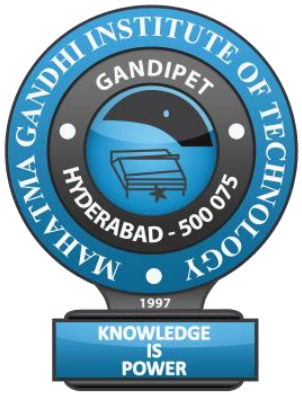
LABORATARY MANUAL

Of

Data Communications and Networks Lab

For 3rd Year , 1st Semester B.Tech (R18) ECE



Department of Electronics and Communication

Engineering

Mahatma Gandhi Institute of Technology

Chaitanya Bharathi (P.O), GandiPet, Hyderabad -75

EC506PC: DATA COMMUNICATIONS AND NETWORKS LAB

B.Tech. III Year I Semester

L	T	P	C
0	0	3	1.5

Note:

A. Minimum of 12 Experiments have to be conducted

B. All the Experiments may be Conducted using Network Simulation software like NS-2, NSG-2.1 and Wire SHARK/equivalent software.

Note: For Experiments 2 to 10 Performance may be evaluated through simulation by using the parameters Throughput, Packet Delivery Ratio, Delay etc.

1. Writing a TCL Script to create two nodes and links between nodes
2. Writing a TCL Script to transmit data between nodes
3. Evaluate the performance of various LAN Topologies
4. Evaluate the performance of Drop Tail and RED queue management schemes
5. Evaluate the performance of CBQ and FQ Scheduling Mechanisms
6. Evaluate the performance of TCP and UDP Protocols
7. Evaluate the performance of TCP, New Reno and Vegas
8. Evaluate the performance of AODV and DSR routing protocols
9. Evaluate the performance of AODV and DSDV routing protocols
10. Evaluate the performance of IEEE 802.11 and IEEE 802.15.4
11. Evaluate the performance of IEEE 802.11 and SMAC
12. Capturing and Analysis of TCP and IP Packets
13. Simulation and Analysis of ICMP and IGMP Packets
14. Analyze the Protocols SCTP, ARP, NetBIOS, IPX VINES
15. Analysis of HTTP, DNS and DHCP Protocols

Major Equipment Required:

Required software (Open Source) like NS-2, NSG-2.1 and Wire SHARK

1-CREATING TWO NODES

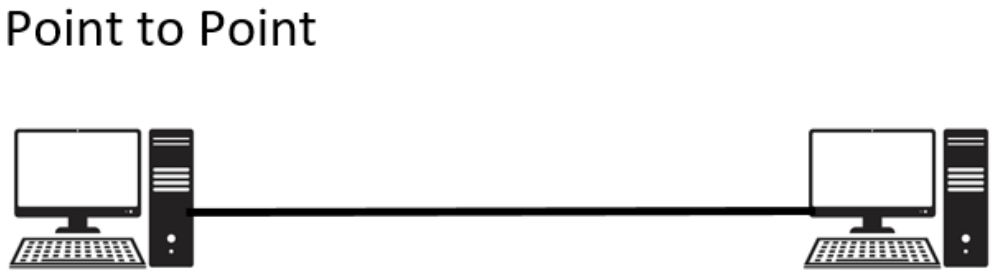
Aim: To write a TCL Script to create two nodes and links between nodes

Software Required : NS2 / NSG2.1

Operating System : Ubuntu

Procedure:

1. Go to the directory where the NSG2.jar file is present. (~/Desktop/NS2)
2. After going to the directory to open the NSG2.jar file. type the command in terminal
   java –jar NSG2.jar
3. Click on scenario and select new wired scenario.
4. Place 2 nodes by clicking on the work area.
5. Go to parameters and click on done.
6. Go to tcl script and save it on desktop.
7. The TCL script has been generated.



Create Link between the Nodes

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(stop)   10.0                ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile

#===================================
#        Nodes Definition
#===================================
#Create 2 nodes
set n0 [$ns node]
set n1 [$ns node]

#===================================
#        Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50

#Give node position (for NAM)
$ns duplex-link-op $n0 $n1 orient right
```

```
#================================
#       Agents Definition
#================================

#================================
#       Applications Definition
#================================

#================================
#       Termination
#================================
#Define a 'finish' procedure
proc finish {} {
   global ns tracefile namfile
   $ns flush-trace
   close $tracefile
   close $namfile
   exec nam out.nam &
   exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

Result: The TCL script is generated by using NSG2.1 to create two nodes and link between the nodes.

Reasoning Questions

1. What is the purpose of Network Simulation?
2. What is NS2?
3. Which language we use to create Network scenario file executed by NS2?
4. Why we are using the program NSG2.jar?

2- POINT to POINT COMMUNICATION

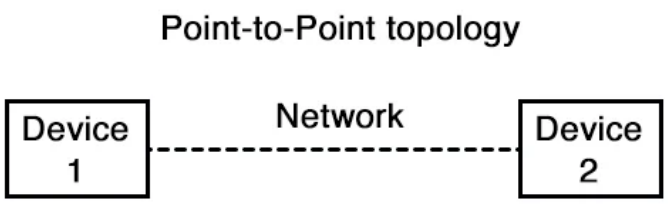
Aim: To write a TCL Script to transmit data between two nodes

Software Required : NS2 / NSG2.1

Operating System : Ubuntu

Procedure:

1. Go to the directory where the NSG2.jar file is present. (~/Desktop/NS2)
2. To open the NSG2.jar file. type the command in terminal java –jar NSG2.jar
3. Click on scenario and select new wired scenario.
4. Place 2 nodes by clicking on the work area.
5. Select the link as Drop-Tail and connect the two nodes.
6. In Agent Tab select a source node and a destination node and give their agents as TCP and TCPSink. Connect to source & destination i.e. TCP & TCPSink
7. Then give FTP as application to source node only in Application tab.
8. Go to parameters and change trace file, nam file names, click on done.
9. Click on tcl to get tcl code. Save the file as <filename>.tcl.
10. Go to the directory ~/Desktop/NS2
11. See the contents of folder by giving command: ls.
12. Run the tcl file by giving command: ns <filename>.tcl.
13. The movement of packets can be seen under NAM console.
14. Corresponding trace file and nam file are generated on the ~/Desktop/NS2.
15. The end to end delay and throughput can be calculated by the following commands:
    awk –f wired1.awk <tracefile>.tr



B. Point to Point Communication

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#=================================
#     Simulation parameters setup
#=================================
set val(stop)   10.0                ;# time of simulation end

#=================================
#       Initialization
#=================================
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
set tracefile [open exp2.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open exp2.nam w]
$ns namtrace-all $namfile

#=================================
#       Nodes Definition
#=================================
#Create 2 nodes
set n0 [$ns node]
set n1 [$ns node]

#=================================
#       Links Definition
#=================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50

#Give node position (for NAM)
$ns duplex-link-op $n0 $n1 orient right
```

```
#==================================
#        Agents Definition
#==================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#==================================
#        Applications Definition
#==================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 9.0 "$ftp0 stop"


#==================================
#        Termination
#==================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam exp2.nam &
    exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

TRACE FILE:

```
+ 1 0 1 tcp 40 ------- 0 0.0 1.0 0 0
- 1 0 1 tcp 40 ------- 0 0.0 1.0 0 0
r 1.010003 0 1 tcp 40 ------- 0 0.0 1.0 0 0
+ 1.010003 1 0 ack 40 ------- 0 1.0 0.0 0 1
- 1.010003 1 0 ack 40 ------- 0 1.0 0.0 0 1
r 1.020006 1 0 ack 40 ------- 0 1.0 0.0 0 1
+ 1.020006 0 1 tcp 1540 ------- 0 0.0 1.0 1 2
- 1.020006 0 1 tcp 1540 ------- 0 0.0 1.0 1 2
+ 1.020006 0 1 tcp 1540 ------- 0 0.0 1.0 2 3
- 1.02013 0 1 tcp 1540 ------- 0 0.0 1.0 2 3
r 1.03013 0 1 tcp 1540 ------- 0 0.0 1.0 1 2
+ 1.03013 1 0 ack 40 ------- 0 1.0 0.0 1 4
- 1.03013 1 0 ack 40 ------- 0 1.0 0.0 1 4
r 1.030253 0 1 tcp 1540 ------- 0 0.0 1.0 2 3
+ 1.030253 1 0 ack 40 ------- 0 1.0 0.0 2 5
- 1.030253 1 0 ack 40 ------- 0 1.0 0.0 2 5
r 1.040133 1 0 ack 40 ------- 0 1.0 0.0 1 4
+ 1.040133 0 1 tcp 1540 ------- 0 0.0 1.0 3 6
- 1.040133 0 1 tcp 1540 ------- 0 0.0 1.0 3 6
+ 1.040133 0 1 tcp 1540 ------- 0 0.0 1.0 4 7
r 1.040256 1 0 ack 40 ------- 0 1.0 0.0 2 5
+ 1.040256 0 1 tcp 1540 ------- 0 0.0 1.0 5 8
+ 1.040256 0 1 tcp 1540 ------- 0 0.0 1.0 6 9
- 1.040256 0 1 tcp 1540 ------- 0 0.0 1.0 4 7
- 1.040379 0 1 tcp 1540 ------- 0 0.0 1.0 5 8
- 1.040502 0 1 tcp 1540 ------- 0 0.0 1.0 6 9
r 1.050256 0 1 tcp 1540 ------- 0 0.0 1.0 3 6
+ 1.050256 1 0 ack 40 ------- 0 1.0 0.0 3 10
- 1.050256 1 0 ack 40 ------- 0 1.0 0.0 3 10
r 1.050379 0 1 tcp 1540 ------- 0 0.0 1.0 4 7
+ 1.050379 1 0 ack 40 ------- 0 1.0 0.0 4 11
- 1.050379 1 0 ack 40 ------- 0 1.0 0.0 4 11
r 1.050502 0 1 tcp 1540 ------- 0 0.0 1.0 5 8
```

Calculations:

TOPOLOGY	NUMBER OF NODES	END TO END DELAY	THROUGHPUT
Point to Point Communication			

Result The TCL script is generated by using NSG2.1 to create two nodes and to transmit data between nodes

Reasoning Questions

1. What is the input file given to NS2?
2. Which files are generated by NS2, explain the significance?
3. Draw the NS2 architecture?
4. What is the trace file format for wired networks?
5. Give the formula for the parameters calculated?

3-LAN TOPOLOGIES

Aim: To evaluate the performance of various LAN Topologies

Software Required : NS2 / NSG2.1

Operating System : Ubuntu

Procedure:

1. Double click on NSG2.1 icon.
2. Click on scenario and select new wired scenario.
3. Place desired number of nodes , select the link as Drop-Tail and connect the nodes in different LAN topologies.
4. Select a source node and a destination node and give their agents as TCP and TCP Sink. Then give FTP as application to source node only.
5. Go to parameters and click on done. Go to tcl script and save it on desktop.
6. Go to terminal and change directory to desktop as: cd Desktop.
7. See the contents of desktop by giving command: ls.
8. Run the tcl file by giving command: ns <filename>.tcl.
9. The movement of packets can be seen under NAM console.
10. Corresponding trace file and nam file are generated on the desktop.
11. The end to end delay and throughput can be calculated by the following commands:

    awk –f endtoenddelay.awk <tracefile>.tr
    awk –f throughput.awk <tracefile>.tr

A. BUS TOPOLOGY

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(stop)   10.0                ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
set tracefile [open bus.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open bus.nam w]
$ns namtrace-all $namfile

#===================================
#        Nodes Definition
#===================================
#Create 4 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#===================================
#        Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50
$ns duplex-link $n1 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n2 50
$ns duplex-link $n2 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50

#Give node position (for NAM)
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right
$ns duplex-link-op $n2 $n3 orient right
```

```
#=================================
#        Agents Definition
#=================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n3 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#=================================
#        Applications Definition
#=================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#=================================
#        Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam bus.nam &
    exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

B. RING TOPOLOGY

TCL SCRIPT

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(stop)   10.0                ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
set tracefile [open ring.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open ring.nam w]
$ns namtrace-all $namfile

#===================================
#        Nodes Definition
#===================================
#Create 5 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#===================================
#        Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50
$ns duplex-link $n1 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n2 50
$ns duplex-link $n2 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50
$ns duplex-link $n3 $n4 100.0Mb 10ms DropTail
$ns queue-limit $n3 $n4 50
$ns duplex-link $n4 $n0 100.0Mb 10ms DropTail
$ns queue-limit $n4 $n0 50
```

```
#Give node position (for NAM)
$ns duplex-link-op $n0 $n1 orient right-up
$ns duplex-link-op $n1 $n2 orient right-down
$ns duplex-link-op $n2 $n3 orient left-down
$ns duplex-link-op $n3 $n4 orient left-up
$ns duplex-link-op $n4 $n0 orient left-up

#=================================
#        Agents Definition
#=================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#=================================
#        Applications Definition
#=================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#=================================
#        Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam ring.nam &
    exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

C. STAR TOPOLOGY

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(stop)   10.0                ;# time of simulation end

#===================================
#       Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
set tracefile [open star.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open star.nam w]
$ns namtrace-all $namfile

#===================================
#       Nodes Definition
#===================================
#Create 6 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

#===================================
#       Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n1 $n0 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n0 50
$ns duplex-link $n2 $n0 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n0 50
$ns duplex-link $n3 $n0 100.0Mb 10ms DropTail
$ns queue-limit $n3 $n0 50
$ns duplex-link $n4 $n0 100.0Mb 10ms DropTail
$ns queue-limit $n4 $n0 50
$ns duplex-link $n5 $n0 100.0Mb 10ms DropTail
$ns queue-limit $n5 $n0 50

#Give node position (for NAM)
$ns duplex-link-op $n1 $n0 orient right-down
```

```
$ns duplex-link-op $n2 $n0 orient left-down
$ns duplex-link-op $n3 $n0 orient left-up
$ns duplex-link-op $n4 $n0 orient right-up
$ns duplex-link-op $n5 $n0 orient right-up

#=================================
#       Agents Definition
#=================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n1 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#=================================
#       Applications Definition
#=================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#=================================
#       Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
   global ns tracefile namfile
   $ns flush-trace
   close $tracefile
   close $namfile
   exec nam star.nam &
   exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

Calculations:

TOPOLOGY	NUMBER OF NODES	END TO END DELAY	THROUGHPUT
BUS			
RING			
STAR			

Result:

Reasoning Questions

1. What are the different LAN Topologies?
2. What are the advantages and disadvantages of bus topology?
3. Why star topology is more suitable for Networks?
4. What are the drawbacks of ring topology?
5. What are the limitations of mesh topology?

4-QUEUE MANAGEMENT SCHEMES

Aim: To evaluate the performance of Drop-Tail and RED Queue Management Schemes.

Software Required : NS2 / NSG2.1

Operating System : Ubuntu

Procedure:

1. Double click on NSG2.1 icon.
2. Click on scenario and select new wired scenario.
3. Place desired number of nodes and select the link as Drop-Tail or RED and connect the nodes.
4. Select a source node and a destination node and give their agents as TCP and TCP Sink. Then give FTP as application to source node only.
5. Go to parameters and click on done. Go to tcl script and save it on desktop.
6. Go to terminal and change directory to desktop as: cd Desktop.
7. See the contents of desktop by giving command: ls.
8. Run the tcl file by giving command: ns <filename>.tcl.
9. The movement of packets can be seen under NAM console.
10. Corresponding trace file and nam file are generated on the desktop.
11. The end to end delay and throughput can be calculated by the following commands:

    awk –f endtoenddelay.awk <tracefile>.tr
    awk –f throughput.awk <tracefile>.tr

A. DROP-TAIL

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(stop)   20.0                ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile

#===================================
#        Nodes Definition
#===================================
#Create 5 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#===================================
#        Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n1 50
$ns duplex-link $n1 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n2 50
$ns duplex-link $n2 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50
$ns duplex-link $n3 $n4 100.0Mb 10ms DropTail
$ns queue-limit $n3 $n4 50
$ns duplex-link $n4 $n2 100.0Mb 10ms DropTail
$ns queue-limit $n4 $n2 50
```

```
$ns duplex-link $n1 $n4 100.0Mb 10ms DropTail
$ns queue-limit $n1 $n4 50
$ns duplex-link $n0 $n4 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n4 50

#Give node position (for NAM)
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n4 orient left-down
$ns duplex-link-op $n4 $n2 orient right-up
$ns duplex-link-op $n1 $n4 orient right-down
$ns duplex-link-op $n0 $n4 orient right-down

#===================================
#        Agents Definition
#===================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n3 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#===================================
#        Applications Definition
#===================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#===================================
#        Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

B. RED

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(stop)   20.0                ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile

#===================================
#        Nodes Definition
#===================================
#Create 5 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#===================================
#        Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n0 $n1 100.0Mb 10ms RED
$ns queue-limit $n0 $n1 50
$ns duplex-link $n1 $n2 100.0Mb 10ms RED
$ns queue-limit $n1 $n2 50
$ns duplex-link $n2 $n3 100.0Mb 10ms RED
$ns queue-limit $n2 $n3 50
$ns duplex-link $n3 $n4 100.0Mb 10ms RED
$ns queue-limit $n3 $n4 50
$ns duplex-link $n4 $n0 100.0Mb 10ms RED
$ns queue-limit $n4 $n0 50
$ns duplex-link $n1 $n4 100.0Mb 10ms RED
$ns queue-limit $n1 $n4 50
$ns duplex-link $n1 $n3 100.0Mb 10ms RED
```

```
$ns queue-limit $n1 $n3 50

#Give node position (for NAM)
$ns duplex-link-op $n0 $n1 orient right-up
$ns duplex-link-op $n1 $n2 orient right
$ns duplex-link-op $n2 $n3 orient left-down
$ns duplex-link-op $n3 $n4 orient left

$ns duplex-link-op $n4 $n0 orient left-up
$ns duplex-link-op $n1 $n4 orient left-down
$ns duplex-link-op $n1 $n3 orient right-down

#=================================
#        Agents Definition
#=================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2
$ns connect $tcp0 $sink2
$tcp0 set packetSize_ 1500


#=================================
#        Applications Definition
#=================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#=================================
#        Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

Calculations:

QUEUING SCHEME	NUMBER OF NODES	END TO END DELAY	THROUGHPUT
DROP-TAIL			
RED			

Result:

Reasoning Questions

1. What is Droptail Queue Management Scheme?
2. What is RED Queue Managemnt Scheme?
3. What are the drawbacks of Droptail?
4. Draw the flowchart for RED.
5. Which is better Droptail or RED? Justify.

5- SCHEDULING MECHANISMS

Aim: To evaluate the performance of CBQ and FQ scheduling

mechanisms. **Software Required:** NS2 / NSG2.1

Operating System : Ubuntu

Procedure:

1. Double click on NSG2.1 icon.
2. Click on scenario and select new wired scenario.
3. Place desired number of nodes and select the link as CBQ or FQ and connect the nodes.
4. Select a source node and a destination node and give their agents as TCP and TCP Sink. Then give FTP as application to source node only.
5. Go to parameters and click on done. Go to tcl script and save it on desktop.
6. Go to terminal and change directory to desktop as: cd Desktop.
7. See the contents of desktop by giving command: ls.
8. Run the tcl file by giving command: ns <filename>.tcl.
9. The movement of packets can be seen under NAM console.
10. Corresponding trace file and nam file are generated on the desktop.
11. The end to end delay and throughput can be calculated by the following commands:

    awk –f endtoenddelay.awk <tracefile>.tr

    awk –f throughput.awk <tracefile>.tr

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(stop)   20.0                ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile

#===================================
#        Nodes Definition
#===================================
#Create 5 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#===================================
#        Links Definition
#===================================
#Createlinks between nodes
$ns duplex-link $n0 $n3 100.0Mb 10ms FQ
$ns queue-limit $n0 $n3 50
$ns duplex-link $n3 $n1 100.0Mb 10ms FQ
$ns queue-limit $n3 $n1 50
$ns duplex-link $n1 $n2 100.0Mb 10ms FQ
$ns queue-limit $n1 $n2 50
$ns duplex-link $n2 $n4 100.0Mb 10ms FQ
$ns queue-limit $n2 $n4 50
$ns duplex-link $n4 $n0 100.0Mb 10ms FQ
$ns queue-limit $n4 $n0 50
$ns duplex-link $n3 $n2 100.0Mb 10ms FQ
$ns queue-limit $n3 $n2 50
$ns duplex-link $n1 $n0 100.0Mb 10ms FQ
$ns queue-limit $n1 $n0 50
$ns duplex-link $n1 $n4 100.0Mb 10ms FQ
```

```
$ns queue-limit $n1 $n4 50

#Give node position (for NAM)
$ns duplex-link-op $n0 $n3 orient right-up
$ns duplex-link-op $n3 $n1 orient right-down
$ns duplex-link-op $n1 $n2 orient left-down
$ns duplex-link-op $n2 $n4 orient right
$ns duplex-link-op $n4 $n0 orient left-up
$ns duplex-link-op $n3 $n2 orient right-down
$ns duplex-link-op $n1 $n0 orient left

$ns duplex-link-op $n1 $n4 orient right-down

#=================================
#        Agents Definition
#=================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#=================================
#        Applications Definition
#=================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#=================================
#        Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

Calculations:

SCHEDULING MECHANISM	NUMBER OF NODES	END TO END DELAY	THROUGHPUT
CBQ			
FQ			

Result:

Reasoning Questions

1. What is CBQ scheduling mechanism?
2. What is FQ Scheduling mechanism?
3. Which is better CBQ or FQ? Why?
4. Why is mostly implemented in wired Networks?
5. What is meant by a scheduling mechanism?

6-TCP AND UDP PROTOCOLS

Aim: To evaluate the performance of TCP and UDP protocols.

Software Required: NS2 / NSG2.1

Operating System : Ubuntu

Procedure:

1. Double click on NSG2.1 icon.
2. Click on scenario and select new wireless scenario.
3. Place desired number of nodes.
4. Select a source node and a destination node and give their agents as TCP and TCP sink or UDP and NULL. Then give FTP (for TCP) and CBR (for UDP) as application to source node only.
5. Go to parameters and click on done. Go to tcl script and save it on desktop.
6. Go to terminal and change directory to desktop as: cd Desktop.
7. See the contents of desktop by giving command: ls.
8. Run the tcl file by giving command: ns <filename>.tcl.
9. The movement of packets can be seen under NAM console.
10. Corresponding trace file and nam file are generated on the desktop.
11. The end to end delay and throughput and packet delivery ratio can be calculated by the following commands:

    awk –f endtoenddelay.awk <tracefile>.tr

    awk –f throughput.awk <tracefile>.tr

    awk –f packetdeliveryratio.awk <tracefile>.tr

A. TCP PROTOCOL

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     5                          ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      633                        ;# X dimension of topography
set val(y)      812                        ;# Y dimension of topography
set val(stop)   20.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
```

```
            -ifqType       $val(ifq) \
            -ifqLen        $val(ifqlen) \
            -antType       $val(ant) \
            -propType      $val(prop) \
            -phyType       $val(netif) \
            -channel       $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      ON \
            -movementTrace ON

#=================================
#       Nodes Definition
#=================================
#Create 5 nodes
set n0 [$ns node]
$n0 set X_ 362
$n0 set Y_ 712
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 491
$n1 set Y_ 660
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 454
$n2 set Y_ 481
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 358
$n3 set Y_ 544
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 533
$n4 set Y_ 574
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

#=================================
#       Agents Definition
#=================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500
```

```
#=================================
#        Applications Definition
#=================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"

#=================================
#        Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

B. UDP PROTOCOL

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     5                          ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      942                        ;# X dimension of topography
set val(y)      741                        ;# Y dimension of topography
set val(stop)   20.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
```

```
            -ifqType       $val(ifq) \
            -ifqLen        $val(ifqlen) \
            -antType       $val(ant) \
            -propType      $val(prop) \
            -phyType       $val(netif) \
            -channel       $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      ON \
            -movementTrace ON

#=================================
#       Nodes Definition
#=================================
#Create 5 nodes
set n0 [$ns node]
$n0 set X_ 470
$n0 set Y_ 634
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 671
$n1 set Y_ 639
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 842
$n2 set Y_ 641
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 545
$n3 set Y_ 563
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 709
$n4 set Y_ 579
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

#=================================
#       Agents Definition
#=================================
#Setup a UDP connection
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set null1 [new Agent/Null]
$ns attach-agent $n4 $null1
$ns connect $udp0 $null1
$udp0 set packetSize_ 1500
```

```
#=================================
#       Applications Definition
#=================================
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 1000
$cbr0 set rate_ 1.0Mb
$cbr0 set random_ null
$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr0 stop"


#=================================
#       Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

TRACE FILE:

```
s 1.000000000 _0_ AGT  --- 0 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [0] 0 0
r 1.000000000 _0_ RTR  --- 0 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [0] 0 0
s 1.000000000 _0_ RTR  --- 0 AODV 48 [0 0 0 0] -------  [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUEST)
s 1.000535000 _0_ MAC  --- 0 AODV 106 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQU
r 1.001383344 _3_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUE
r 1.001383670 _1_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUE
r 1.001383817 _4_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUE
r 1.001408344 _3_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUE
r 1.001408670 _1_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUE
r 1.001408817 _4_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUE
s 1.001408817 _4_ RTR  --- 0 AODV 44 [0 0 0 0] -------  [4:255 0:255 30 0] [0x4 1 [4 4] 10.000000] (REPLY)
s 1.001550391 _3_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] -------  [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUE
s 1.001625391 _3_ MAC  --- 0 AODV 106 [0 ffffffff 3 800] -------  [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQU
s 1.002289897 _1_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] -------  [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUE
r 1.002473735 _0_ MAC  --- 0 AODV 48 [0 ffffffff 3 800] -------  [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUE
r 1.002473881 _1_ MAC  --- 0 AODV 48 [0 ffffffff 3 800] -------  [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUE
r 1.002473940 _4_ MAC  --- 0 AODV 48 [0 ffffffff 3 800] -------  [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUE
r 1.002498735 _0_ RTR  --- 0 AODV 48 [0 ffffffff 3 800] -------  [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUE
r 1.002498881 _1_ RTR  --- 0 AODV 48 [0 ffffffff 3 800] -------  [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUE
r 1.002498940 _4_ RTR  --- 0 AODV 48 [0 ffffffff 3 800] -------  [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUE
s 1.002683881 _1_ MAC  --- 0 AODV 106 [0 ffffffff 1 800] -------  [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQU
s 1.002683940 _4_ MAC  --- 0 ARP 86 [0 ffffffff 4 806] -------  [REQUEST 4/4 0/0]
D 1.002684451 _2_ MAC  COL 0 ARP 86 [0 ffffffff 4 806] -------  [REQUEST 4/4 0/0]
D 1.002684489 _3_ MAC  COL 0 ARP 86 [0 ffffffff 4 806] -------  [REQUEST 4/4 0/0]
D 1.002684758 _0_ MAC  COL 0 ARP 86 [0 ffffffff 4 806] -------  [REQUEST 4/4 0/0]
D 1.003532372 _3_ MAC  COL 0 AODV 106 [0 ffffffff 1 800] -------  [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQU
D 1.003532451 _2_ MAC  COL 0 AODV 106 [0 ffffffff 1 800] -------  [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQU
D 1.003532552 _0_ MAC  COL 0 AODV 106 [0 ffffffff 1 800] -------  [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQU
s 1.008000000 _0_ AGT  --- 1 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [1] 0 0
r 1.008000000 _0_ RTR  --- 1 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [1] 0 0
s 1.016000000 _0_ AGT  --- 2 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [2] 0 0
r 1.016000000 _0_ RTR  --- 2 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [2] 0 0
s 1.024000000 _0_ AGT  --- 3 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [3] 0 0
r 1.024000000 _0_ RTR  --- 3 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [3] 0 0
s 1.032000000 _0_ AGT  --- 4 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [4] 0 0
r 1.032000000 _0_ RTR  --- 4 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [4] 0 0
s 1.040000000 _0_ AGT  --- 5 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [5] 0 0
r 1.040000000 _0_ RTR  --- 5 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [5] 0 0
s 1.048000000 _0_ AGT  --- 6 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [6] 0 0
r 1.048000000 _0_ RTR  --- 6 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [6] 0 0
s 1.056000000 _0_ AGT  --- 7 cbr 1000 [0 0 0 0] -------  [0:0 4:0 32 0] [7] 0 0
```

Calculations:

TYPE OF PROTOCOL	NUMBER OF NODES	END TO END DELAY	THROUGHPUT	PACKET DELIVERY RATIO
TCP				
UDP				

Result:

Reasoning Questions

1. Distinguish between TCP and UDP protocols.
2. For which applications TCP is better.
3. For which applications UDP is better.
4. Which is more reliable TCP or UDP?
5. In which layer of ISO – OSI model are they implemented?

7-TCP VARIANTS

Aim: To evaluate the performance of TCP, TCP New Reno and TCP Vegas.

Software Required: NS2 / NSG2.1

Operating System : Ubuntu

Procedure:

1. Double click on NSG2.1 icon.
2. Click on scenario and select new wireless scenario.
3. Place desired number of nodes.
4. Select a source node and a destination node and give their agents as TCP and TCP sink or TCP New Reno and TCP sink or TCP Vegas and TCP sink . Then give FTP as application to source node only.
5. Go to parameters and click on done. Go to tcl script and save it on desktop.
6. Go to terminal and change directory to desktop as: cd Desktop.
7. See the contents of desktop by giving command: ls.
8. Run the tcl file by giving command: ns <filename>.tcl.
9. The movement of packets can be seen under NAM console.
10. Corresponding trace file and nam file are generated on the desktop.
11. The end to end delay and throughput and packet delivery ratio can be calculated by the following the commands:
    awk –f endtoenddelay.awk <tracefile>.tr

    awk –f throughput.awk <tracefile>.tr

    awk –f packetdeliveryratio.awk <tracefile>.tr

A. TCP

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     5                          ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      633                        ;# X dimension of topography
set val(y)      812                        ;# Y dimension of topography
set val(stop)   20.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
```

```
            -macType       $val(mac) \
            -ifqType       $val(ifq) \
            -ifqLen        $val(ifqlen) \
            -antType       $val(ant) \
            -propType      $val(prop) \
            -phyType       $val(netif) \
            -channel       $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      ON \
            -movementTrace ON

#===================================
#       Nodes Definition
#===================================
#Create 5 nodes
set n0 [$ns node]
$n0 set X_ 362
$n0 set Y_ 712
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 491
$n1 set Y_ 660
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 454
$n2 set Y_ 481
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 358
$n3 set Y_ 544
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 533
$n4 set Y_ 574
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

#===================================
#       Agents Definition
#===================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp0 $sink1
```

```
$tcp0 set packetSize_ 1500


#===================================
#        Applications Definition
#===================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#===================================
#        Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

.

B. TCP New Reno

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     5                          ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      649                        ;# X dimension of topography
set val(y)      758                        ;# Y dimension of topography
set val(stop)   20.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
```

```
            -ifqLen       $val(ifqlen) \
            -antType      $val(ant) \
            -propType     $val(prop) \
            -phyType      $val(netif) \
            -channel      $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      ON \
            -movementTrace ON

#=================================
#       Nodes Definition
#=================================
#Create 5 nodes
set n0 [$ns node]
$n0 set X_ 311
$n0 set Y_ 629
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 421
$n1 set Y_ 658
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 549
$n2 set Y_ 581
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 416
$n3 set Y_ 406
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 444
$n4 set Y_ 574
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

#=================================
#       Agents Definition
#=================================
#Setup a TCP/Newreno connection
set tcp0 [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n2 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500
```

```
#==================================
#        Applications Definition
#==================================
#Setup a FTP Application over TCP/Newreno connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"

#==================================
#        Termination
#==================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

C. TCP Vegas

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     5                          ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      831                        ;# X dimension of topography
set val(y)      748                        ;# Y dimension of topography
set val(stop)   20.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
```

```
            -ifqLen       $val(ifqlen) \
            -antType      $val(ant) \
            -propType     $val(prop) \
            -phyType      $val(netif) \
            -channel      $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      ON \
            -movementTrace ON

#=================================
#       Nodes Definition
#=================================
#Create 5 nodes
set n0 [$ns node]
$n0 set X_ 355
$n0 set Y_ 621
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 731
$n1 set Y_ 648
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 556
$n2 set Y_ 647
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 630
$n3 set Y_ 370
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 563
$n4 set Y_ 486
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

#=================================
#       Agents Definition
#=================================
#Setup a TCP/Vegas connection
set tcp0 [new Agent/TCP/Vegas]
$ns attach-agent $n0 $tcp0
set sink2 [new Agent/TCPSink]
$ns attach-agent $n3 $sink2
$ns connect $tcp0 $sink2
$tcp0 set packetSize_ 1500
```

```
#=================================
#        Applications Definition
#=================================
#Setup a FTP Application over TCP/Vegas connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 8.0 "$ftp0 stop"

#=================================
#        Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

TRACE FILE:

```
s 1.000000000 _0_ AGT  --- 0 tcp 1500 [0 0 0 0] -------  [0:0 3:0 32 0] [0 0] 0 0
r 1.000000000 _0_ RTR  --- 0 tcp 1500 [0 0 0 0] -------  [0:0 3:0 32 0] [0 0] 0 0
s 1.000000000 _0_ RTR  --- 0 AODV 48 [0 0 0 0] -------  [0:255 -1:255 30 0] [0x2 1 1 [3 0] [0 4]] (REQUEST)
s 1.000635000 _0_ MAC  --- 0 AODV 106 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [3 0] [0 4]] (REQUEST)
r 1.001483676 _2_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [3 0] [0 4]] (REQUEST)
r 1.001483827 _4_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [3 0] [0 4]] (REQUEST)
r 1.001508676 _2_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [3 0] [0 4]] (REQUEST)
r 1.001508827 _4_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] -------  [0:255 -1:255 30 0] [0x2 1 1 [3 0] [0 4]] (REQUEST)
s 1.005920409 _2_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] -------  [2:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
s 1.006155409 _2_ MAC  --- 0 AODV 106 [0 ffffffff 2 800] -------  [2:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.007003947 _4_ MAC  --- 0 AODV 48 [0 ffffffff 2 800] -------  [2:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.007003993 _1_ MAC  --- 0 AODV 48 [0 ffffffff 2 800] -------  [2:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.007004085 _0_ MAC  --- 0 AODV 48 [0 ffffffff 2 800] -------  [2:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.007028947 _4_ RTR  --- 0 AODV 48 [0 ffffffff 2 800] -------  [2:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.007028993 _1_ RTR  --- 0 AODV 48 [0 ffffffff 2 800] -------  [2:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.007029085 _0_ RTR  --- 0 AODV 48 [0 ffffffff 2 800] -------  [2:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
s 1.009375010 _1_ RTR  --- 0 AODV 48 [0 ffffffff 2 800] -------  [1:255 -1:255 28 0] [0x2 3 1 [3 0] [0 4]] (REQUEST)
s 1.009500381 _4_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] -------  [4:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
s 1.009810010 _1_ MAC  --- 0 AODV 106 [0 ffffffff 1 800] -------  [1:255 -1:255 28 0] [0x2 3 1 [3 0] [0 4]] (REQUEST)
r 1.010658594 _2_ MAC  --- 0 AODV 48 [0 ffffffff 1 800] -------  [1:255 -1:255 28 0] [0x2 3 1 [3 0] [0 4]] (REQUEST)
r 1.010658788 _4_ MAC  --- 0 AODV 48 [0 ffffffff 1 800] -------  [1:255 -1:255 28 0] [0x2 3 1 [3 0] [0 4]] (REQUEST)
r 1.010683594 _2_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] -------  [1:255 -1:255 28 0] [0x2 3 1 [3 0] [0 4]] (REQUEST)
r 1.010683788 _4_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] -------  [1:255 -1:255 28 0] [0x2 3 1 [3 0] [0 4]] (REQUEST)
s 1.010748788 _4_ MAC  --- 0 AODV 106 [0 ffffffff 4 800] -------  [4:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.011597235 _3_ MAC  --- 0 AODV 48 [0 ffffffff 4 800] -------  [4:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.011597325 _2_ MAC  --- 0 AODV 48 [0 ffffffff 4 800] -------  [4:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.011597566 _1_ MAC  --- 0 AODV 48 [0 ffffffff 4 800] -------  [4:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.011597615 _0_ MAC  --- 0 AODV 48 [0 ffffffff 4 800] -------  [4:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
s 1.011622235 _3_ RTR  --- 0 AODV 48 [0 ffffffff 4 800] -------  [4:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
s 1.011622235 _3_ RTR  --- 0 AODV 44 [0 0 0 0] -------  [3:255 0:255 30 4] [0x4 1 [3 4] 10.000000] (REPLY)
r 1.011622325 _2_ RTR  --- 0 AODV 48 [0 ffffffff 4 800] -------  [4:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.011622566 _1_ RTR  --- 0 AODV 48 [0 ffffffff 4 800] -------  [4:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
r 1.011622615 _0_ RTR  --- 0 AODV 48 [0 ffffffff 4 800] -------  [4:255 -1:255 29 0] [0x2 2 1 [3 0] [0 4]] (REQUEST)
s 1.012217235 _3_ MAC  --- 0 ARP 86 [0 ffffffff 3 806] -------  [REQUEST 3/3 0/4]
r 1.012905681 _4_ MAC  --- 0 ARP 28 [0 ffffffff 3 806] -------  [REQUEST 3/3 0/4]
s 1.013320681 _4_ MAC  --- 0 RTS 44 [52e 3 4 0]
r 1.013673128 _3_ MAC  --- 0 RTS 44 [52e 3 4 0]
s 1.013683128 _3_ MAC  --- 0 CTS 38 [3f4 4 0 0]
r 1.013987574 _4_ MAC  --- 0 CTS 38 [3f4 4 0 0]
s 1.013997574 _4_ MAC  --- 0 ARP 86 [13a 3 4 806] -------  [REPLY 4/4 3/3]
r 1.014686021 _3_ MAC  --- 0 ARP 28 [13a 3 4 806] -------  [REPLY 4/4 3/3]
```

Calculations:

TYPE OF PROTOCOL	NUMBER OF NODES	END TO END DELAY	THROUGHPUT	PACKET DELIVERY RATIO
TCP				
TCP New Reno				
TCP Vegas				

Result:

Reasoning Questions

1. What are the different variants of TCP?
2. What are the advantages of TCP?
3. What is TCP Reno?
4. What is TCP Vegas?
5. How TCP Reno and TCP Vegas are different from TCP?

8 &9 -TCP ROUTING PROTOCOLS

Aim: To evaluate the performance of AODV, DSDV and DSR routing protocols.

Software Required: NS2 / NSG2.1

Operating System : Ubuntu

Procedure:

1. Double click on NSG2.1 icon.
2. Click on scenario and select new wireless scenario.
3. Place desired number of nodes.
4. Select a source node and a destination node and give their agents as TCP and TCP sink or UDP and NULL. Then give FTP (for TCP) and CBR (for UDP) as application to source node only.
5. Go to parameters and select the routing protocol as AODV or DSDV or DSR and click on done. Go to tcl script and save it on desktop.
6. Go to terminal and change directory to desktop as: cd Desktop.
7. See the contents of desktop by giving command: ls.
8. Run the tcl file by giving command: ns <filename>.tcl.
9. The movement of packets can be seen under NAM console.
10. Corresponding trace file and nam file are generated on the desktop.
11. The end to end delay and throughput and packet delivery ratio can be calculated by the following commands:
    awk –f endtoenddelay.awk <tracefile>.tr

    awk –f throughput.awk <tracefile>.tr

    awk –f packetdeliveryratio.awk <tracefile>.tr

A. AODV

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     5                          ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      633                        ;# X dimension of topography
set val(y)      812                        ;# Y dimension of topography
set val(stop)   20.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
```

```
            -ifqLen       $val(ifqlen) \
            -antType      $val(ant) \
            -propType     $val(prop) \
            -phyType      $val(netif) \
            -channel      $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      ON \
            -movementTrace ON

#=================================
#       Nodes Definition
#=================================
#Create 5 nodes
set n0 [$ns node]
$n0 set X_ 362
$n0 set Y_ 712
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 491
$n1 set Y_ 660
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 454
$n2 set Y_ 481
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 358
$n3 set Y_ 544
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 533
$n4 set Y_ 574
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

#=================================
#       Agents Definition
#=================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500
```

```
#=================================
#        Applications Definition
#=================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#=================================
#        Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

B. DSDV

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     10                         ;# number of mobilenodes
set val(rp)     DSDV                       ;# routing protocol
set val(x)      975                        ;# X dimension of topography
set val(y)      824                        ;# Y dimension of topography
set val(stop)   20.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
                -ifqLen        $val(ifqlen) \
                -antType       $val(ant) \
                -propType      $val(prop) \
```

```
            -phyType       $val(netif) \
            -channel       $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      ON \
            -movementTrace ON

#================================
#       Nodes Definition
#================================
#Create 10 nodes
set n0 [$ns node]
$n0 set X_ 458
$n0 set Y_ 673
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 591
$n1 set Y_ 686
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 574
$n2 set Y_ 724
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 564
$n3 set Y_ 526
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 656
$n4 set Y_ 611
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
set n5 [$ns node]
$n5 set X_ 764
$n5 set Y_ 651
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20
set n6 [$ns node]
$n6 set X_ 735
$n6 set Y_ 722
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20
set n7 [$ns node]
$n7 set X_ 875
$n7 set Y_ 573
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20
set n8 [$ns node]
```

```
$n8 set X_ 700
$n8 set Y_ 526
$n8 set Z_ 0.0
$ns initial_node_pos $n8 20
set n9 [$ns node]
$n9 set X_ 804
$n9 set Y_ 501
$n9 set Z_ 0.0
$ns initial_node_pos $n9 20

#===================================
#        Agents Definition
#===================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n7 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#===================================
#        Applications Definition
#===================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#===================================
#        Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

C. DSR

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     5                          ;# number of mobilenodes
set val(rp)     DSR                        ;# routing protocol
set val(x)      955                        ;# X dimension of topography
set val(y)      763                        ;# Y dimension of topography
set val(stop)   20.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
                -ifqLen        $val(ifqlen) \
                -antType       $val(ant) \
                -propType      $val(prop) \
```

```
            -phyType       $val(netif) \
            -channel       $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      ON \
            -movementTrace ON

#================================
#       Nodes Definition
#================================
#Create 5 nodes
set n0 [$ns node]
$n0 set X_ 507
$n0 set Y_ 636
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 691
$n1 set Y_ 655
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 855
$n2 set Y_ 663
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 716
$n3 set Y_ 553
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 812
$n4 set Y_ 573
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

#================================
#       Agents Definition
#================================
#Setup a UDP connection
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set null2 [new Agent/Null]
$ns attach-agent $n2 $null2
$ns connect $udp0 $null2
$udp0 set packetSize_ 1500


#================================
#       Applications Definition
#================================
```

```
#Setup a CBR Application over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 1000
$cbr0 set rate_ 1.0Mb
$cbr0 set random_ null
$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr0 stop"


#=================================
#        Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

TRACE FILE:

```
Sconfig 0.00000 tap: on snoop: rts? on errs? on
Sconfig 0.00000 salvage: on !bd replies? on
Sconfig 0.00000 grat error: on grat reply: on
Sconfig 0.00000 $reply for props: on ring 0 search: on
Sconfig 0.00000 using MOBICACHE
s 1.000000000 _0_ AGT  --- 0 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [0] 0 0
r 1.000000000 _0_ RTR  --- 0 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [0] 0 0
s 1.001191771 _0_ RTR  --- 1 DSR 32 [0 0 0 0] ------- [0:255 2:255 32 0] 1 [1 1] [0 1 0 0->0] [0 0 0 0->0]
s 1.001286771 _0_ MAC  --- 1 DSR 90 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 1 [1 1] [0 1 0 0->0] [0 0 0 0->0]
r 1.002007388 _1_ MAC  --- 1 DSR 32 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 1 [1 1] [0 1 0 0->0] [0 0 0 0->0]
r 1.002007521 _3_ MAC  --- 1 DSR 32 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 1 [1 1] [0 1 0 0->0] [0 0 0 0->0]
r 1.002032388 _1_ RTR  --- 1 DSR 32 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 1 [1 1] [0 1 0 0->0] [0 0 0 0->0]
r 1.002032521 _3_ RTR  --- 1 DSR 32 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 1 [1 1] [0 1 0 0->0] [0 0 0 0->0]
s 1.008000000 _0_ AGT  --- 2 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [1] 0 0
r 1.008000000 _0_ RTR  --- 2 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [1] 0 0
s 1.016000000 _0_ AGT  --- 4 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [2] 0 0
r 1.016000000 _0_ RTR  --- 4 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [2] 0 0
s 1.024000000 _0_ AGT  --- 9 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [3] 0 0
r 1.024000000 _0_ RTR  --- 9 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [3] 0 0
s 1.032000000 _0_ AGT  --- 11 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [4] 0 0
r 1.032000000 _0_ RTR  --- 11 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [4] 0 0
s 1.040000000 _0_ AGT  --- 13 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [5] 0 0
r 1.040000000 _0_ RTR  --- 13 cbr 1000 [0 0 0 0] ------- [0:0 2:0 32 0] [5] 0 0
s 1.040968009 _0_ RTR  --- 12 DSR 32 [0 0 0 0] ------- [0:255 2:255 32 0] 1 [1 2] [0 2 0 0->0] [0 0 0 0->0]
s 1.041523009 _0_ MAC  --- 12 DSR 90 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 1 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.042243625 _1_ MAC  --- 12 DSR 32 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 1 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.042243758 _3_ MAC  --- 12 DSR 32 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 1 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.042268625 _1_ RTR  --- 12 DSR 32 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 1 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.042268758 _3_ RTR  --- 12 DSR 32 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 1 [1 2] [0 2 0 0->0] [0 0 0 0->0]
f 1.043853762 _1_ RTR  --- 12 DSR 48 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
s 1.044108762 _1_ MAC  --- 12 DSR 106 [0 ffffffff 1 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
f 1.044624165 _3_ RTR  --- 12 DSR 48 [0 ffffffff 0 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.044957112 _3_ MAC  --- 12 DSR 48 [0 ffffffff 1 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.044957250 _4_ MAC  --- 12 DSR 48 [0 ffffffff 1 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.044957310 _2_ MAC  --- 12 DSR 48 [0 ffffffff 1 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.044957379 _0_ MAC  --- 12 DSR 48 [0 ffffffff 1 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.044982112 _3_ RTR  --- 12 DSR 48 [0 ffffffff 1 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.044982250 _4_ RTR  --- 12 DSR 48 [0 ffffffff 1 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.044982310 _2_ RTR  --- 12 DSR 48 [0 ffffffff 1 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
r 1.044982379 _0_ RTR  --- 12 DSR 48 [0 ffffffff 1 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
s 1.045147112 _3_ MAC  --- 12 DSR 106 [0 ffffffff 3 800] ------- [0:255 2:255 32 0] 2 [1 2] [0 2 0 0->0] [0 0 0 0->0]
```

Calculations:

TYPE OF ROUTING PROTOCOL	NUMBER OF NODES	END TO END DELAY	THROUGHPUT	PACKET DELIVERY RATIO
AODV				
DSDV				
DSR				

Result:

Reasoning Questions

1. What are the two categories of routing protocols in MANETs?
2. What are the advantages and disadvantages of on-demand routing protocols?
3. Explain AODV routing protocol.
4. Explain DSRV routing protocol.
5. Explain DSR routing protocol.

10-IEEE 802.11 AND WPAN

Aim: To evaluate the performance of IEEE 802.11 and WPAN (IEEE 802.15.4).

Software Required: NS2 / NSG2.1

Operating System : Ubuntu

Procedure:

1. Double click on NSG2.1 icon.
2. Click on scenario and select new wireless scenario.
3. Place desired number of nodes.
4. Select a source node and a destination node and give their agents as TCP and TCP sink. Then give FTP as application to source node only.
5. Go to parameters and click on done. Go to tcl script and save it on desktop. Double click on tcl file on desktop. For IEEE 802.11 the network interface should be: phy/wireless phy and MAC type should be: Mac/802_11. For WPAN the network interface should be: Phy/WirelessPhy/802_15_4 and MAC type should be: Mac/802_15_4.
6. Go to terminal and change directory to desktop as: cd Desktop.
7. See the contents of desktop by giving command: ls.
8. Run the tcl file by giving command: ns <filename>.tcl.
9. The movement of packets can be seen under NAM console.
10. Corresponding trace file and nam file are generated on the desktop.
11. The end to end delay and throughput and packet delivery ratio can be calculated by the following commands:

    awk –f endtoenddelay.awk <tracefile>.tr

    awk –f throughput.awk <tracefile>.tr

    awk –f packetdeliveryratio.awk <tracefile>.tr

A. IEEE 802.11

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     5                          ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      633                        ;# X dimension of topography
set val(y)      812                        ;# Y dimension of topography
set val(stop)   20.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
                -ifqLen        $val(ifqlen) \
                -antType       $val(ant) \
```

```
            -propType      $val(prop) \
            -phyType       $val(netif) \
            -channel       $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      ON \
            -movementTrace ON

#=================================
#       Nodes Definition
#=================================
#Create 5 nodes
set n0 [$ns node]
$n0 set X_ 362
$n0 set Y_ 712
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 491
$n1 set Y_ 660
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 454
$n2 set Y_ 481
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 358
$n3 set Y_ 544
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 533
$n4 set Y_ 574
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

#=================================
#       Agents Definition
#=================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#=================================
#       Applications Definition
```

```
#=================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#=================================
#       Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
   global ns tracefile namfile
   $ns flush-trace
   close $tracefile
   close $namfile
   exec nam out.nam &
   exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
   $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

B. WPAN (IEEE 802.15.4)

TCL SCRIPT:

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy/802_15_4          ;# network interface type
set val(mac)    Mac/802_15_4               ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     5                          ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      727                        ;# X dimension of topography
set val(y)      684                        ;# Y dimension of topography
set val(stop)   10.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open in20.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out20.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
                -ifqLen        $val(ifqlen) \
                -antType       $val(ant) \
                -propType      $val(prop) \
```

```
            -phyType       $val(netif) \
            -channel       $chan \
            -topoInstance  $topo \
            -agentTrace    ON \
            -routerTrace   ON \
            -macTrace      ON \
            -movementTrace ON

#=================================
#       Nodes Definition
#=================================
#Create 5 nodes
set n0 [$ns node]
$n0 set X_ 322
$n0 set Y_ 568
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 446
$n1 set Y_ 584
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 473
$n2 set Y_ 372
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 311
$n3 set Y_ 368
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 627
$n4 set Y_ 410
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

#=================================
#       Agents Definition
#=================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#=================================
#       Applications Definition
#=================================
```

```
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"


#=================================
#       Termination
#=================================
#Define a 'finish' procedure
proc finish {} {
   global ns tracefile namfile
   $ns flush-trace
   close $tracefile
   close $namfile
   exec nam out20.nam &
   exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
   $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

TRACE FILE:

```
s 1.000000000 _0_ AGT  --- 0 tcp 40 [0 0 0 0] ------- [0:0 4:0 32 0] [0 0] 0 0
r 1.000000000 _0_ RTR  --- 0 tcp 40 [0 0 0 0] ------- [0:0 4:0 32 0] [0 0] 0 0
s 1.000000000 _0_ RTR  --- 0 AODV 48 [0 0 0 0] ------- [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUEST)
s 1.002585000 _0_ MAC  --- 0 AODV 55 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUEST)
r 1.005177417 _1_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUEST)
r 1.005177668 _3_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUEST)
r 1.005177825 _2_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUEST)
r 1.005202417 _1_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUEST)
r 1.005202668 _3_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUEST)
r 1.005202825 _2_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [4 0] [0 4]] (REQUEST)
s 1.005630613 _3_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
s 1.005975613 _3_ MAC  --- 0 AODV 55 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
s 1.006843871 _1_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.008568153 _2_ MAC  --- 0 AODV 48 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.008568281 _0_ MAC  --- 0 AODV 48 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.008593153 _2_ RTR  --- 0 AODV 48 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.008593281 _0_ RTR  --- 0 AODV 48 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
s 1.011156871 _1_ MAC  --- 0 AODV 55 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
s 1.011603501 _2_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [2:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.013749288 _0_ MAC  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.013749583 _2_ MAC  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.013774288 _0_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.013774583 _2_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
s 1.017324501 _2_ MAC  --- 0 AODV 55 [0 ffffffff 2 800] ------- [2:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.019917030 _4_ MAC  --- 0 AODV 48 [0 ffffffff 2 800] ------- [2:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.019917041 _3_ MAC  --- 0 AODV 48 [0 ffffffff 2 800] ------- [2:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.019917214 _1_ MAC  --- 0 AODV 48 [0 ffffffff 2 800] ------- [2:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.019917326 _0_ MAC  --- 0 AODV 48 [0 ffffffff 2 800] ------- [2:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.019942030 _4_ RTR  --- 0 AODV 48 [0 ffffffff 2 800] ------- [2:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
s 1.019942030 _4_ RTR  --- 0 AODV 44 [0 0 0 0] ------- [4:255 0:255 30 2] [0x4 1 [4 4] 10.000000] (REPLY)
r 1.019942041 _3_ RTR  --- 0 AODV 48 [0 ffffffff 2 800] ------- [2:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.019942214 _1_ RTR  --- 0 AODV 48 [0 ffffffff 2 800] ------- [2:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
r 1.019942326 _0_ RTR  --- 0 AODV 48 [0 ffffffff 2 800] ------- [2:255 -1:255 29 0] [0x2 2 1 [4 0] [0 4]] (REQUEST)
s 1.020587030 _4_ MAC  --- 0 ARP 35 [0 ffffffff 4 806] ------- [REQUEST 4/4 0/2]
r 1.022539559 _2_ MAC  --- 0 ARP 28 [0 ffffffff 4 806] ------- [REQUEST 4/4 0/2]
s 1.023184559 _2_ MAC  --- 0 ARP 35 [0 4 2 806] ------- [REPLY 2/2 4/4]
s 1.024689088 _4_ MAC  --- 0 ACK 5 [0 2 4 0]
r 1.025041616 _2_ MAC  --- 0 ACK 5 [0 2 4 0]
r 1.025681088 _4_ MAC  --- 0 ARP 28 [0 4 2 806] ------- [REPLY 2/2 4/4]
s 1.027926088 _4_ MAC  --- 0 AODV 51 [0 2 4 800] ------- [4:255 0:255 30 2] [0x4 1 [4 4] 10.000000] (REPLY)
s 1.029942616 _2_ MAC  --- 0 ACK 5 [0 4 2 0]
```

Calculations:

MAC TYPE	NUMBER OF NODES	END TO END DELAY	THROUGHPUT	PACKET DELIVERY RATIO
IEEE 802.11				
IEEE 802.15.4				

Result:

Reasoning Questions

1. IEEE 802.11 is the standard for which networks?
2. What are the challenges in WLANs?
3. What are the advantages of PANs?
4. What are the commercial alternatives of PANs?
5. What are “hidden terminal” and “exposed terminal” Problems in WLANs?

12-CAPTURE AND ANALYSIS OF TCP AND IP PACKETS

Aim: To capture and analyze TCP and IP packets through Wireshark software.

Software Required : Wireshark

Procedure:

1 .Open Wireshark-packet capture software.

2. Open capture tab.

3. Select interface.

4. Select on existing Ethernet card.

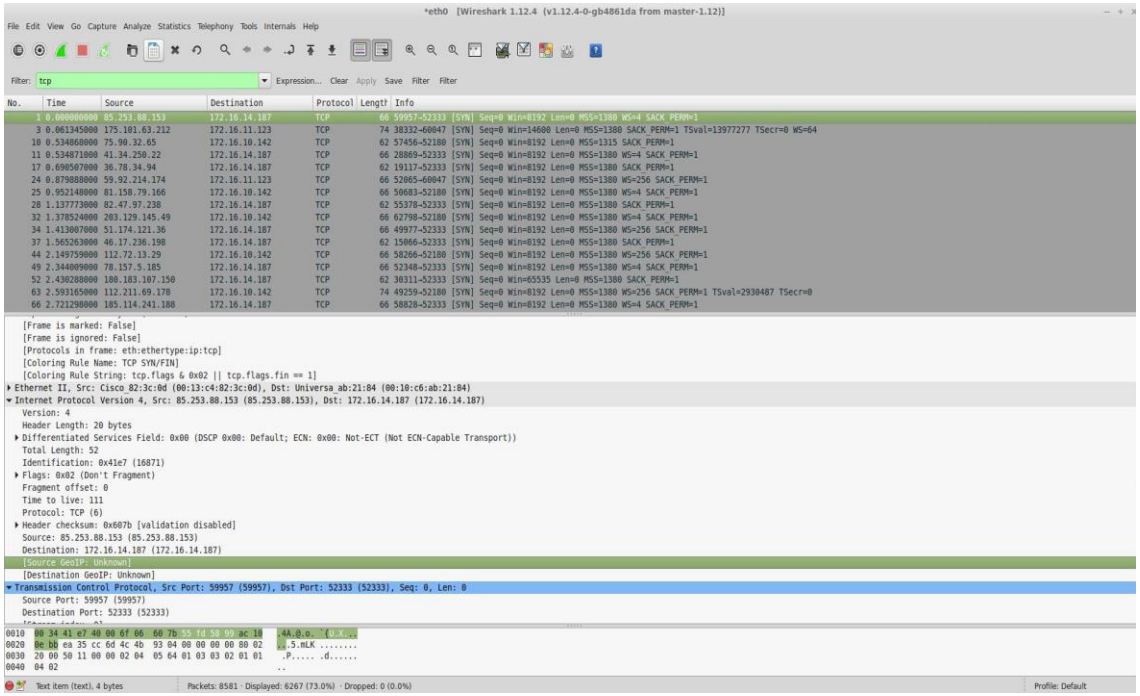
5. Click on start capture area over some period of time browse the internet.

6. Click on stop capture button.

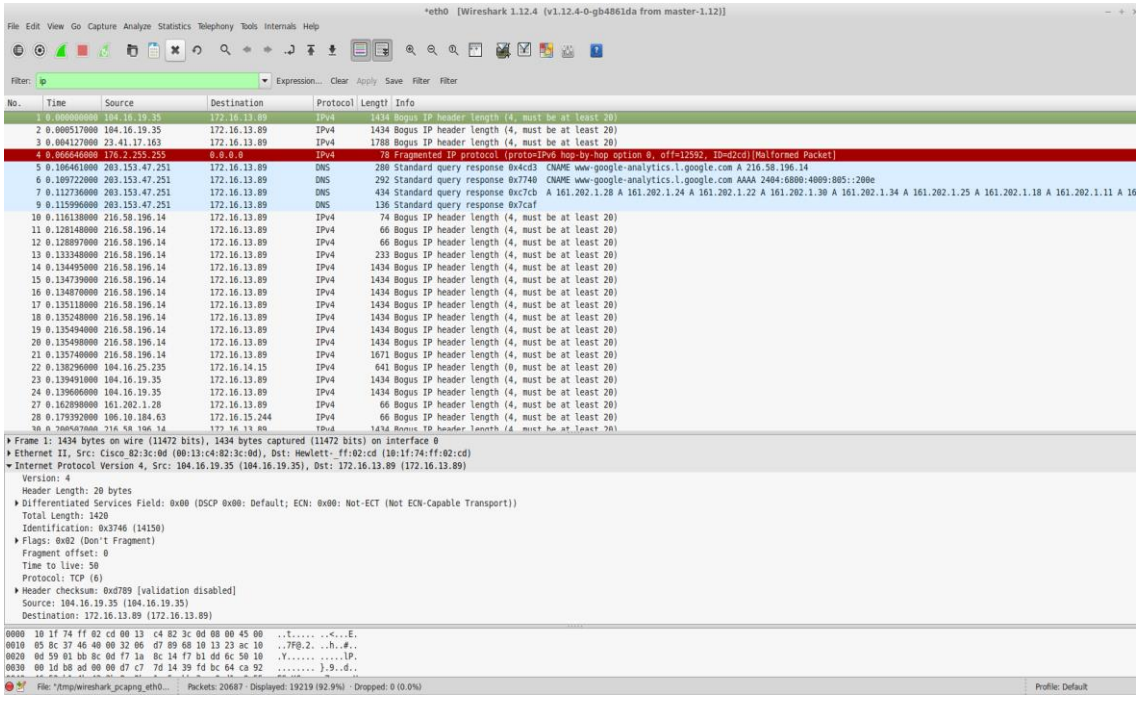
7. For TCP: a) Click on analyze button, display filters and select TCP only and click ok.
   b) All the TCP packets will be listed.
   c) Select the required TCP packets and analyze the packet by right clicking.
   d) Go to transmission control protocol, src port and dst port.
   e) Check various fields like sequence numbers, ack number, header length, flags, etc.

8. For IP: a) Click on analyze button and select IP only and click ok.
   b) All the IP packets will be listed.
   c) Select the required IP packet and analyze the packet by right clicking.
   d) Go to IPV4.
   e) Check various fields like source address, destination address, header length, differentiated services, flags, checksum, etc.

TCP PACKETS IN ETHERNET



IP PACKETS IN ETHERNET



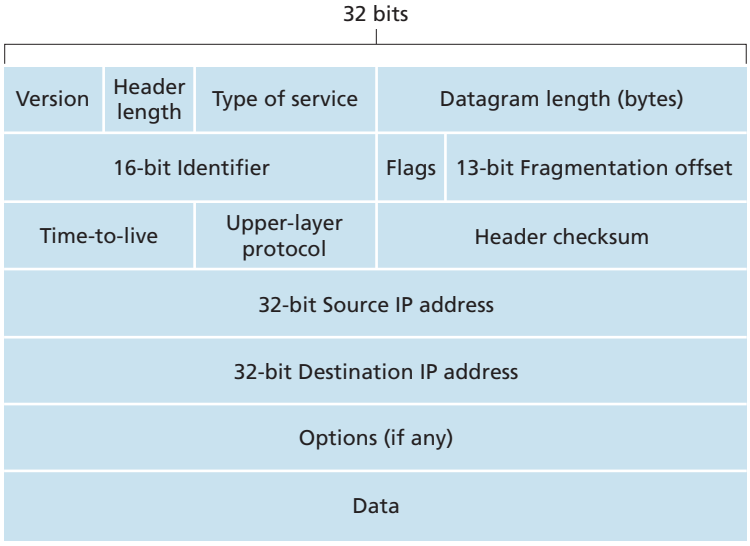
Result:

Reasoning Questions

1. What is the Role of TCP/IP?
2. What is the difference between connectionless and connection oriented protocol?
3. What is IP?
4. What are the disadvantages of TCP.
5. What is the format for IP Header?

IPv4 Analysis

IPv4 datagram structure:



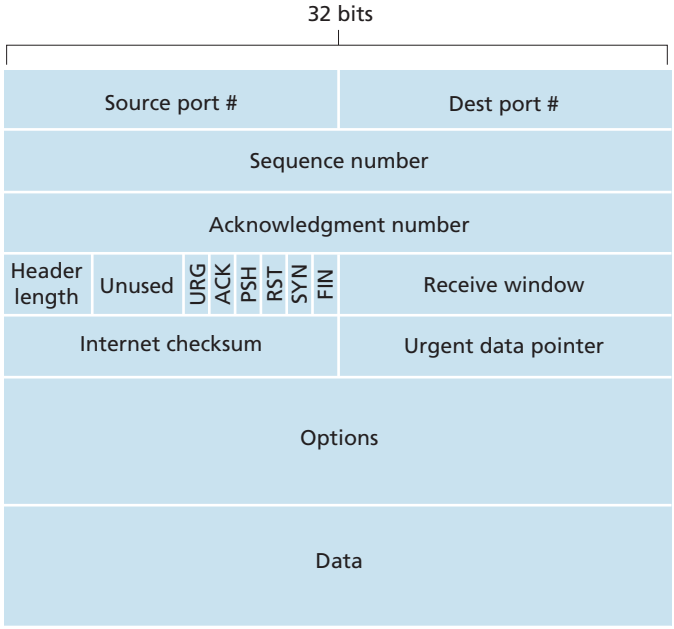
Observation table:

- Visit **www.google.com** from your browser and note down the details of any **IP datagram** that you observed

Field Name	Field length (bits)	Field Value
Version		
Header length		
Type of service		
Datagram length		
16-bit identifier		
Flags		
13-bit fragmentation offset		
Time-to-live		
Upper-layer protocol		
Header checksum		
32-bit source IP address		
32-bit destination IP address		
Options (if any)		
Data		

TCP Analysis

TCP segment structure:



Observation table:

- Visit **www.google.com** from your browser and note down the details of any **TCP segment** that you observed

Field Name	Field length (bits)	Field Value
Source port		
Dest port		
Sequence number		
Acknowledgement number		
Header length		
Flags (URG, ACK, PSH, RST, SYN, FIN)		
Receive window		
Internet checksum		
Urgent data pointer		
Options		
Data		

13-SIMULATION AND ANALYSIS OF ICMP AND IGMP PACKETS

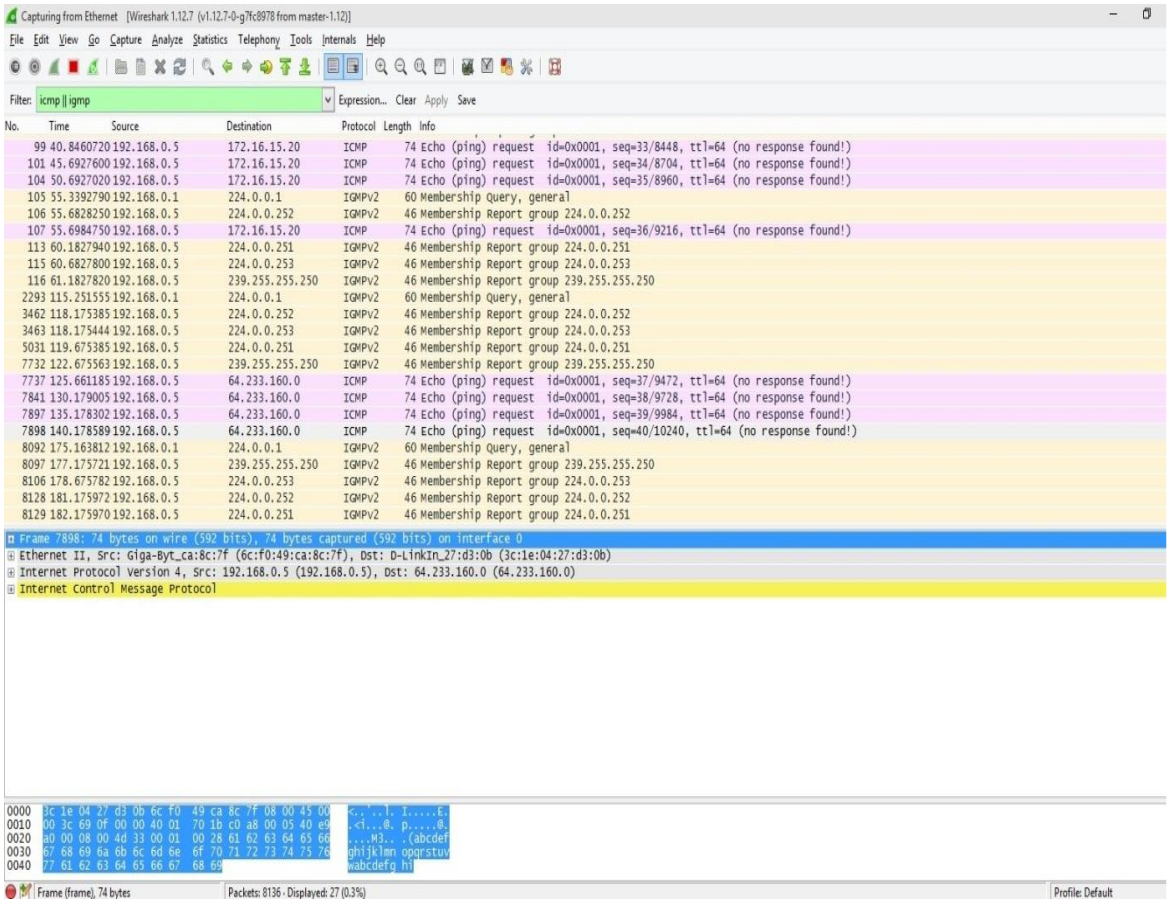
Aim: To simulate and analyze ICMP and IGMP packets through wireshark Software

Software Required: Wireshark

Procedure:

1. Open wireshark-packet capture software.
2. Open capture tab.
3. Select interface.
4. Select an existing Ethernet card.
5. Click on start capture.
6. Open terminal (LINUX) or cmd (WINDOWS) type following command:
7. "ping <address>".
8. Click on stop capture button.
9. Click on analyze button, display filters, select now.
10. Click on filter expression and enter "icmp/igmp".
11. All the ICMP and IGMP packets will be listed.
12. Select the required ICMP and IGMP packets and analyze the packet by right clicking.
13. Analyze the ICMP or IGMP packet in full.

ICMP AND IGMP PACKETS IN ETHERNET



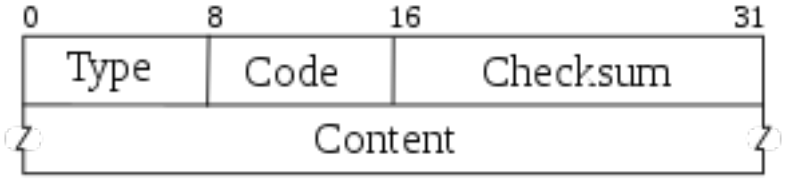
Result:

Reasoning Questions

1. What is ICMP?
2. What is header length of ICMP message?
3. What is the difference ICMP and IGMP?
4. How many times a membership report is send in IGMP?
5. In which packet IGMP packet is carried?

ICMP Analysis

ICMP datagram structure:



Observation table:

- Ping **www.google.com** from your terminal and note down the details of any **ICMP datagram** that you observed

Field Name	Field length (bits)	Field Value
Type		
Code		
Checksum		
Content		

14- ANALYSIS THE PROTOCOLS OF SCTP, ARP, NETBIOS, IPX VINES

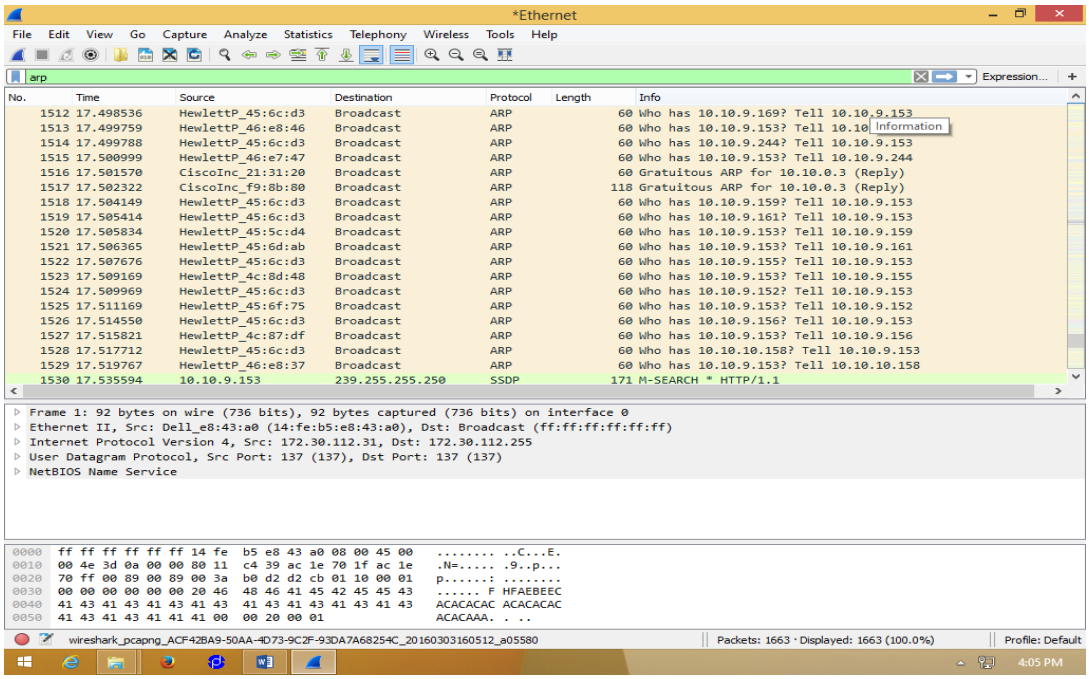
Aim: Analyze the protocols of SCTP, ARP, NetBIOS and IPX Vines packets through Wireshark Software

Software Required: Wireshark

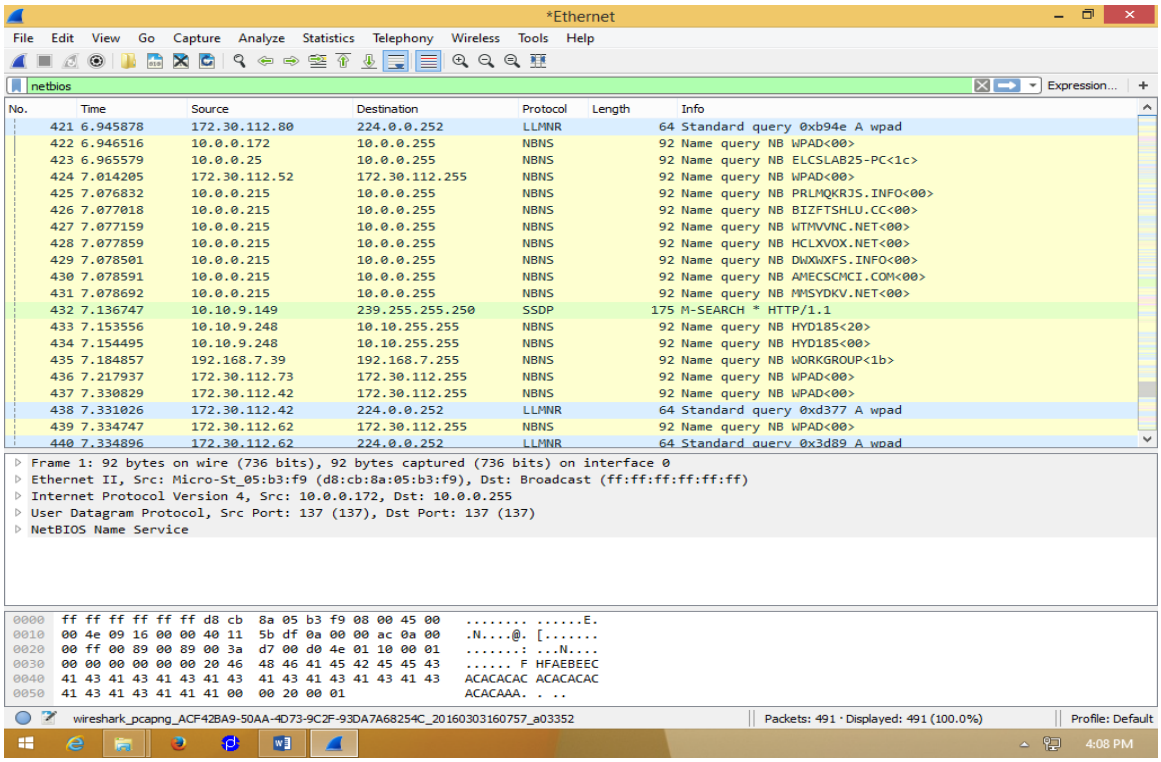
Procedure:

1. Open Wireshark-packet capture software.
2. Open capture tab.
3. Select interface.
4. Select an existing Ethernet card.
5. Click on start capture.
6. Open google and browse any streaming video files
7. Click on stop capture button.
8. Click on analyze button, display filters, select now.
9. Click on filter expression and enter “sctp/arp/NetBios(NBNS),IPX Vines”.
10. All the sctp, arp, NetBIOS(NBNS) and IPX Vines packets will be listed.
11. Select the sctp, arp, NetBIOS(NBNS) and IPX Vines required packets and analyze the packet by right clicking.
12. Note down the details listed.

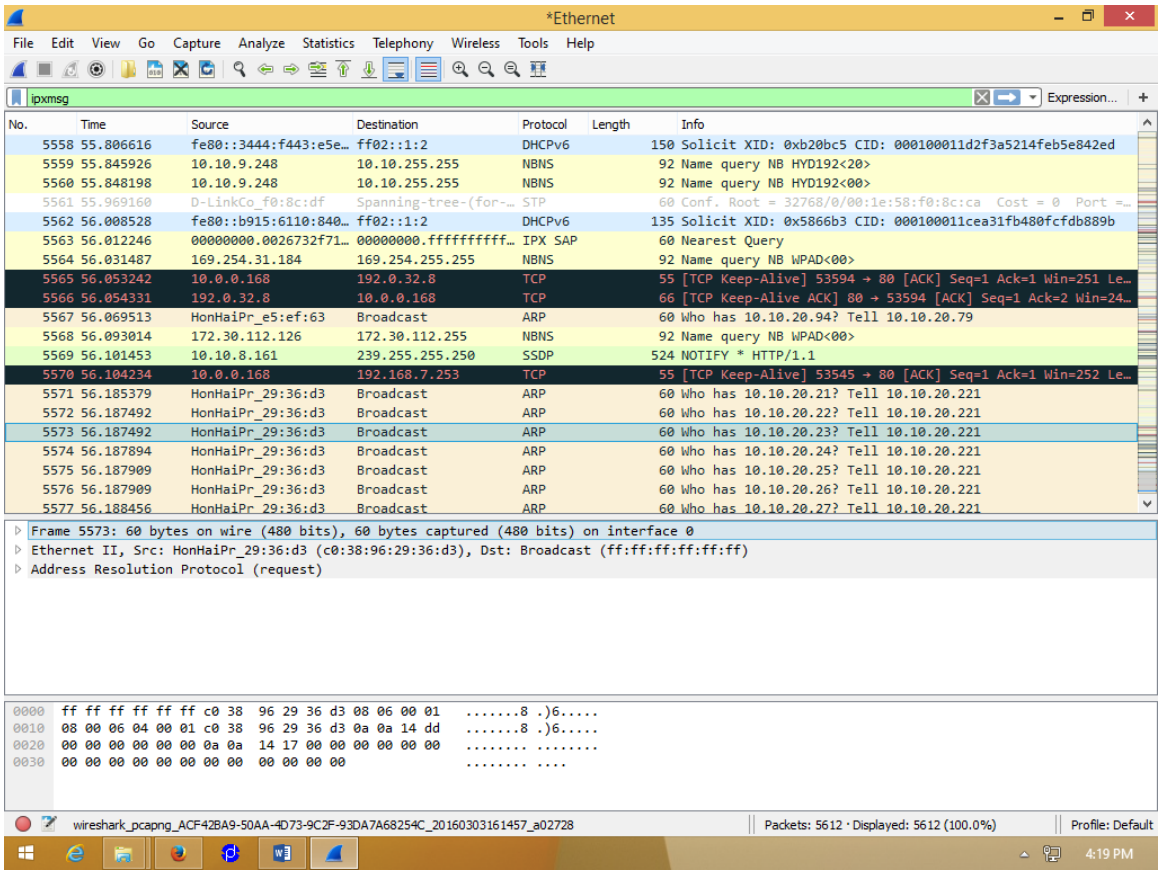
ARP PACKETS IN ETHERNET



NetBIOS PACKETS IN ETHERNET



IPX PACKETS IN ETHERNET
(Exchanging of Messages)



Result:

Reasoning Questions

1. What are features of SCTP?
2. What is difference of SCTP and TCP?
3. What is difference between logical and link address?
4. What are types of packets used in ARP?
5. What is NetBIOS?

The Address Resolution Protocol (ARP) Analysis

Observation table:

- Clear your ARP cache from your terminal window using 'arp -d' command
- Visit **www.google.com** from your browser and note down the details of any **ARP** that you observed

Field Name	Field length (bits)	Field Value
Hardware type		
Protocol type		
Hardware size		
Protocol size		
Opcode		
Sender MAC address		
Sender IP address		
Target MAC address		
Target IP address		

15-ANALYSIS THE PROTOCOLS OF HTTP, DNS and DHCP Protoco

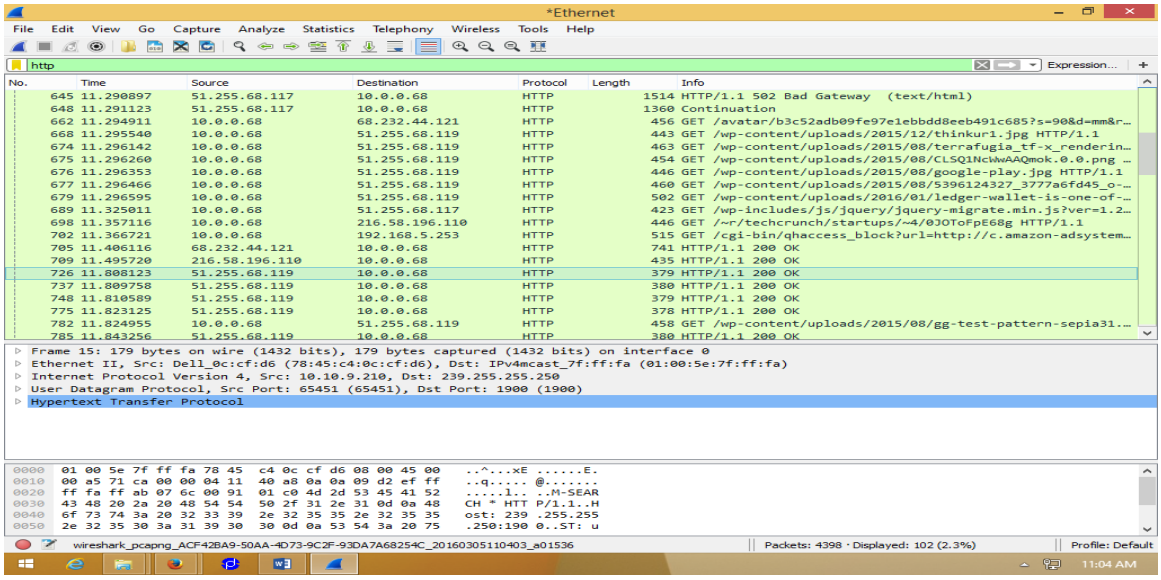
Aim: Analyze the protocols of HTTP, DNS and DHCP protocols through Wireshark Software

Software Required: Wireshark

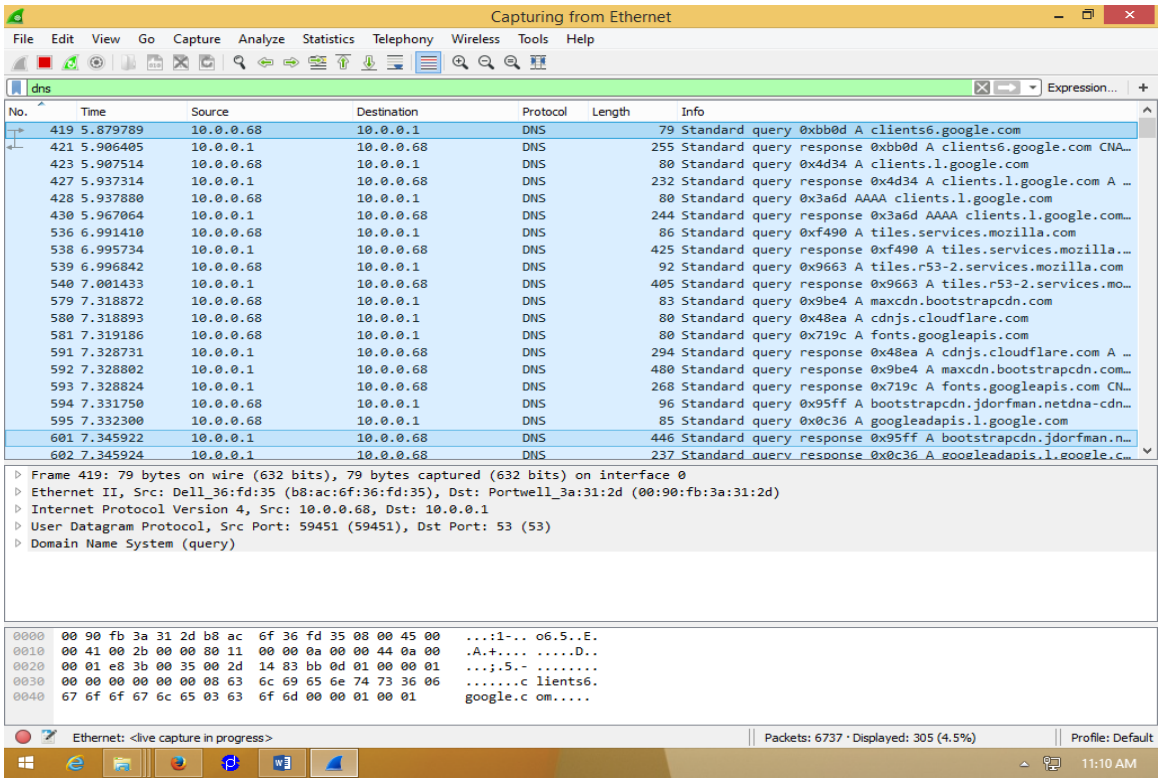
Procedure:

1. Open Wireshark-packet capture software.
2. Open capture tab.
3. Select interface.
4. Select an existing Ethernet card.
5. Click on start capture.
6. Open google and browse any streaming video files
7. Click on stop capture button.
8. Click on analyze button, display filters, select now.
9. Click on filter expression and enter “http, dns and dhcp”.
10. All the http,dns and dhcp packets will be listed.
11. Select the http, dns and dhcp packets and analyze the packet by right clicking.
12. Note down the details listed.

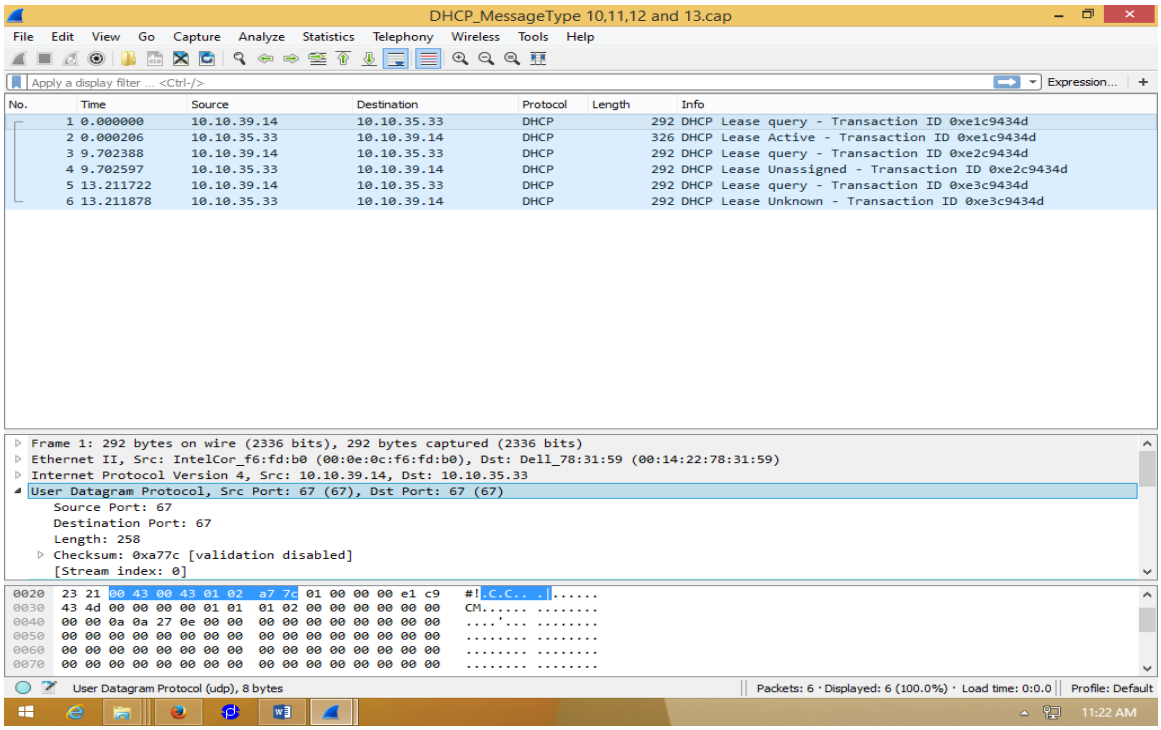
HTTP PROTOCOLS IN ETHERNET



DNS PACKETS IN ETHERNET



DHCP PACKETS IN ETHERNET
(Exchanging of Messages)



Result:

Reasoning Questions

1. What is the purpose if HTTP?
2. What are the port numbers of HTTP?
3. Define DNS.
4. What does DNS database contain?
5. What are port number of DHCP?
6. What are DHCP Packets?

The HTTP Analysis

Observation table:

- Visit **www.mgit.ac.in** site from your browser and note down the following details for a basic HTTP GET/response interaction

Question	Response
What version of HTTP is your browser running?	
What version of HTTP is the server running?	
What is the IP address of your computer ?	
What is the status code returned from the server to your browser ?	
When was the HTML file that you are retrieving last modified at the server ?	
How many bytes of content are being returned to your browser ?	