61,63/S13/S1/NL

ASSIGNMENT 4

Aim: Implementation of specific network topology with respect to UDP.
LO Mapping: Assignment matches LO3 and LO5.
Theory:
User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. So, there is no need to establish a connection before data transfer. The UDP helps to establish low-latency and loss-tolerating connections over the network. The UDP enables process-to-process communication.
What is User Datagram Protocol?
Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of the Internet services; provides assured delivery, reliability, and much more but all these services cost us additional overhead andlatency. Here, UDP comes into the picture. For real-time services like computergaming, voice or video communication, and live conferences; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also saves bandwidth.
● Used for simple request-response communication when the size of datais less andhence there is lesser concern about flow and error control.
● It is a suitable protocol for multicasting as UDP supports packet switching.
● UDP is used for some routing update protocols like RIP(Routing InformationProtocol).
● Normally used for real-time applications which can not tolerate unevendelays betweensections of a received message.

CODE:

```
# Create a simulator object
set ns [new Simulator]

# Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

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

# Set np for trace file
set np [open out.tr w]
```

```
$ns trace-all $np

# Define a 'finish' procedure
proc finish {} {
    global ns nf np
    $ns flush-trace
    # Close the NAM trace file
    close $nf
    # Execute NAM on the trace file
    exec nam out.nam &
    exit 0
}

# Create four nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]  ;# Central node for the star topology
set n3 [$ns node]

# Create links between the nodes (star topology: all nodes connect to n2)
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 2Mb 10ms DropTail

# Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10

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

# Monitor the queue for link (n2-n3) (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5

# Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

# Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp

# Setting packet size
$cbr set packet_size_ 1000
# Setting bit rate
$cbr set rate_ 1mb
# Setting random false means no noise
```

```
$cbr set random_ false

# Schedule events for the CBR traffic
$ns at 0.1 "$cbr start"
$ns at 4.5 "$cbr stop"

# Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

# Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"

# Run the simulation
$ns run
```

OUTPUT-