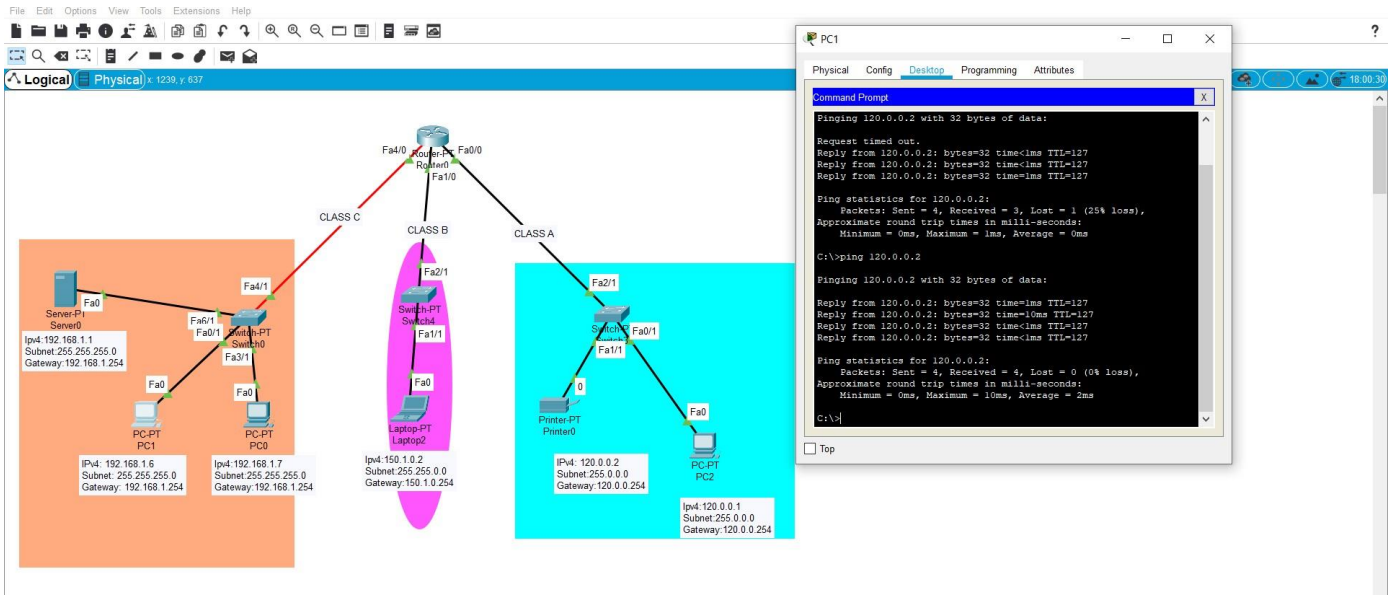
TASK 2: Configuring the network components by assigning IP addresses to all.



TASK 3: Sharing and downloading files from the FTP server.



TASK 4: Pinging the printer.

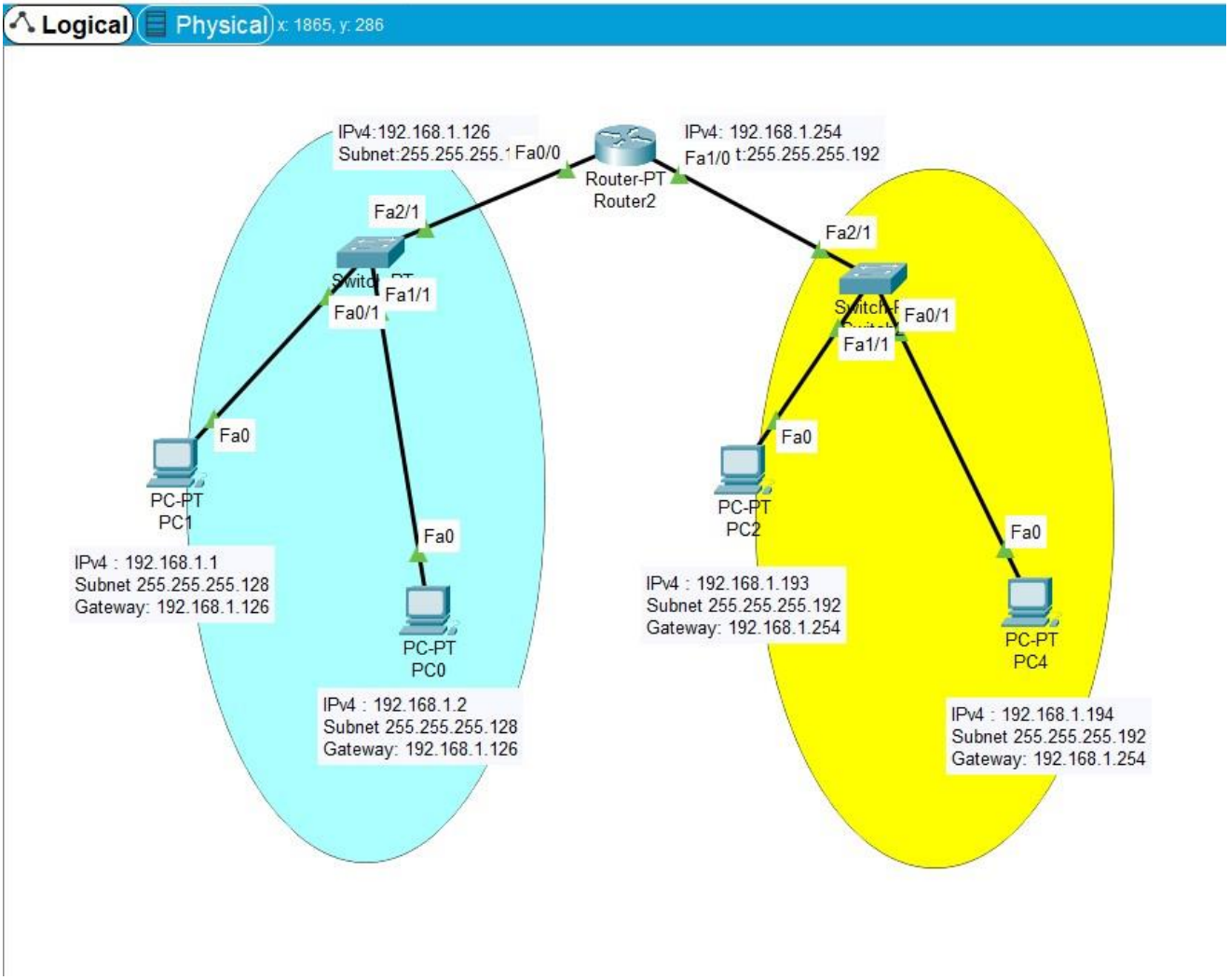


Result: File and printer sharing was successful between Class A,B and C networks.

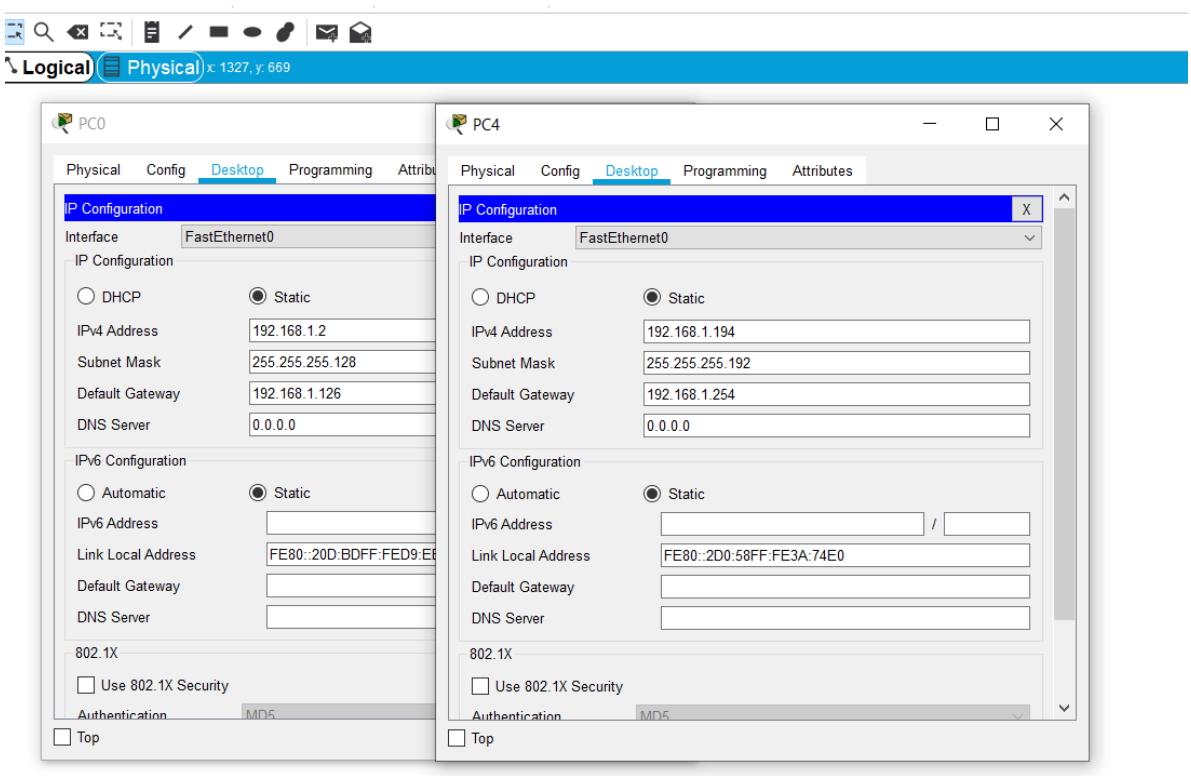
4. Subnet planning and its implementation.

TASK 1: Creating the network topology by dragging all the components in the logical workspace.

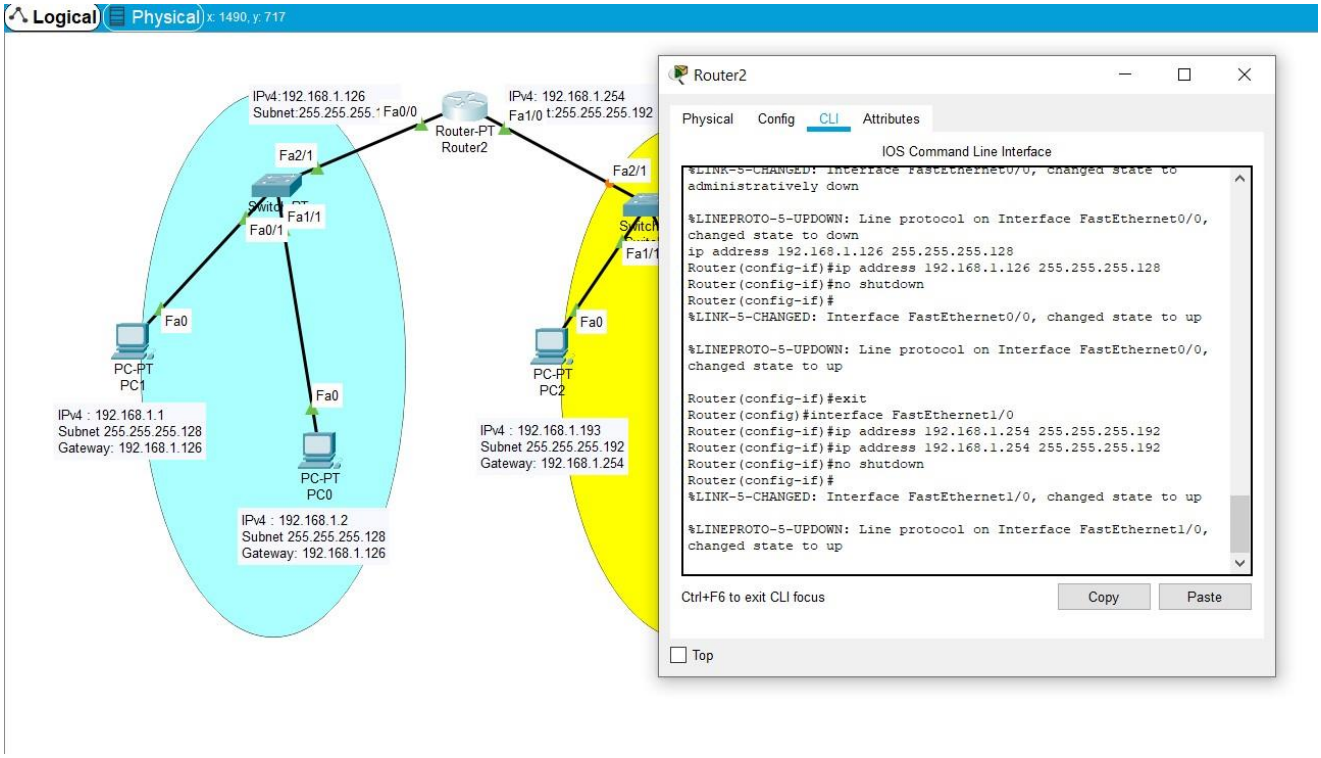
TOPOLOGY:



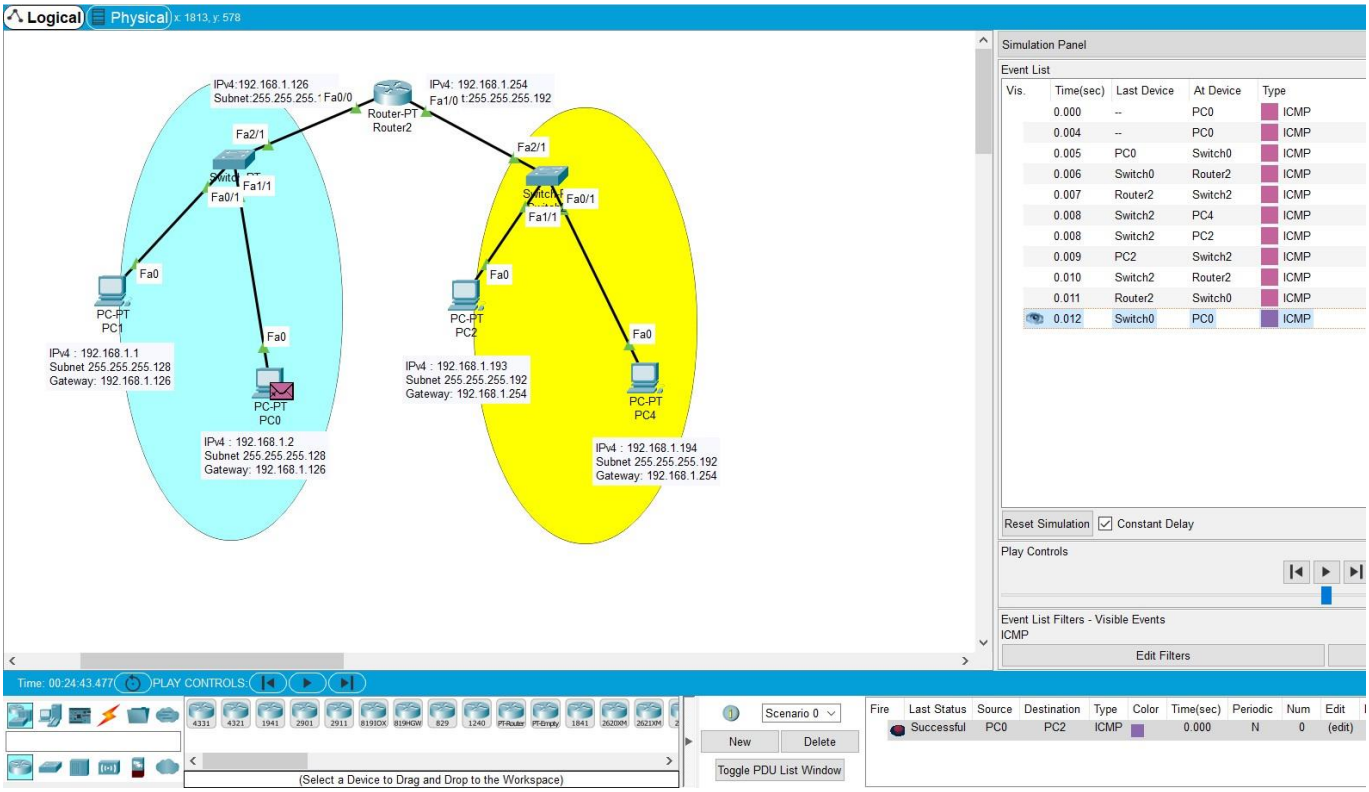
TASK 2: Configuring the network components by assigning IP addresses to all.



TASK 3: Configuring the router using CLI commands.



TASK 4: Sending and receiving a packet and observing in simulation mode.



Result: The two networks were successfully configured with subnet masks and the ip addresses were assigned accordingly. Packet exchange simulation was also done successfully.

5. TCP one way / two-way communication and UDP one way / two-way communication.

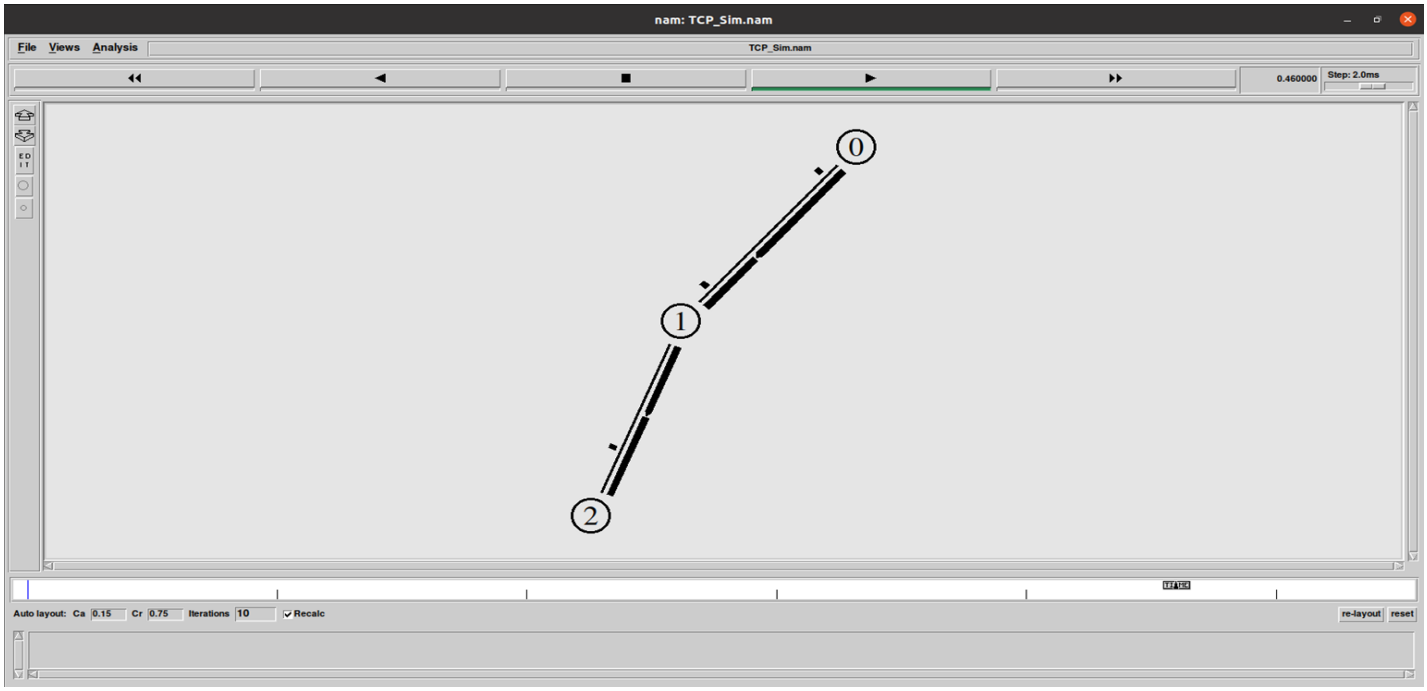
TCP:

Program:

```
#TCP one/two-way communication

# CREATE A SIMULATOR OBJECT
set ns [new Simulator]

# CREATE ANIMATION FILE OR NAM FILE
set nf [open TCP_Sim.nam w]
$ns namtrace-all $nf

# CREATE A TRACEFILE -- THIS IS JUST FOR LOGGING PURPOSE
set tf [open TCP_Trace.tr w]
$ns trace-all $tf

#CREATE NODES
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

# CREATE LINKS BETWEEN NODES WITH DROPTAIL QUEUE
# DROPTAIL MEANS DROPPING THE TAIL
$ns duplex-link $n0 $n1 10Mb 1ms DropTail
$ns duplex-link $n1 $n2 10Mb 1ms DropTail
```

```
#CREATION OF TCP AGENT
# tcp for sender
# sink for receiver
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp

set sink0 [new Agent/TCPSink]
$ns attach-agent $n2 $sink0

# CREATION OF APPLICATION -- CBR AND FTP
# FTP - File Transfer Protocol (Ex: Downloading a file from a network)
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp
$ns connect $tcp $sink0
# Start the traffic
$ns at 0.2 "$ftp0 start"
$ns at 0.5 "finish"

# FINISH PROCESURE: The following procedure is called at 0.5 seconds
proc finish { } {
global ns nf tf
$ns flush-trace
close  $nf
close $tf
exec nam TCP_Sim.nam &
exit 0  }
$ns run
```

Output:



Trace file:

Open
TCP_Trace.tr
~/DCN/P5
TCP.tcl
TCP_Trace.tr

```
 1 + 0.2 0 1 tcp 40 ------- 0 0.0 2.0 0 0
 2 - 0.2 0 1 tcp 40 ------- 0 0.0 2.0 0 0
 3 r 0.201032 0 1 tcp 40 ------- 0 0.0 2.0 0 0
 4 + 0.201032 1 2 tcp 40 ------- 0 0.0 2.0 0 0
 5 - 0.201032 1 2 tcp 40 ------- 0 0.0 2.0 0 0
 6 r 0.202064 1 2 tcp 40 ------- 0 0.0 2.0 0 0
 7 + 0.202064 2 1 ack 40 ------- 0 2.0 0.0 0 1
 8 - 0.202064 2 1 ack 40 ------- 0 2.0 0.0 0 1
 9 r 0.203096 2 1 ack 40 ------- 0 2.0 0.0 0 1
10 + 0.203096 1 0 ack 40 ------- 0 2.0 0.0 0 1
11 - 0.203096 1 0 ack 40 ------- 0 2.0 0.0 0 1
12 r 0.204128 1 0 ack 40 ------- 0 2.0 0.0 0 1
13 + 0.204128 0 1 tcp 1040 ------- 0 0.0 2.0 1 2
14 - 0.204128 0 1 tcp 1040 ------- 0 0.0 2.0 1 2
15 + 0.204128 0 1 tcp 1040 ------- 0 0.0 2.0 2 3
16 - 0.20496 0 1 tcp 1040 ------- 0 0.0 2.0 2 3
17 r 0.20596 0 1 tcp 1040 ------- 0 0.0 2.0 1 2
18 + 0.20596 1 2 tcp 1040 ------- 0 0.0 2.0 1 2
19 - 0.20596 1 2 tcp 1040 ------- 0 0.0 2.0 1 2
20 r 0.206792 0 1 tcp 1040 ------- 0 0.0 2.0 2 3
21 + 0.206792 1 2 tcp 1040 ------- 0 0.0 2.0 2 3
22 - 0.206792 1 2 tcp 1040 ------- 0 0.0 2.0 2 3
23 r 0.207792 1 2 tcp 1040 ------- 0 0.0 2.0 1 2
24 + 0.207792 2 1 ack 40 ------- 0 2.0 0.0 1 4
25 - 0.207792 2 1 ack 40 ------- 0 2.0 0.0 1 4
26 r 0.208624 1 2 tcp 1040 ------- 0 0.0 2.0 2 3
27 + 0.208624 2 1 ack 40 ------- 0 2.0 0.0 2 5
28 - 0.208624 2 1 ack 40 ------- 0 2.0 0.0 2 5
29 r 0.208824 2 1 ack 40 ------- 0 2.0 0.0 1 4
30 + 0.208824 1 0 ack 40 ------- 0 2.0 0.0 1 4
31 - 0.208824 1 0 ack 40 ------- 0 2.0 0.0 1 4
32 r 0.209656 2 1 ack 40 ------- 0 2.0 0.0 2 5
33 + 0.209656 1 0 ack 40 ------- 0 2.0 0.0 2 5
34 - 0.209656 1 0 ack 40 ------- 0 2.0 0.0 2 5
35 r 0.209856 1 0 ack 40 ------- 0 2.0 0.0 1 4
36 + 0.209856 0 1 tcp 1040 ------- 0 0.0 2.0 3 6
37 - 0.209856 0 1 tcp 1040 ------- 0 0.0 2.0 3 6
38 + 0.209856 0 1 tcp 1040 ------- 0 0.0 2.0 4 7
39 r 0.210688 1 0 ack 40 ------- 0 2.0 0.0 2 5
40 + 0.210688 0 1 tcp 1040 ------- 0 0.0 2.0 5 8
41 + 0.210688 0 1 tcp 1040 ------- 0 0.0 2.0 6 9
42 - 0.210688 0 1 tcp 1040 ------- 0 0.0 2.0 4 7
```

Plain Text Tat

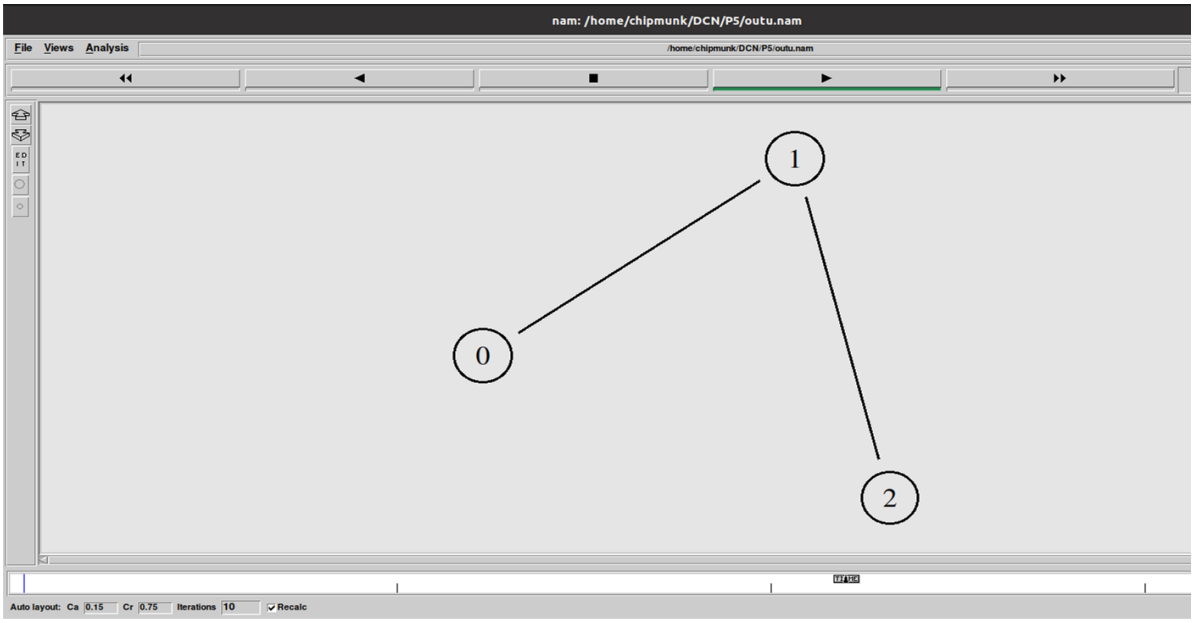
UDP:

Program:

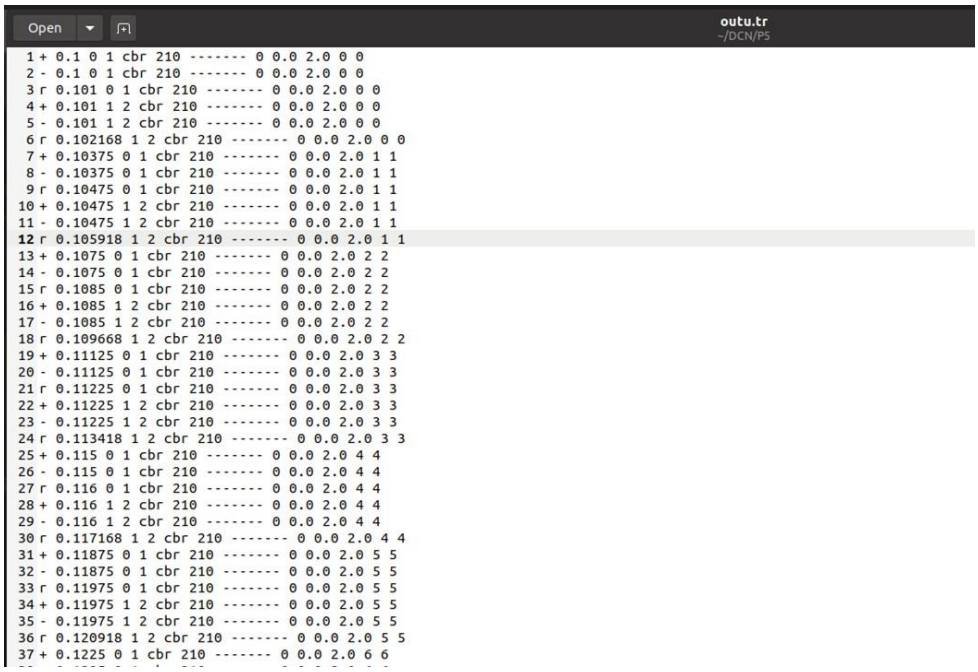
```
#CREATE NS OBJECT
set ns [new Simulator]

#CREATE NAM FILE
set nf [open outu.nam w]
$ns namtrace-all $nf

#CREATE TRACE FILE
set tf [open outu.tr w]
$ns trace-all $tf

#CREATE NODES
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

#CREATE LINKS
$ns duplex-link $n0 $n1 100000Mb 1ms DropTail
$ns duplex-link $n1 $n2 10Mb 1ms DropTail

#CREATE UDP AGENT AND NULL AGENT
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp

set null [new Agent/Null]
$ns attach-agent $n2 $null

set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set interval 0.005
```

```
$cbr set packetSize 50000
$ns connect $udp $null

#FINISH PROCEDURE
proc finish { } {
global ns nf tf
$ns flush-trace
close  $nf
close $tf
exec nam outu.nam &
exec gedit outu.tr &
exit 0
}

#START THE TRAFFIC AND RUN
$ns at 0.1 "$cbr start"
$ns at 0.3 "finish"
$ns run
```

Output:



Trace file:

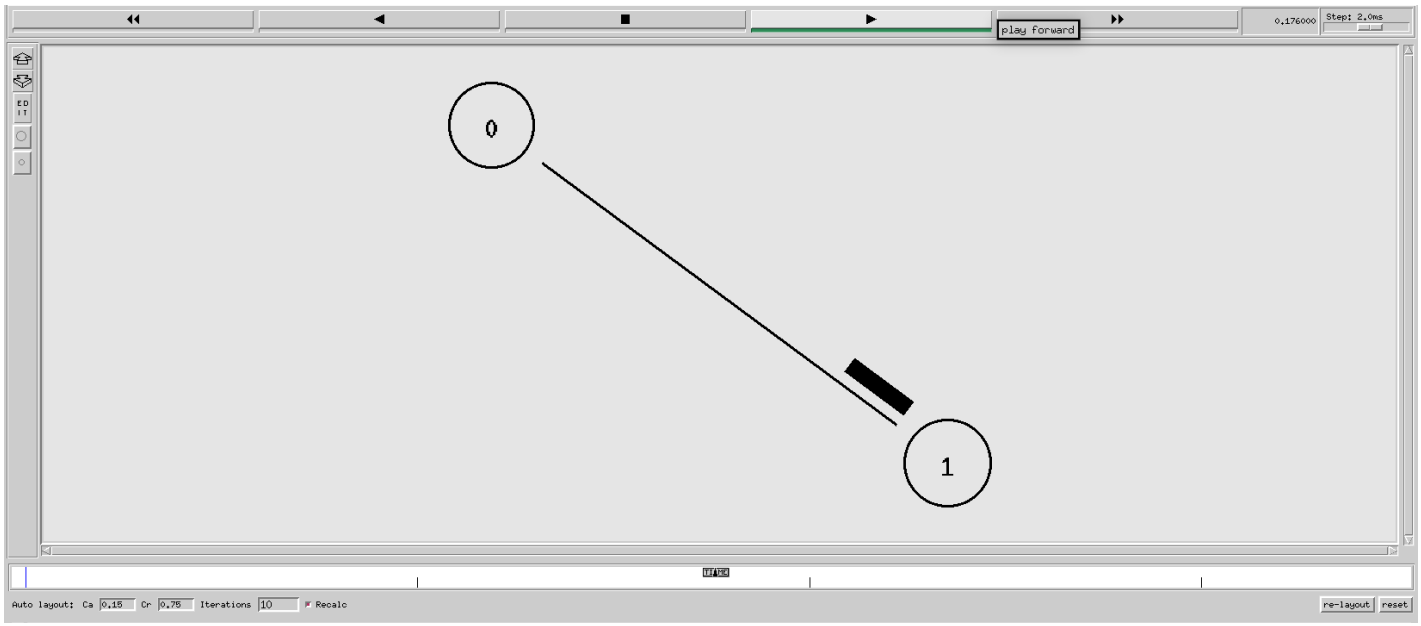


6. Program in NS3 to connect two nodes.

```
#CREATE NS OBJECT
set ns [new Simulator]

#CREATE NAM FILE
set nf [open out.nam w]
$ns namtrace-all $nf

#CREATE TRACE FILE
set tf [open out.tr w]
$ns trace-all $tf

#CREATE NODES
set n0 [$ns node]
set n1 [$ns node]

#CREATE LINKS

$ns duplex-link $n0 $n1 10Mb 1ms DropTail

#CREATE UDP AGENT AND NULL AGENT
```

```
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp

set null [new Agent/Null]
$ns attach-agent $n1 $null

set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp

$cbr set interval 0.005
$cbr set packetSize 50000

$ns connect $udp $null

#FINISH PROCEDURE
proc finish { } {
global ns nf tf
$ns flush-trace
close  $nf
close $tf
exec nam out.nam &
exec gedit out.tr &
exit 0
}
#START THE TRAFFIC AND RUN
$ns at 0.1 "$cbr start"
$ns at 0.3 "finish"
$ns run
```

Output:



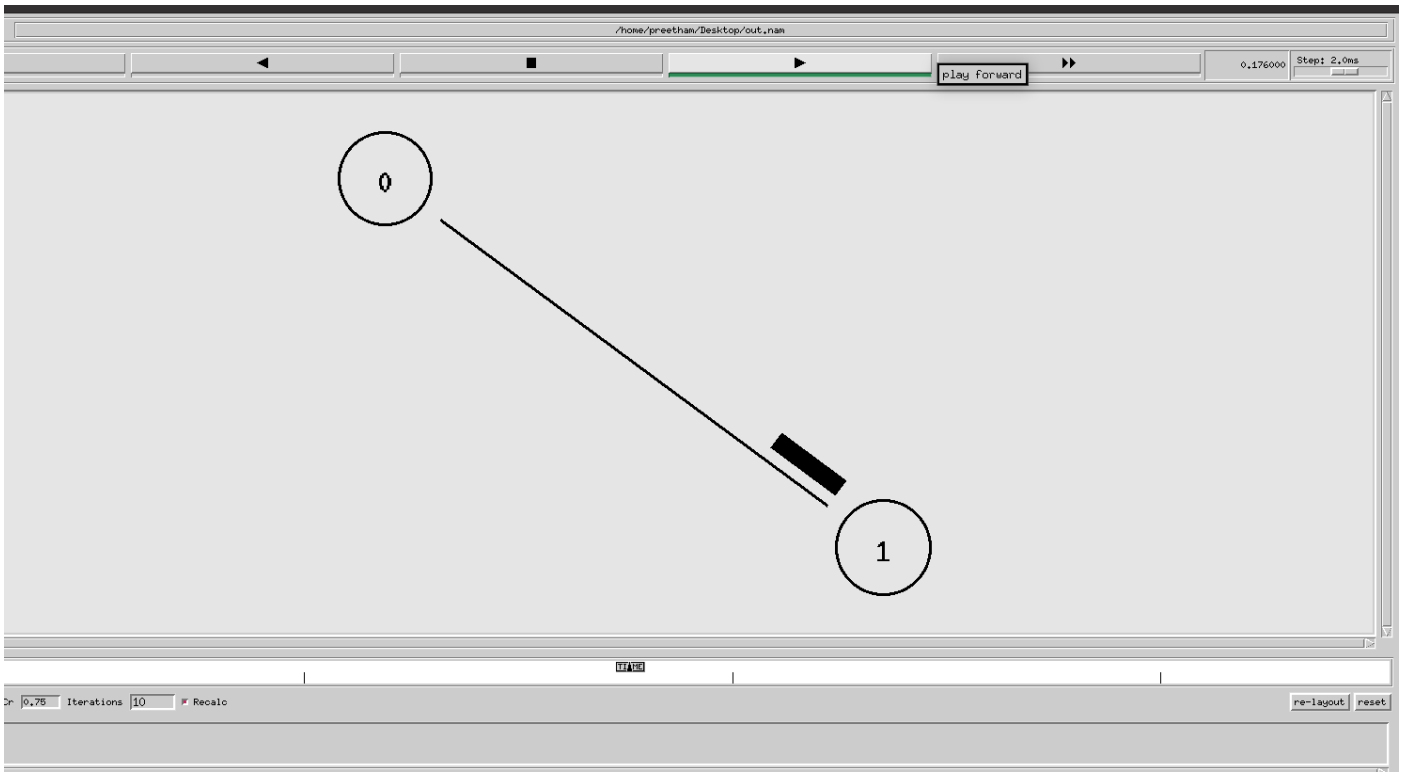

```
10 + 0.11125 0 1 cbr 210 ------- 0 0.0 1.0 3 3
11 - 0.11125 0 1 cbr 210 ------- 0 0.0 1.0 3 3
12 r 0.112418 0 1 cbr 210 ------- 0 0.0 1.0 3 3
13 + 0.115 0 1 cbr 210 ------- 0 0.0 1.0 4 4
14 - 0.115 0 1 cbr 210 ------- 0 0.0 1.0 4 4
15 r 0.116168 0 1 cbr 210 ------- 0 0.0 1.0 4 4
16 + 0.11875 0 1 cbr 210 ------- 0 0.0 1.0 5 5
17 - 0.11875 0 1 cbr 210 ------- 0 0.0 1.0 5 5
18 r 0.119918 0 1 cbr 210 ------- 0 0.0 1.0 5 5
19 + 0.1225 0 1 cbr 210 ------- 0 0.0 1.0 6 6
20 - 0.1225 0 1 cbr 210 ------- 0 0.0 1.0 6 6
21 r 0.123668 0 1 cbr 210 ------- 0 0.0 1.0 6 6
22 + 0.12625 0 1 cbr 210 ------- 0 0.0 1.0 7 7
23 - 0.12625 0 1 cbr 210 ------- 0 0.0 1.0 7 7
24 r 0.127418 0 1 cbr 210 ------- 0 0.0 1.0 7 7
25 + 0.13 0 1 cbr 210 ------- 0 0.0 1.0 8 8
26 - 0.13 0 1 cbr 210 ------- 0 0.0 1.0 8 8
27 r 0.131168 0 1 cbr 210 ------- 0 0.0 1.0 8 8
28 + 0.13375 0 1 cbr 210 ------- 0 0.0 1.0 9 9
29 - 0.13375 0 1 cbr 210 ------- 0 0.0 1.0 9 9
30 r 0.134918 0 1 cbr 210 ------- 0 0.0 1.0 9 9
31 + 0.1375 0 1 cbr 210 ------- 0 0.0 1.0 10 10
32 - 0.1375 0 1 cbr 210 ------- 0 0.0 1.0 10 10
33 r 0.138668 0 1 cbr 210 ------- 0 0.0 1.0 10 10
34 + 0.14125 0 1 cbr 210 ------- 0 0.0 1.0 11 11
35 - 0.14125 0 1 cbr 210 ------- 0 0.0 1.0 11 11
36 r 0.142418 0 1 cbr 210 ------- 0 0.0 1.0 11 11
37 + 0.145 0 1 cbr 210 ------- 0 0.0 1.0 12 12
38 - 0.145 0 1 cbr 210 ------- 0 0.0 1.0 12 12
39 r 0.146168 0 1 cbr 210 ------- 0 0.0 1.0 12 12
40 + 0.14875 0 1 cbr 210 ------- 0 0.0 1.0 13 13
41 - 0.14875 0 1 cbr 210 ------- 0 0.0 1.0 13 13
```

7. Program in NS3 for connecting three nodes considering one node as central node.

```
#CREATE NS OBJECT
set ns [new Simulator]

#CREATE NAM FILE
set nf [open out.nam w]
$ns namtrace-all $nf

#CREATE TRACE FILE
set tf [open out.tr w]
$ns trace-all $tf

#CREATE NODES
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

#CREATE LINKS
$ns duplex-link $n0 $n1 10Mb 1ms DropTail
$ns duplex-link $n1 $n2 10Mb 1ms DropTail

#CREATE UDP AGENT AND NULL AGENT
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp

set null [new Agent/Null]
$ns attach-agent $n2 $null

set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp

$cbr set interval 0.005
$cbr set packetSize 50000

$ns connect $udp $null

#CREATION OF TCP AGENT
# tcp for sender
```

```
# sink for receiver
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp

set sink0 [new Agent/TCPSink]
$ns attach-agent $n2 $sink0

# CREATION OF APPLICATION -- CBR AND FTP
# FTP - File Transfer Protocol (Ex: Downloading a file from a network)
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp

$ns connect $tcp $sink0

# Start the traffic
$ns at 0.2 "$ftp0 start"
$ns at 0.5 "finish"

#FINISH PROCEDURE
proc finish { } {
global ns nf tf
$ns flush-trace
close  $nf
close $tf
exec nam out.nam &
exec gedit out.tr &
exit 0
}
#START THE TRAFFIC AND RUN
$ns at 0.1 "$cbr start"
$ns at 0.3 "finish"

$ns run
```

Output:



Trace file:

p6.tcl

```
1 + 0.1 0 1 cbr 210 ------- 0 0.0 1.0 0 0
2 - 0.1 0 1 cbr 210 ------- 0 0.0 1.0 0 0
3 r 0.101168 0 1 cbr 210 ------- 0 0.0 1.0 0 0
4 + 0.10375 0 1 cbr 210 ------- 0 0.0 1.0 1 1
5 - 0.10375 0 1 cbr 210 ------- 0 0.0 1.0 1 1
6 r 0.104918 0 1 cbr 210 ------- 0 0.0 1.0 1 1
7 + 0.1075 0 1 cbr 210 ------- 0 0.0 1.0 2 2
8 - 0.1075 0 1 cbr 210 ------- 0 0.0 1.0 2 2
9 r 0.108668 0 1 cbr 210 ------- 0 0.0 1.0 2 2
10 + 0.11125 0 1 cbr 210 ------- 0 0.0 1.0 3 3
11 - 0.11125 0 1 cbr 210 ------- 0 0.0 1.0 3 3
12 r 0.112418 0 1 cbr 210 ------- 0 0.0 1.0 3 3
13 + 0.115 0 1 cbr 210 ------- 0 0.0 1.0 4 4
14 - 0.115 0 1 cbr 210 ------- 0 0.0 1.0 4 4
15 r 0.116168 0 1 cbr 210 ------- 0 0.0 1.0 4 4
16 + 0.11875 0 1 cbr 210 ------- 0 0.0 1.0 5 5
17 - 0.11875 0 1 cbr 210 ------- 0 0.0 1.0 5 5
18 r 0.119918 0 1 cbr 210 ------- 0 0.0 1.0 5 5
19 + 0.1225 0 1 cbr 210 ------- 0 0.0 1.0 6 6
20 - 0.1225 0 1 cbr 210 ------- 0 0.0 1.0 6 6
21 r 0.123668 0 1 cbr 210 ------- 0 0.0 1.0 6 6
22 + 0.12625 0 1 cbr 210 ------- 0 0.0 1.0 7 7
23 - 0.12625 0 1 cbr 210 ------- 0 0.0 1.0 7 7
24 r 0.127418 0 1 cbr 210 ------- 0 0.0 1.0 7 7
25 + 0.13 0 1 cbr 210 ------- 0 0.0 1.0 8 8
26 - 0.13 0 1 cbr 210 ------- 0 0.0 1.0 8 8
27 r 0.131168 0 1 cbr 210 ------- 0 0.0 1.0 8 8
28 + 0.13375 0 1 cbr 210 ------- 0 0.0 1.0 9 9
29 - 0.13375 0 1 cbr 210 ------- 0 0.0 1.0 9 9
30 r 0.134918 0 1 cbr 210 ------- 0 0.0 1.0 9 9
31 + 0.1375 0 1 cbr 210 ------- 0 0.0 1.0 10 10
32 - 0.1375 0 1 cbr 210 ------- 0 0.0 1.0 10 10
33 r 0.138668 0 1 cbr 210 ------- 0 0.0 1.0 10 10
34 + 0.14125 0 1 cbr 210 ------- 0 0.0 1.0 11 11
35 - 0.14125 0 1 cbr 210 ------- 0 0.0 1.0 11 11
36 r 0.142418 0 1 cbr 210 ------- 0 0.0 1.0 11 11
37 + 0.145 0 1 cbr 210 ------- 0 0.0 1.0 12 12
38 - 0.145 0 1 cbr 210 ------- 0 0.0 1.0 12 12
39 r 0.146168 0 1 cbr 210 ------- 0 0.0 1.0 12 12
40 + 0.14875 0 1 cbr 210 ------- 0 0.0 1.0 13 13
41 - 0.14875 0 1 cbr 210 ------- 0 0.0 1.0 13 13
```

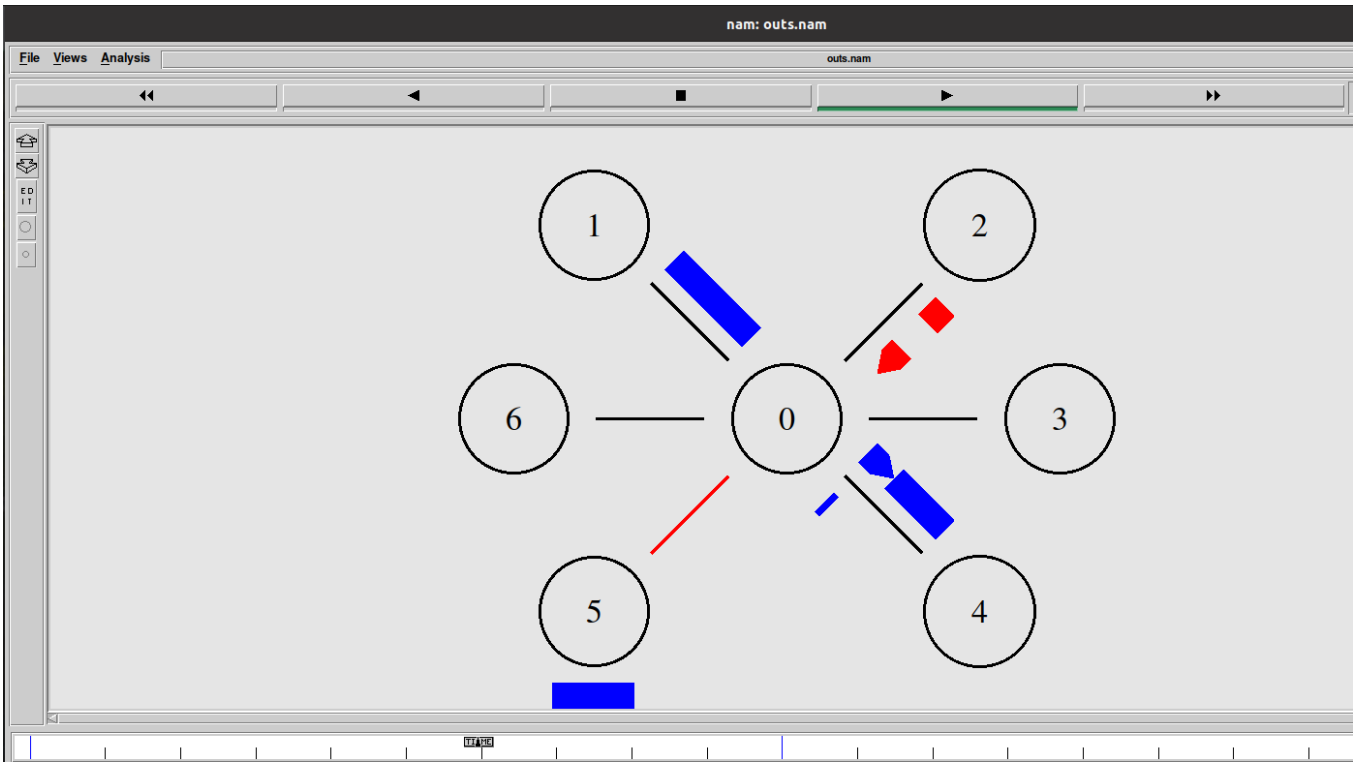
8. Program in NS3 to implement star topology and ring topology.

Star Topology Program:

```
set ns  [new Simulator]

$ns color 1 blue
$ns color 2 red

$ns rtproto DV

set nf [open outs.nam w]
$ns namtrace-all $nf

set nf [open outs.tr w]
$ns trace-all $nf

proc finish {} {
        global ns nf
        $ns flush-trace
        close $nf
        exec nam outs.nam
        exec gedit outs.tr
        exit 0
        }

#creating Nodes
for {set i 0} {$i<7} {incr i} {
set n($i) [$ns node]
}

#Creating Links
for {set i 1} {$i<7} {incr i} {
$ns duplex-link $n(0) $n($i) 512Kb 10ms SFQ
}

#Orienting The nodes
$ns duplex-link-op $n(0) $n(1) orient left-up
$ns duplex-link-op $n(0) $n(2) orient right-up
$ns duplex-link-op $n(0) $n(3) orient right
$ns duplex-link-op $n(0) $n(4) orient right-down
$ns duplex-link-op $n(0) $n(5) orient left-down
$ns duplex-link-op $n(0) $n(6) orient left

#TCP_Config
set tcp0 [new Agent/TCP]
```

```
$tcp0 set class_ 1
$ns attach-agent $n(1) $tcp0

set sink0 [new Agent/TCPSink]
$ns attach-agent $n(4) $sink0

$ns connect $tcp0 $sink0

#UDP_Config
set udp0 [new Agent/UDP]
$udp0 set class_ 2
$ns attach-agent $n(2) $udp0

set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0

$ns connect $udp0 $null0

#CBR Config
set cbr0 [new Application/Traffic/CBR]
$cbr0 set rate_ 256Kb
$cbr0 attach-agent $udp0

#FTP Config
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0


#Scheduling Events
$ns rtmodel-at 0.5 down $n(0) $n(5)
$ns rtmodel-at 0.9 up $n(0) $n(5)

$ns rtmodel-at 0.7 down $n(0) $n(4)
$ns rtmodel-at 1.2 up $n(0) $n(4)

$ns at 0.1 "$ftp0 start"
$ns at 1.5 "$ftp0 stop"

$ns at 0.2 "$cbr0 start"
$ns at 1.3 "$cbr0 stop"

$ns at 2.0 "finish"
$ns run
```

Output for star:



Open

outs.tr
~/DCN

star.tcl

```
 99 - 0.221875 0 1 ack 40 ------- 1 4.0 1.0 2 22
100 + 0.22625 2 0 cbr 210 ------- 2 2.0 5.0 4 28
101 - 0.22625 2 0 cbr 210 ------- 2 2.0 5.0 4 28
102 r 0.226406 2 0 cbr 210 ------- 2 2.0 5.0 2 23
103 + 0.226406 0 5 cbr 210 ------- 2 2.0 5.0 2 23
104 - 0.226406 0 5 cbr 210 ------- 2 2.0 5.0 2 23
105 r 0.226563 0 5 cbr 210 ------- 2 2.0 5.0 0 20
106 r 0.230696 4 0 rtProtoDV 7 ------- 0 4.2 0.1 -1 27
107 - 0.2325 1 0 tcp 1040 ------- 1 1.0 4.0 4 25
108 r 0.2325 0 1 ack 40 ------- 1 4.0 1.0 2 22
109 + 0.2325 1 0 tcp 1040 ------- 1 1.0 4.0 5 29
110 + 0.2325 1 0 tcp 1040 ------- 1 1.0 4.0 6 30
111 + 0.232813 2 0 cbr 210 ------- 2 2.0 5.0 5 31
112 - 0.232813 2 0 cbr 210 ------- 2 2.0 5.0 5 31
113 r 0.232969 2 0 cbr 210 ------- 2 2.0 5.0 3 26
114 + 0.232969 0 5 cbr 210 ------- 2 2.0 5.0 3 26
115 - 0.232969 0 5 cbr 210 ------- 2 2.0 5.0 3 26
116 r 0.233125 0 5 cbr 210 ------- 2 2.0 5.0 1 21
117 + 0.239375 2 0 cbr 210 ------- 2 2.0 5.0 6 32
118 - 0.239375 2 0 cbr 210 ------- 2 2.0 5.0 6 32
119 r 0.239531 2 0 cbr 210 ------- 2 2.0 5.0 4 28
120 + 0.239531 0 5 cbr 210 ------- 2 2.0 5.0 4 28
121 - 0.239531 0 5 cbr 210 ------- 2 2.0 5.0 4 28
122 r 0.239688 0 5 cbr 210 ------- 2 2.0 5.0 2 23
123 r 0.2425 1 0 tcp 1040 ------- 1 1.0 4.0 3 24
124 + 0.2425 0 4 tcp 1040 ------- 1 1.0 4.0 3 24
125 - 0.2425 0 4 tcp 1040 ------- 1 1.0 4.0 3 24
126 + 0.245938 2 0 cbr 210 ------- 2 2.0 5.0 7 33
127 - 0.245938 2 0 cbr 210 ------- 2 2.0 5.0 7 33
128 r 0.246094 2 0 cbr 210 ------- 2 2.0 5.0 5 31
129 + 0.246094 0 5 cbr 210 ------- 2 2.0 5.0 5 31
130 - 0.246094 0 5 cbr 210 ------- 2 2.0 5.0 5 31
131 r 0.24625 0 5 cbr 210 ------- 2 2.0 5.0 3 26
132 - 0.24875 1 0 tcp 1040 ------- 1 1.0 4.0 5 29
133 + 0.2525 2 0 cbr 210 ------- 2 2.0 5.0 8 34
134 - 0.2525 2 0 cbr 210 ------- 2 2.0 5.0 8 34
135 r 0.252656 2 0 cbr 210 ------- 2 2.0 5.0 6 32
136 + 0.252656 0 5 cbr 210 ------- 2 2.0 5.0 6 32
137 - 0.252656 0 5 cbr 210 ------- 2 2.0 5.0 6 32
138 r 0.252813 0 5 cbr 210 ------- 2 2.0 5.0 4 28
139 r 0.25875 1 0 tcp 1040 ------- 1 1.0 4.0 4 25
140 + 0.25875 0 4 tcp 1040 ------- 1 1.0 4.0 4 25
```

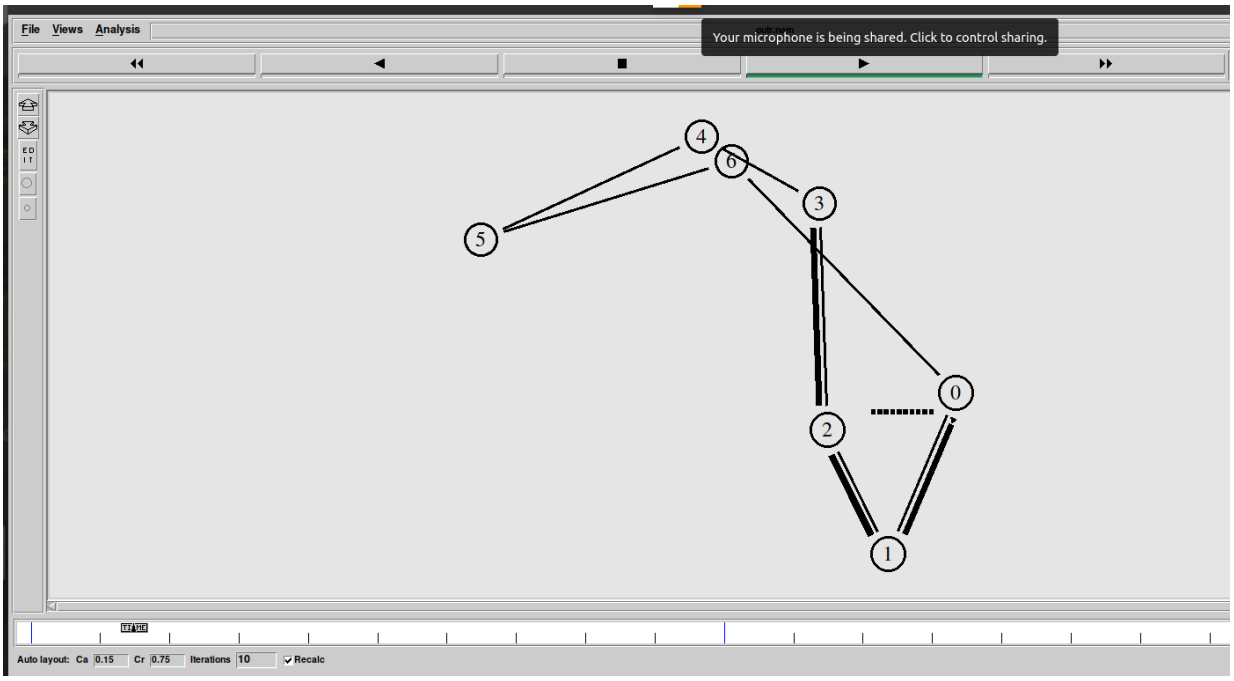
Ring Topology Program:

This Program will create a ring topology using less number of statements in TCL Language

```
set ns [new Simulator]
$ns rtproto DV

set nf [open outr.nam w]
$ns namtrace-all $nf

set nf [open outr.tr w]
$ns trace-all $nf

proc finish {} {
        global ns nf
        $ns flush-trace
        close $nf
        exec nam outr.nam
        exec gedit outr.tr
        exit 0
        }

#Creating Nodes
for {set i 0} {$i<7} {incr i} {
set n($i) [$ns node]
}

#Creating Links
for {set i 0} {$i<7} {incr i} {
$ns duplex-link $n($i) $n([expr ($i+1)%7]) 512Kb 5ms DropTail
}

$ns duplex-link-op $n(0) $n(1) queuePos 1
$ns duplex-link-op $n(0) $n(6) queuePos 1

#Creating UDP agent and attching to node 0
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
```

```
#Creating Null agent and attaching to node 3
set null0 [new Agent/Null]
$ns attach-agent $n(3) $null0

$ns connect $udp0 $null0


#Creating a CBR agent and attaching it to udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 1024
$cbr0 set interval_ 0.01
$cbr0 attach-agent $udp0

$ns rtmodel-at 0.4 down $n(2) $n(3)
$ns rtmodel-at 1.0 up $n(2) $n(3)

$ns at 0.01 "$cbr0 start"
$ns at 1.5 "$cbr0 stop"

$ns at 2.0 "finish"
$ns run
```

Output for ring:



Trace file:

ring.tcl hybrid.tcl

```
2610 + 1.333625 2 3 cbr 24 ------- 0 0.0 3.0 237 341
2611 r 1.334719 4 3 cbr 1000 ------- 0 0.0 3.0 186 270
2612 r 1.335094 4 3 cbr 24 ------- 0 0.0 3.0 187 271
2613 - 1.33625 6 5 cbr 24 ------- 0 0.0 3.0 193 277
2614 r 1.336625 0 6 cbr 1000 ------- 0 0.0 3.0 194 278
2615 + 1.336625 6 5 cbr 1000 ------- 0 0.0 3.0 194 278
2616 - 1.336625 6 5 cbr 1000 ------- 0 0.0 3.0 194 278
2617 r 1.337 0 6 cbr 24 ------- 0 0.0 3.0 195 279
2618 + 1.337 6 5 cbr 24 ------- 0 0.0 3.0 195 279
2619 r 1.337875 2 3 cbr 1000 ------- 0 0.0 3.0 234 338
2620 r 1.33825 2 3 cbr 24 ------- 0 0.0 3.0 235 339
2621 - 1.339625 0 1 cbr 24 ------- 0 0.0 3.0 241 345
2622 + 1.34 0 1 cbr 1000 ------- 0 0.0 3.0 266 370
2623 + 1.34 0 1 cbr 24 ------- 0 0.0 3.0 267 371
2624 - 1.34 0 1 cbr 1000 ------- 0 0.0 3.0 242 346
2625 - 1.340984 5 4 cbr 24 ------- 0 0.0 3.0 191 275
2626 r 1.34125 6 5 cbr 1000 ------- 0 0.0 3.0 192 276
2627 + 1.34125 5 4 cbr 1000 ------- 0 0.0 3.0 192 276
2628 - 1.341359 5 4 cbr 1000 ------- 0 0.0 3.0 192 276
2629 r 1.341625 6 5 cbr 24 ------- 0 0.0 3.0 193 277
2630 + 1.341625 5 4 cbr 24 ------- 0 0.0 3.0 193 277
2631 - 1.34425 1 2 cbr 24 ------- 0 0.0 3.0 239 343
2632 r 1.344625 0 1 cbr 1000 ------- 0 0.0 3.0 240 344
2633 + 1.344625 1 2 cbr 1000 ------- 0 0.0 3.0 240 344
2634 - 1.344625 1 2 cbr 1000 ------- 0 0.0 3.0 240 344
2635 r 1.345 0 1 cbr 24 ------- 0 0.0 3.0 241 345
2636 + 1.345 1 2 cbr 24 ------- 0 0.0 3.0 241 345
2637 - 1.345719 4 3 cbr 24 ------- 0 0.0 3.0 189 273
2638 r 1.345984 5 4 cbr 1000 ------- 0 0.0 3.0 190 274
2639 + 1.345984 4 3 cbr 1000 ------- 0 0.0 3.0 190 274
2640 - 1.346094 4 3 cbr 1000 ------- 0 0.0 3.0 190 274
2641 r 1.346359 5 4 cbr 24 ------- 0 0.0 3.0 191 275
2642 + 1.346359 4 3 cbr 24 ------- 0 0.0 3.0 191 275
2643 - 1.347625 0 6 cbr 24 ------- 0 0.0 3.0 197 281
2644 - 1.348 0 6 cbr 1000 ------- 0 0.0 3.0 198 286
2645 - 1.348875 2 3 cbr 24 ------- 0 0.0 3.0 237 341
2646 r 1.34925 1 2 cbr 1000 ------- 0 0.0 3.0 238 342
2647 + 1.34925 2 3 cbr 1000 ------- 0 0.0 3.0 238 342
2648 - 1.34925 2 3 cbr 1000 ------- 0 0.0 3.0 238 342
2649 r 1.349625 1 2 cbr 24 ------- 0 0.0 3.0 239 343
2650 + 1.349625 2 3 cbr 24 ------- 0 0.0 3.0 239 343
2651 + 1.35 0 1 cbr 1000 ------- 0 0.0 3.0 268 372
```

9. Program in NS3 for connecting multiple routers and nodes and building a hybrid topology.

Hybrid:

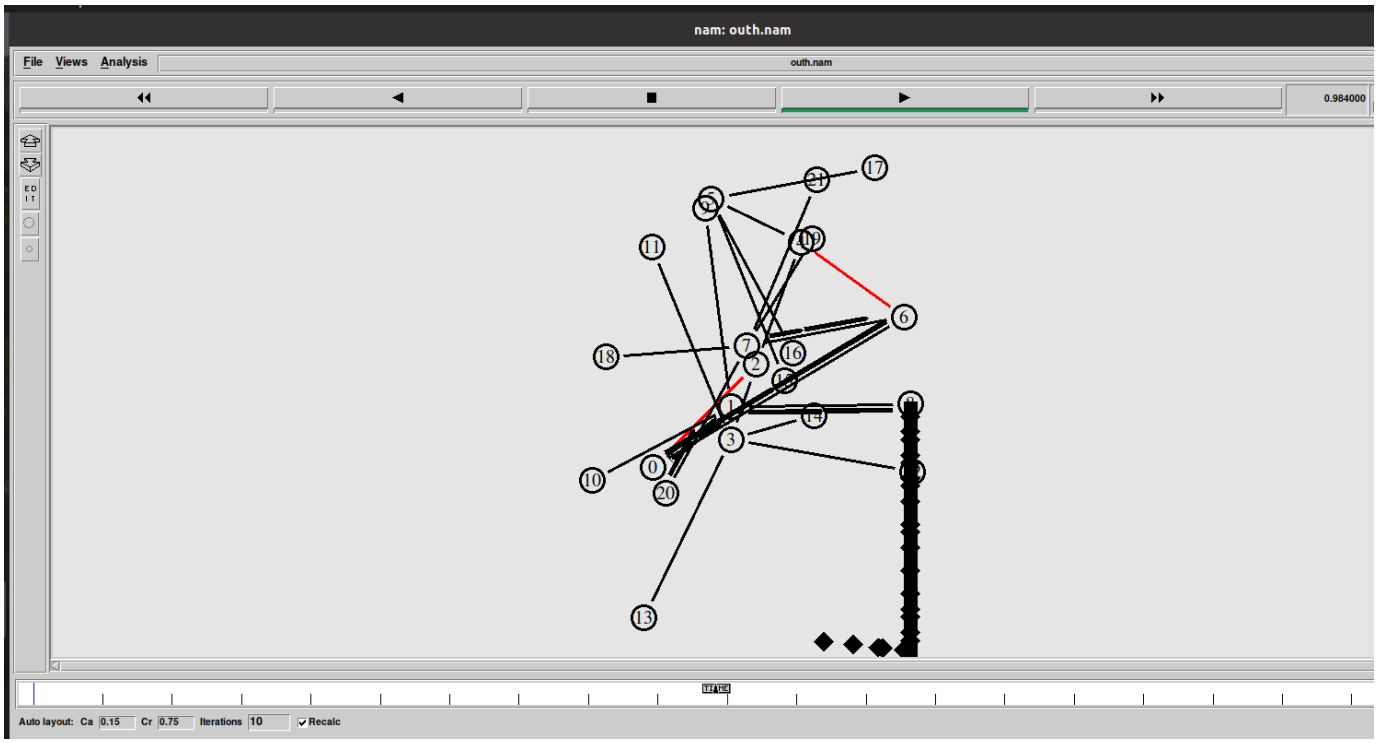
```
set ns [new Simulator]
set nf [open outh.nam w]
$ns namtrace-all $nf

set nf [open outh.tr w]
$ns trace-all $nf

proc finish {} {
        global ns nf
        $ns flush-trace
        close $nf
        exec nam outh.nam
```

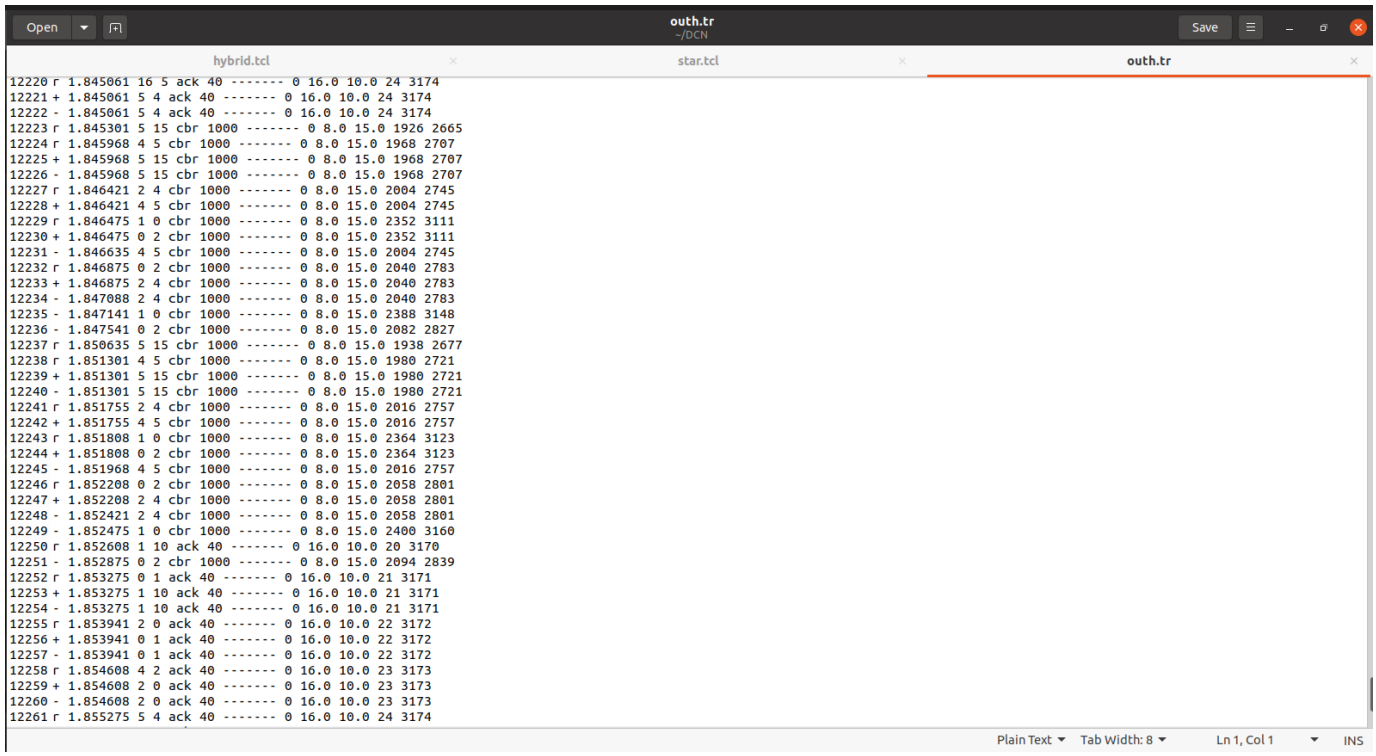
```
        exec gedit outh.tr
        exit 0
}

$ns rtproto DV

#-----------------------------------------------------------------
# create nodes and establish links
for {set i 1} {$i<5} {incr i} {
set r($i) [$ns node]
set h($i) [$ns node]
}

for {set i 1} {$i<15} {incr i} {
set p($i) [$ns node]
}

#Creating Links between h and p
for {set i 1} {$i<4} {incr i} {
$ns duplex-link $h(1) $p($i) 1.5Mb 10ms SFQ
$ns duplex-link $h(3) $p([expr ($i+7)]) 1.5Mb 10ms SFQ
}

for {set i 4} {$i<8} {incr i} {
$ns duplex-link $h(2) $p($i) 1.5Mb 10ms SFQ
$ns duplex-link $h(4) $p([expr ($i+7)]) 1.5Mb 10ms SFQ
}


#Creating Links between r nodes and connecting r to h
for {set i 1} {$i<4} {incr i} {
$ns duplex-link $r($i) $r([expr ($i+1)]) 1.5Mb 10ms SFQ
$ns duplex-link $r($i) $h($i) 1.5Mb 10ms SFQ
}

$ns duplex-link $r(4) $r(1) 1.5Mb 10ms SFQ
$ns duplex-link $r(4) $h(4) 1.5Mb 10ms SFQ



# orienting the nodes
$ns duplex-link-op $r(1) $h(1) orient up
$ns duplex-link-op $r(2) $h(2) orient right
$ns duplex-link-op $r(3) $h(3) orient down
$ns duplex-link-op $r(4) $h(4) orient left
```

```
#
#creating tcp agents and attach
set tcp3 [new Agent/TCP]
set tcp9 [new Agent/TCPSink]

$ns attach-agent $p(3) $tcp3
$ns attach-agent $p(9) $tcp9

$ns connect $tcp3 $tcp9

set tcp5 [new Agent/TCP]
set tcp12 [new Agent/TCPSink]

$ns attach-agent $p(5) $tcp5
$ns attach-agent $p(12) $tcp12

$ns connect $tcp5 $tcp12

#creating FTP application for tcp agents
set ftp3 [new Application/FTP]
set ftp5 [new Application/FTP]

$ftp3 attach-agent $tcp3
$ftp5 attach-agent $tcp5
# ----------

#creating udp agents and attach
set udp13 [new Agent/UDP]
set udp6 [new Agent/Null]

$ns attach-agent $p(13) $udp13
$ns attach-agent $p(6) $udp6

$ns connect $udp13 $udp6

set udp1 [new Agent/UDP]
set udp8 [new Agent/Null]

$ns attach-agent $p(1) $udp1
$ns attach-agent $p(8) $udp8

$ns connect $udp1 $udp8
```

```
#creating CBR applications for udp
set cbr13 [new Application/Traffic/CBR]

# send 50 packets in 1 second i.e 1 packet every 1/50 second
$cbr13 set packetSize_ 1536
$cbr13 set interval_ 0.02
$cbr13 attach-agent $udp13

set cbr1 [new Application/Traffic/CBR]

# send 400 packets in 1 second i.e 1 packet every 1/400 second
$cbr1 set packetSize_ 5632
$cbr1 set interval_ 0.0025
$cbr1 attach-agent $udp1

# ------------------------------------------------------------

# setting events

$ns rtmodel-at 0.7 down $r(1) $r(2)
$ns rtmodel-at 1.0 up $r(1) $r(2)

$ns rtmodel-at 0.9 down $r(4) $r(3)
$ns rtmodel-at 1.3 up $r(4) $r(3)


$ns at 0.2 "$ftp3 start"
$ns at 1.8 "$ftp3 stop"

$ns at 0.3 "$ftp5 start"
$ns at 1.4 "$ftp5 stop"


$ns at 0.4 "$cbr13 start"
$ns at 1.6 "$cbr13 stop"

$ns at 0.7 "$cbr1 start"
$ns at 1.7 "$cbr1 stop"

$ns at 2.0 "finish"
$ns run
```

Output:






```
12220 r 1.845061 16 5 ack 40 ------- 0 16.0 10.0 24 3174
12221 + 1.845061 5 4 ack 40 ------- 0 16.0 10.0 24 3174
12222 - 1.845061 5 4 ack 40 ------- 0 16.0 10.0 24 3174
12223 r 1.845301 5 15 cbr 1000 ------- 0 8.0 15.0 1926 2665
12224 r 1.845968 4 5 cbr 1000 ------- 0 8.0 15.0 1968 2707
12225 + 1.845968 5 15 cbr 1000 ------- 0 8.0 15.0 1968 2707
12226 - 1.845968 5 15 cbr 1000 ------- 0 8.0 15.0 1968 2707
12227 r 1.846421 2 4 cbr 1000 ------- 0 8.0 15.0 2004 2745
12228 + 1.846421 4 5 cbr 1000 ------- 0 8.0 15.0 2004 2745
12229 r 1.846475 1 0 cbr 1000 ------- 0 8.0 15.0 2352 3111
12230 + 1.846475 0 2 cbr 1000 ------- 0 8.0 15.0 2352 3111
12231 - 1.846635 4 5 cbr 1000 ------- 0 8.0 15.0 2004 2745
12232 r 1.846875 0 2 cbr 1000 ------- 0 8.0 15.0 2040 2783
12233 + 1.846875 2 4 cbr 1000 ------- 0 8.0 15.0 2040 2783
12234 - 1.847088 2 4 cbr 1000 ------- 0 8.0 15.0 2040 2783
12235 - 1.847141 1 0 cbr 1000 ------- 0 8.0 15.0 2388 3148
12236 - 1.847541 0 2 cbr 1000 ------- 0 8.0 15.0 2082 2827
12237 r 1.850635 5 15 cbr 1000 ------- 0 8.0 15.0 1938 2677
12238 r 1.851301 4 5 cbr 1000 ------- 0 8.0 15.0 1980 2721
12239 + 1.851301 5 15 cbr 1000 ------- 0 8.0 15.0 1980 2721
12240 - 1.851301 5 15 cbr 1000 ------- 0 8.0 15.0 1980 2721
12241 r 1.851755 2 4 cbr 1000 ------- 0 8.0 15.0 2016 2757
12242 + 1.851755 4 5 cbr 1000 ------- 0 8.0 15.0 2016 2757
12243 r 1.851808 1 0 cbr 1000 ------- 0 8.0 15.0 2364 3123
12244 + 1.851808 0 2 cbr 1000 ------- 0 8.0 15.0 2364 3123
12245 - 1.851968 4 5 cbr 1000 ------- 0 8.0 15.0 2016 2757
12246 r 1.852208 0 2 cbr 1000 ------- 0 8.0 15.0 2058 2801
12247 + 1.852208 2 4 cbr 1000 ------- 0 8.0 15.0 2058 2801
12248 - 1.852421 2 4 cbr 1000 ------- 0 8.0 15.0 2058 2801
12249 - 1.852475 1 0 cbr 1000 ------- 0 8.0 15.0 2400 3160
12250 r 1.852608 1 10 ack 40 ------- 0 16.0 10.0 20 3170
12251 - 1.852875 0 2 cbr 1000 ------- 0 8.0 15.0 2094 2839
12252 r 1.853275 0 1 ack 40 ------- 0 16.0 10.0 21 3171
12253 + 1.853275 1 10 ack 40 ------- 0 16.0 10.0 21 3171
12254 - 1.853275 1 10 ack 40 ------- 0 16.0 10.0 21 3171
12255 r 1.853941 2 0 ack 40 ------- 0 16.0 10.0 22 3172
12256 + 1.853941 0 1 ack 40 ------- 0 16.0 10.0 22 3172
12257 - 1.853941 0 1 ack 40 ------- 0 16.0 10.0 22 3172
12258 r 1.854608 4 2 ack 40 ------- 0 16.0 10.0 23 3173
12259 + 1.854608 2 0 ack 40 ------- 0 16.0 10.0 23 3173
12260 - 1.854608 2 0 ack 40 ------- 0 16.0 10.0 23 3173
12261 r 1.855275 5 4 ack 40 ------- 0 16.0 10.0 24 3174
```

10. Installation and configuration of NetAnim.

Prerequisites

1. mercurial
2. QT4 development packages (recommended version 4.7)

Debian/Ubuntu Linux distribution:

1. apt-get install mercurial
2. apt-get install qt4-dev-tools

Red Hat/Fedora based distribution:

1. yum install mercurial
2. yum install qt4
3. yum install qt4-devel

Mac/OSX

1. mercurial
2. Qt4 : Install Qt4 (including Qt Creator if possible) from http://qt.nokia.com/downloads/

Windows

1. mercurial
2. Qt Creator
3. Microsoft Visual C++ (Visual Studio 2010 and over http://www.microsoft.com/visualstudio/eng/products/visual-studio-express-products) or MinGw compiler

Downloading NetAnim

- NetAnim 3.105:

```
hg clone http://code.nsnam.org/netanim
```

Building and Starting NetAnim

NetAnim uses a QT4 build tool called qmake. Only qmake version 4.7 is supported Please read the #Prerequisites before proceeding

```
cd netanim
make clean
qmake NetAnim.pro (For MAC Users: qmake -spec macx-g++ NetAnim.pro)
make
```

Note: qmake could be "qmake-qt4" in some systems

This should create an executable named "NetAnim" in the same directory. To start the application just type "./NetAnim"

For windows users:

- In the netanim repository you cloned in step: #Downloading NetAnim , there will be a file named "NetAnim.pro". Double-click it. This should open QtCreator.
- In QtCreator to go "Projects" -->"Build settings"-->Tool chain. Here a tool chain should be displayed such as MSVisual C++ compiler or MinGw etc. If not click on "Manage" and point to the local of the tool chain. See http://doc.qt.digia.com/qtcreator-2.4/creator-tool-chains.html
- In QtCreator -->Click "Build All"-->Run, it should open the NetAnim executable.

Summary of features

- Animate packets over wired-links and wireless-links (Limited support for LTE traces. No support for Ipv6)
- Packet timeline with regex filter on packet meta-data.
- Node position statistics with node trajectory plotting(path of a mobile node).
- Print brief packet-meta data on packets

- Use custom icons for nodes
- Parse flow-monitor XML files and display statistics for each flow.
- Show IP and MAC information, including peer IP and MAC for point-to-point links.
- Display double or uint32 valued counters vs time for multiple nodes in a chart or a table.
- Step through a simulation one event at a time and pause the simulation at a given time
- Print the routing table at nodes at various points in time

11. Program in NS3 to implement FTP using TCP bulk transfer.

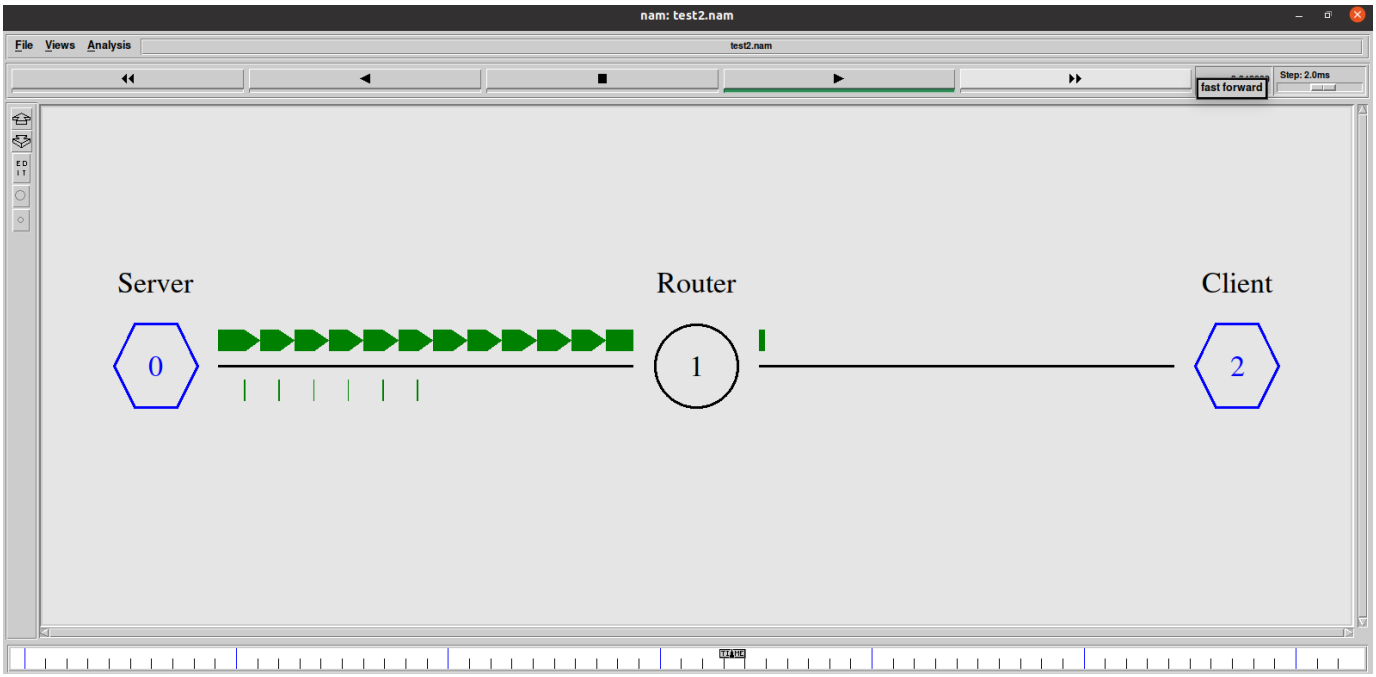
FTP

```
set ns [new Simulator]

#———-creating trace objects————————-#

set nt [open test2.tr w]
$ns trace-all $nt
#———-creating nam objects————————-#

set nf [open test2.nam w]
$ns namtrace-all $nf
#———-Setting color ID————————-#
$ns color 1 darkmagenta
$ns color 2 yellow
$ns color 3 blue
$ns color 4 green
$ns color 5 black

#———- Creating Network————————-#

set totalNodes 3

for {set i 0} {$i < $totalNodes} {incr i} {
set node_($i) [$ns node]
}

set server 0
set router 1
set client 2

#———- Creating Duplex Link————————-#
$ns duplex-link $node_($server) $node_($router) 2Mb 50ms DropTail
$ns duplex-link $node_($router) $node_($client) 2Mb 50ms DropTail
```

```
$ns duplex-link-op $node_($server) $node_($router) orient right
$ns duplex-link-op $node_($router) $node_($client) orient right

#————Labelling————-#

$ns at 0.0 "$node_($server) label Server"
$ns at 0.0 "$node_($router) label Router"
$ns at 0.0 "$node_($client) label Client"

$ns at 0.0 "$node_($server) color blue"
$ns at 0.0 "$node_($client) color blue"

$node_($server) shape hexagon
$node_($client) shape hexagon
#————Data Transfer between Nodes————-#

# Defining a transport agent for sending
set tcp [new Agent/TCP]

# Attaching transport agent to sender node
$ns attach-agent $node_($server) $tcp

# Defining a transport agent for receiving
set sink [new Agent/TCPSink]

# Attaching transport agent to receiver node
$ns attach-agent $node_($client) $sink

#Connecting sending and receiving transport agents
$ns connect $tcp $sink

#Defining Application instance
set ftp [new Application/FTP]

# Attaching transport agent to application agent
$ftp attach-agent $tcp

# Setting flow color
$tcp set fid_ 4

# data packet generation starting time
$ns at 1.0 "$ftp start"

# data packet generation ending time
$ns at 6.0 "$ftp stop"
```

```
#————finish procedure————#

proc finish {} {
global ns nf nt
$ns flush-trace
close  $nf
close $nt
puts "running nam…"
exec nam test2.nam &
exit 0
}

#Calling finish procedure
$ns at 10.0 "finish"
$ns run
```

Output:



Trace file:

Open

```
 1 + 1 0 1 tcp 40 ------- 4 0.0 2.0 0 0
 2 - 1 0 1 tcp 40 ------- 4 0.0 2.0 0 0
 3 r 1.05016 0 1 tcp 40 ------- 4 0.0 2.0 0 0
 4 + 1.05016 1 2 tcp 40 ------- 4 0.0 2.0 0 0
 5 - 1.05016 1 2 tcp 40 ------- 4 0.0 2.0 0 0
 6 r 1.10032 1 2 tcp 40 ------- 4 0.0 2.0 0 0
 7 + 1.10032 2 1 ack 40 ------- 4 2.0 0.0 0 1
 8 - 1.10032 2 1 ack 40 ------- 4 2.0 0.0 0 1
 9 r 1.15048 2 1 ack 40 ------- 4 2.0 0.0 0 1
10 + 1.15048 1 0 ack 40 ------- 4 2.0 0.0 0 1
11 - 1.15048 1 0 ack 40 ------- 4 2.0 0.0 0 1
12 r 1.20064 1 0 ack 40 ------- 4 2.0 0.0 0 1
13 + 1.20064 0 1 tcp 1040 ------- 4 0.0 2.0 1 2
14 - 1.20064 0 1 tcp 1040 ------- 4 0.0 2.0 1 2
15 + 1.20064 0 1 tcp 1040 ------- 4 0.0 2.0 2 3
16 - 1.2048 0 1 tcp 1040 ------- 4 0.0 2.0 2 3
17 r 1.2548 0 1 tcp 1040 ------- 4 0.0 2.0 1 2
18 + 1.2548 1 2 tcp 1040 ------- 4 0.0 2.0 1 2
19 - 1.2548 1 2 tcp 1040 ------- 4 0.0 2.0 1 2
20 r 1.25896 0 1 tcp 1040 ------- 4 0.0 2.0 2 3
21 + 1.25896 1 2 tcp 1040 ------- 4 0.0 2.0 2 3
22 - 1.25896 1 2 tcp 1040 ------- 4 0.0 2.0 2 3
23 r 1.30896 1 2 tcp 1040 ------- 4 0.0 2.0 1 2
24 + 1.30896 2 1 ack 40 ------- 4 2.0 0.0 1 4
25 - 1.30896 2 1 ack 40 ------- 4 2.0 0.0 1 4
26 r 1.31312 1 2 tcp 1040 ------- 4 0.0 2.0 2 3
27 + 1.31312 2 1 ack 40 ------- 4 2.0 0.0 2 5
28 - 1.31312 2 1 ack 40 ------- 4 2.0 0.0 2 5
29 r 1.35912 2 1 ack 40 ------- 4 2.0 0.0 1 4
30 + 1.35912 1 0 ack 40 ------- 4 2.0 0.0 1 4
31 - 1.35912 1 0 ack 40 ------- 4 2.0 0.0 1 4
32 r 1.36328 2 1 ack 40 ------- 4 2.0 0.0 2 5
33 + 1.36328 1 0 ack 40 ------- 4 2.0 0.0 2 5
34 - 1.36328 1 0 ack 40 ------- 4 2.0 0.0 2 5
35 r 1.40928 1 0 ack 40 ------- 4 2.0 0.0 1 4
36 + 1.40928 0 1 tcp 1040 ------- 4 0.0 2.0 3 6
37 - 1.40928 0 1 tcp 1040 ------- 4 0.0 2.0 3 6
38 + 1.40928 0 1 tcp 1040 ------- 4 0.0 2.0 4 7
39 r 1.41344 1 0 ack 40 ------- 4 2.0 0.0 2 5
40 + 1.41344 0 1 tcp 1040 ------- 4 0.0 2.0 5 8
41 + 1.41344 0 1 tcp 1040 ------- 4 0.0 2.0 6 9
42 - 1.41344 0 1 tcp 1040 ------- 4 0.0 2.0 4 7
43 - 1.4176 0 1 tcp 1040 ------- 4 0.0 2.0 5 8
44 - 1.42176 0 1 tcp 1040 ------- 4 0.0 2.0 6 9
```

12. Program in NS3 for connecting multiple routers and nodes and building a hybrid topology and calculating network performance.

HYBRID

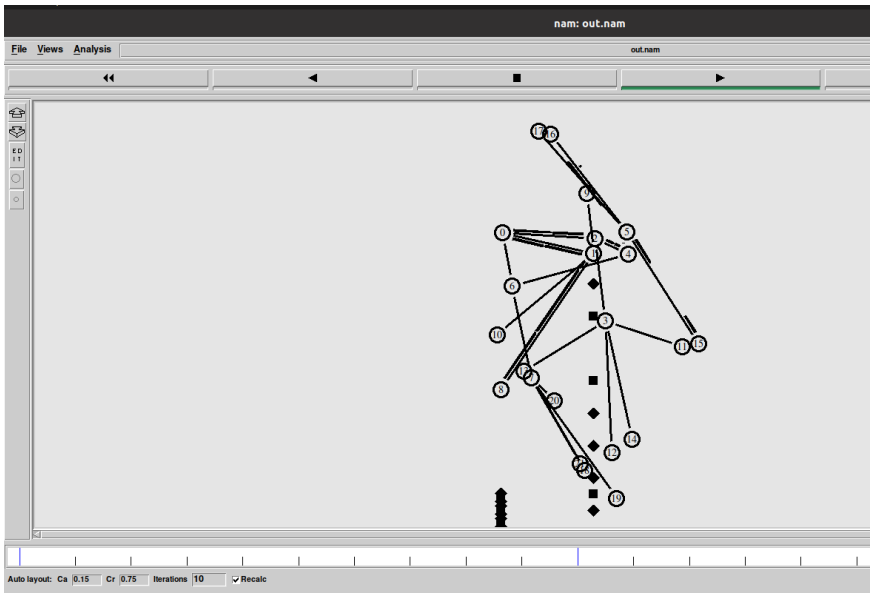
```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf

set tf [open out.tr w]
$ns trace-all $tf


proc finish {} {
        global ns nf
        $ns flush-trace
        close $nf
        exec nam out.nam
        exec gedit out.tr
        exec awk -f hybrid.awk out.tr &
        exit 0
}

$ns rtproto DV

#_____________________________________
# create nodes and establish links
for {set i 1} {$i<5} {incr i} {
set r($i) [$ns node]
set h($i) [$ns node]
}

for {set i 1} {$i<15} {incr i} {
set p($i) [$ns node]
}

#Creating Links between h and p
for {set i 1} {$i<4} {incr i} {
$ns duplex-link $h(1) $p($i) 1.5Mb 10ms SFQ
$ns duplex-link $h(3) $p([expr ($i+7)]) 1.5Mb 10ms SFQ
}

for {set i 4} {$i<8} {incr i} {
$ns duplex-link $h(2) $p($i) 1.5Mb 10ms SFQ
$ns duplex-link $h(4) $p([expr ($i+7)]) 1.5Mb 10ms SFQ
}
```

```
#Creating Links between r nodes and connecting r to h
for {set i 1} {$i<4} {incr i} {
$ns duplex-link $r($i) $r([expr ($i+1)]) 1.5Mb 10ms SFQ
$ns duplex-link $r($i) $h($i) 1.5Mb 10ms SFQ
}

$ns duplex-link $r(4) $r(1) 1.5Mb 10ms SFQ
$ns duplex-link $r(4) $h(4) 1.5Mb 10ms SFQ



# orienting the nodes
$ns duplex-link-op $r(1) $h(1) orient up
$ns duplex-link-op $r(2) $h(2) orient right
$ns duplex-link-op $r(3) $h(3) orient down
$ns duplex-link-op $r(4) $h(4) orient left


#_____________________________________
#creating tcp agents and attach
set tcp3 [new Agent/TCP]
set tcp9 [new Agent/TCPSink]

$ns attach-agent $p(3) $tcp3
$ns attach-agent $p(9) $tcp9

$ns connect $tcp3 $tcp9

set tcp5 [new Agent/TCP]
set tcp12 [new Agent/TCPSink]

$ns attach-agent $p(5) $tcp5
$ns attach-agent $p(12) $tcp12

$ns connect $tcp5 $tcp12

#creating FTP application for tcp agents
set ftp3 [new Application/FTP]
set ftp5 [new Application/FTP]

$ftp3 attach-agent $tcp3
$ftp5 attach-agent $tcp5
# _____________________________________

#creating udp agents and attach
set udp13 [new Agent/UDP]
```

```
set udp6 [new Agent/Null]

$ns attach-agent $p(13) $udp13
$ns attach-agent $p(6) $udp6

$ns connect $udp13 $udp6

set udp1 [new Agent/UDP]
set udp8 [new Agent/Null]

$ns attach-agent $p(1) $udp1
$ns attach-agent $p(8) $udp8

$ns connect $udp1 $udp8


#creating CBR applications for udp
set cbr13 [new Application/Traffic/CBR]

# send 50 packets in 1 second i.e 1 packet every 1/50 second
$cbr13 set packetSize_ 1536
$cbr13 set interval_ 0.02
$cbr13 attach-agent $udp13

set cbr1 [new Application/Traffic/CBR]

# send 400 packets in 1 second i.e 1 packet every 1/400 second
$cbr1 set packetSize_ 5632
$cbr1 set interval_ 0.0025
$cbr1 attach-agent $udp1

# ____________________________________

# setting events

$ns rtmodel-at 0.7 down $r(1) $r(2)
$ns rtmodel-at 1.0 up $r(1) $r(2)

$ns rtmodel-at 0.9 down $r(4) $r(3)
$ns rtmodel-at 1.3 up $r(4) $r(3)


$ns at 0.2 "$ftp3 start"
$ns at 1.8 "$ftp3 stop"

$ns at 0.3 "$ftp5 start"
```

```
$ns at 1.4 "$ftp5 stop"


$ns at 0.4 "$cbr13 start"
$ns at 1.6 "$cbr13 stop"

$ns at 0.7 "$cbr1 start"
$ns at 1.7 "$cbr1 stop"

$ns at 2.0 "finish"
$ns run
```

Output:



Terminal:

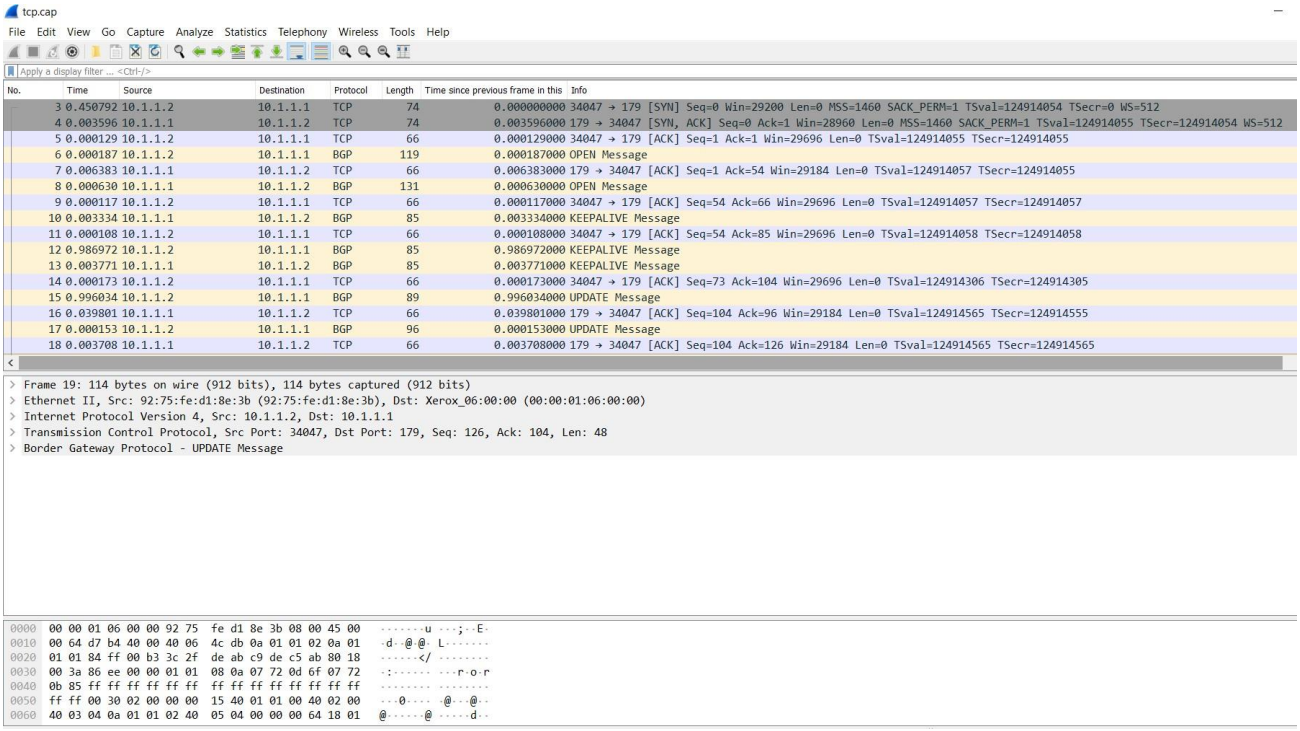
Trace file:

Open

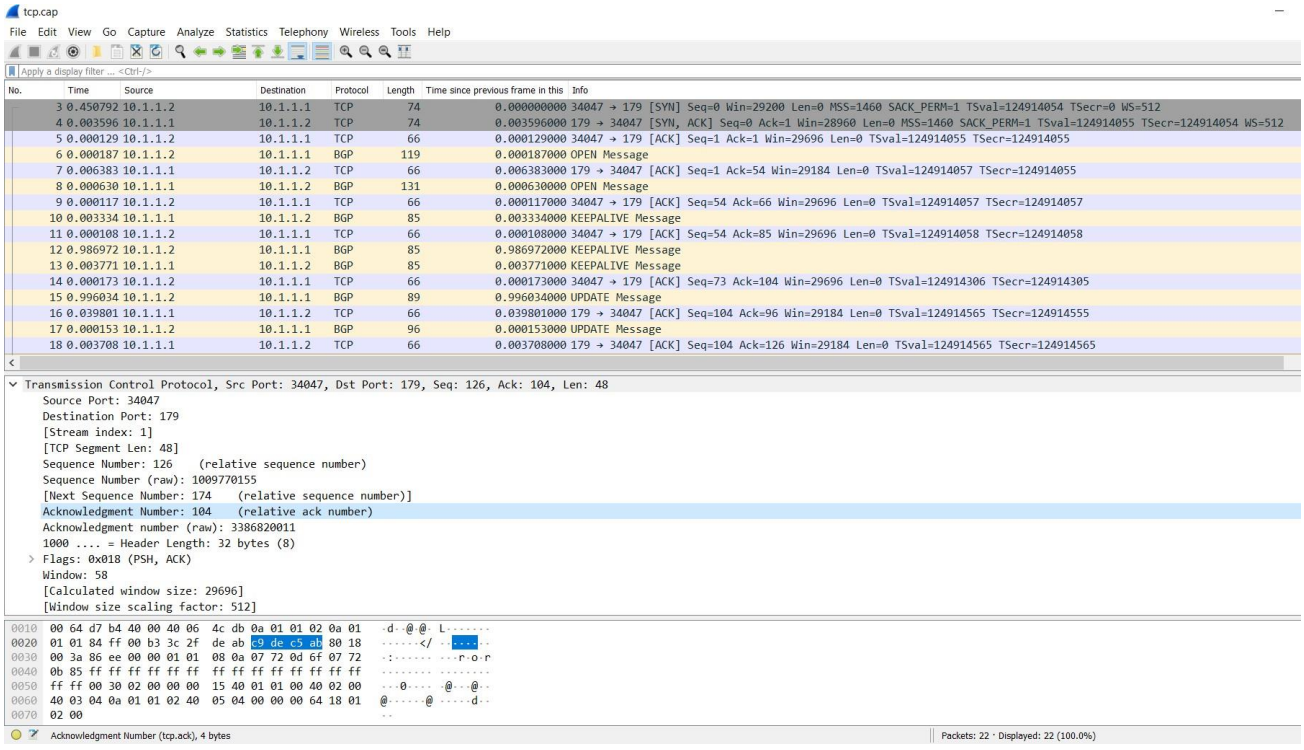
```
 1 + 0.00017 0 1 rtProtoDV 22 ------- 0 0.1 1.1 -1 0
 2 - 0.00017 0 1 rtProtoDV 22 ------- 0 0.1 1.1 -1 0
 3 + 0.00017 0 2 rtProtoDV 22 ------- 0 0.1 2.1 -1 1
 4 - 0.00017 0 2 rtProtoDV 22 ------- 0 0.1 2.1 -1 1
 5 + 0.00017 0 6 rtProtoDV 22 ------- 0 0.1 6.1 -1 2
 6 - 0.00017 0 6 rtProtoDV 22 ------- 0 0.1 6.1 -1 2
 7 + 0.007102 2 0 rtProtoDV 22 ------- 0 2.1 0.1 -1 3
 8 - 0.007102 2 0 rtProtoDV 22 ------- 0 2.1 0.1 -1 3
 9 + 0.007102 2 3 rtProtoDV 22 ------- 0 2.1 3.1 -1 4
10 - 0.007102 2 3 rtProtoDV 22 ------- 0 2.1 3.1 -1 4
11 + 0.007102 2 4 rtProtoDV 22 ------- 0 2.1 4.1 -1 5
12 - 0.007102 2 4 rtProtoDV 22 ------- 0 2.1 4.1 -1 5
13 r 0.010287 0 1 rtProtoDV 22 ------- 0 0.1 1.1 -1 0
14 + 0.010287 1 0 rtProtoDV 22 ------- 0 1.1 0.1 -1 6
15 - 0.010287 1 0 rtProtoDV 22 ------- 0 1.1 0.1 -1 6
16 + 0.010287 1 8 rtProtoDV 22 ------- 0 1.1 8.2 -1 7
17 - 0.010287 1 8 rtProtoDV 22 ------- 0 1.1 8.2 -1 7
18 + 0.010287 1 9 rtProtoDV 22 ------- 0 1.1 9.1 -1 8
19 - 0.010287 1 9 rtProtoDV 22 ------- 0 1.1 9.1 -1 8
20 + 0.010287 1 10 rtProtoDV 22 ------- 0 1.1 10.2 -1 9
21 - 0.010287 1 10 rtProtoDV 22 ------- 0 1.1 10.2 -1 9
22 r 0.010287 0 2 rtProtoDV 22 ------- 0 0.1 2.1 -1 1
23 + 0.010287 2 0 rtProtoDV 22 ------- 0 2.1 0.1 -1 10
24 - 0.010287 2 0 rtProtoDV 22 ------- 0 2.1 0.1 -1 10
25 + 0.010287 2 3 rtProtoDV 22 ------- 0 2.1 3.1 -1 11
26 - 0.010287 2 3 rtProtoDV 22 ------- 0 2.1 3.1 -1 11
27 + 0.010287 2 4 rtProtoDV 22 ------- 0 2.1 4.1 -1 12
28 - 0.010287 2 4 rtProtoDV 22 ------- 0 2.1 4.1 -1 12
29 r 0.010287 0 6 rtProtoDV 22 ------- 0 0.1 6.1 -1 2
30 + 0.010287 6 0 rtProtoDV 22 ------- 0 6.1 0.1 -1 13
31 - 0.010287 6 0 rtProtoDV 22 ------- 0 6.1 0.1 -1 13
32 + 0.010287 6 4 rtProtoDV 22 ------- 0 6.1 4.1 -1 14
33 - 0.010287 6 4 rtProtoDV 22 ------- 0 6.1 4.1 -1 14
34 + 0.010287 6 7 rtProtoDV 22 ------- 0 6.1 7.1 -1 15
35 - 0.010287 6 7 rtProtoDV 22 ------- 0 6.1 7.1 -1 15
36 r 0.01722 2 0 rtProtoDV 22 ------- 0 2.1 0.1 -1 3
37 + 0.01722 0 1 rtProtoDV 22 ------- 0 0.1 1.1 -1 16
38 - 0.01722 0 1 rtProtoDV 22 ------- 0 0.1 1.1 -1 16
39 + 0.01722 0 2 rtProtoDV 22 ------- 0 0.1 2.1 -1 17
40 - 0.01722 0 2 rtProtoDV 22 ------- 0 0.1 2.1 -1 17
41 + 0.01722 0 6 rtProtoDV 22 ------- 0 0.1 6.1 -1 18
42 - 0.01722 0 6 rtProtoDV 22 ------- 0 0.1 6.1 -1 18
43 r 0.01722 2 3 rtProtoDV 22 ------- 0 2.1 3.1 -1 4
44 + 0.01722 3 2 rtProtoDV 22 ------- 0 3.1 2.1 -1 19
```

13. To analyze network traces using wireshark software.

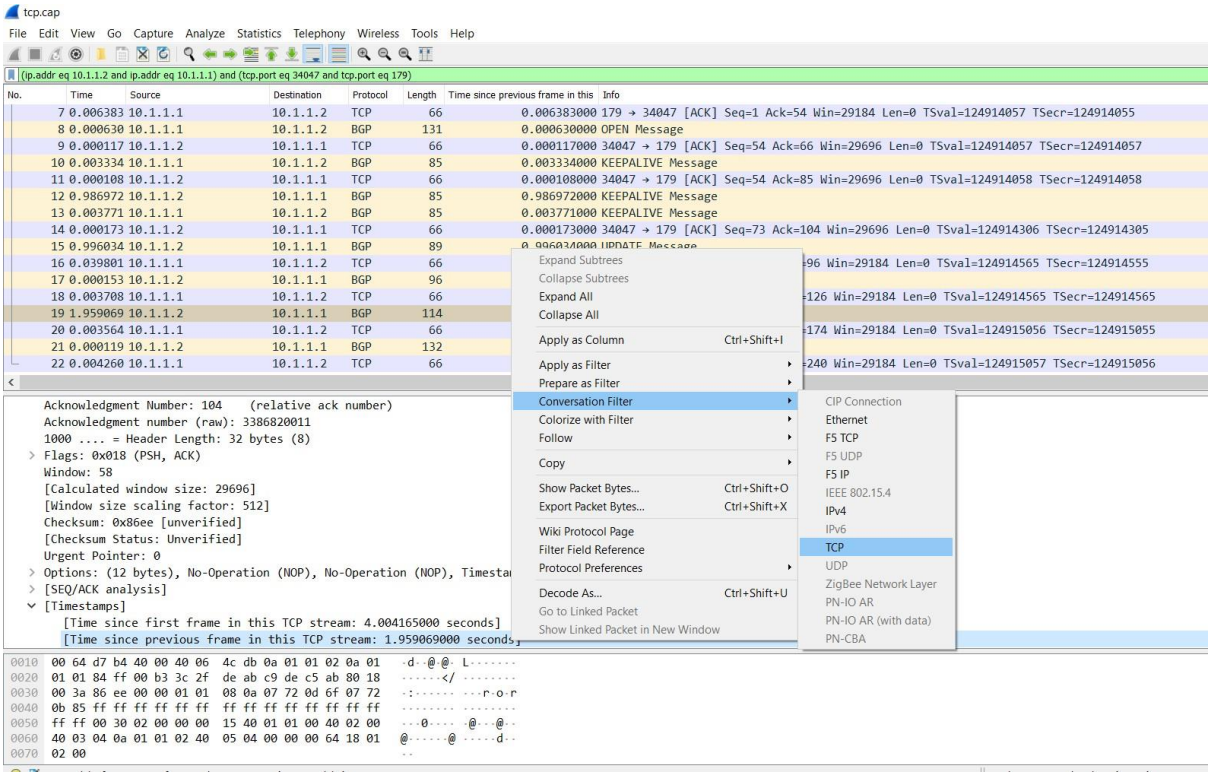
The file given:



TCP-ACK Message:



Using conversation filters:



Time since previous frame in this file: