Aim: Write ns2 to test Simulated Application and Traffic generator application CBR over UDP.

Theory:
1. What is CBR?
● The meaning of CBR is Committed Bit Rate and other meanings are located at the bottom which take place within Networking terminology and CBR has 3 different meanings. All meanings which belong to CBR abbreviation take part only within Networking terminology and other meanings are not found. If you want to see other meanings, please click the CBR meaning link. Thus, you will be directed to a page which indicates all meanings of CBR.Unless there are 3 different meanings CBR abbreviation at the bottom, please search again by typing question structures such as "what does CBR mean in Networking, the meaning of CBR in Networking". Besides, you can search by typing CBR in the search box which is found on our website.

2. What is UDP?
● 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 prior to data transfer. Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of the Internet services, it provides assured delivery, reliability, and much more but all these services cost us additional overhead and latency. Here, UDP comes into the picture. For real-time services like computer gaming, voice or video communication, 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. User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

NS2 script/code:

Description:

This network consists of 4 nodes (n0, n1, n2, n3) as shown in above figure. The duplex links between n0 and n2, and n1 and n2 have 2 Mbps of bandwidth and 10 ms of delay. The duplex link between n2 and n3 has 1.7 Mbps of bandwidth and 20 ms of delay. Each node uses a DropTail queue, of which the maximum size is 10. A "tcp" agent is attached to n0, and a connection is established to a tcp "sink" agent attached to n3. As default, the maximum size of a packet that a "tcp" agent can generate is 1KByte. A tcp "sink" agent generates and sends ACK packets to the sender (tcp agent) and frees the received packets. A "udp" agent that is attached to n1 is connected to a "null" agent attached to n3. A "null" agent just frees the packets received. A "ftp" and a "cbr" traffic generator are attached to "tcp" and "udp" agents respectively, and the "cbr" is configured to generate 1 KByte packets at the rate of 1 Mbps. The "cbr" is set to start at 0.1 sec and stop at 4.5 sec, and "ftp" is set to start at 1.0 sec and stop at 4.0 sec.

```
#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


#Define a 'finish' procedure
```

```
proc finish {} {

        global ns nf

        $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]

set n3 [$ns node]


#Create links between the nodes

$ns duplex-link $n0 $n2 2Mb 10ms DropTail

$ns duplex-link $n1 $n2 2Mb 10ms DropTail

$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail


#Set Queue Size of link (n2-n3) to 10

$ns queue-limit $n2 $n3 10
```

```
#Give node position (for NAM)

$ns duplex-link-op $n0 $n2 orient right-down

$ns duplex-link-op $n1 $n2 orient right-up

$ns duplex-link-op $n2 $n3 orient right


#Monitor the queue for link (n2-n3). (for NAM)

$ns duplex-link-op $n2 $n3 queuePos 0.5



#Setup a TCP connection

set tcp [new Agent/TCP]

$tcp set class_ 2

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink]

$ns attach-agent $n3 $sink

$ns connect $tcp $sink

$tcp set fid_ 1


#Setup a FTP over TCP connection

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set type_ FTP
```

```
#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

$cbr set type_ CBR

$cbr set packet_size_ 1000

$cbr set rate_ 1mb

$cbr set random_ false



#Schedule events for the CBR and FTP agents

$ns at 0.1 "$cbr start"

$ns at 1.0 "$ftp start"

$ns at 4.0 "$ftp stop"
```

$ns at 4.5 "$cbr stop"

#Detach tcp and sink agents (not really necessary)

$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#Call the finish procedure after 5 seconds of simulation time

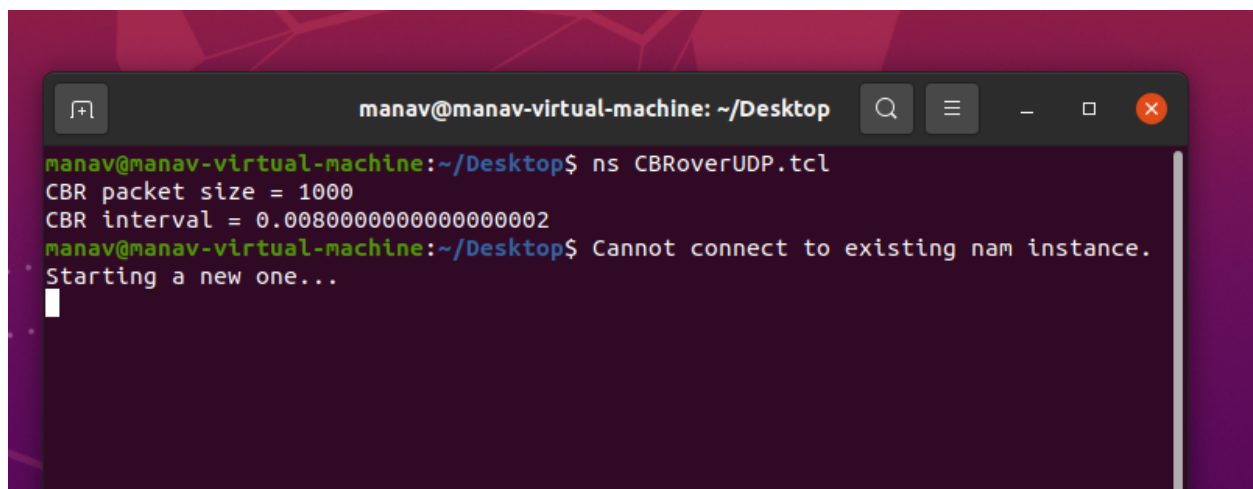$ns at 5.0 "finish"

#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:

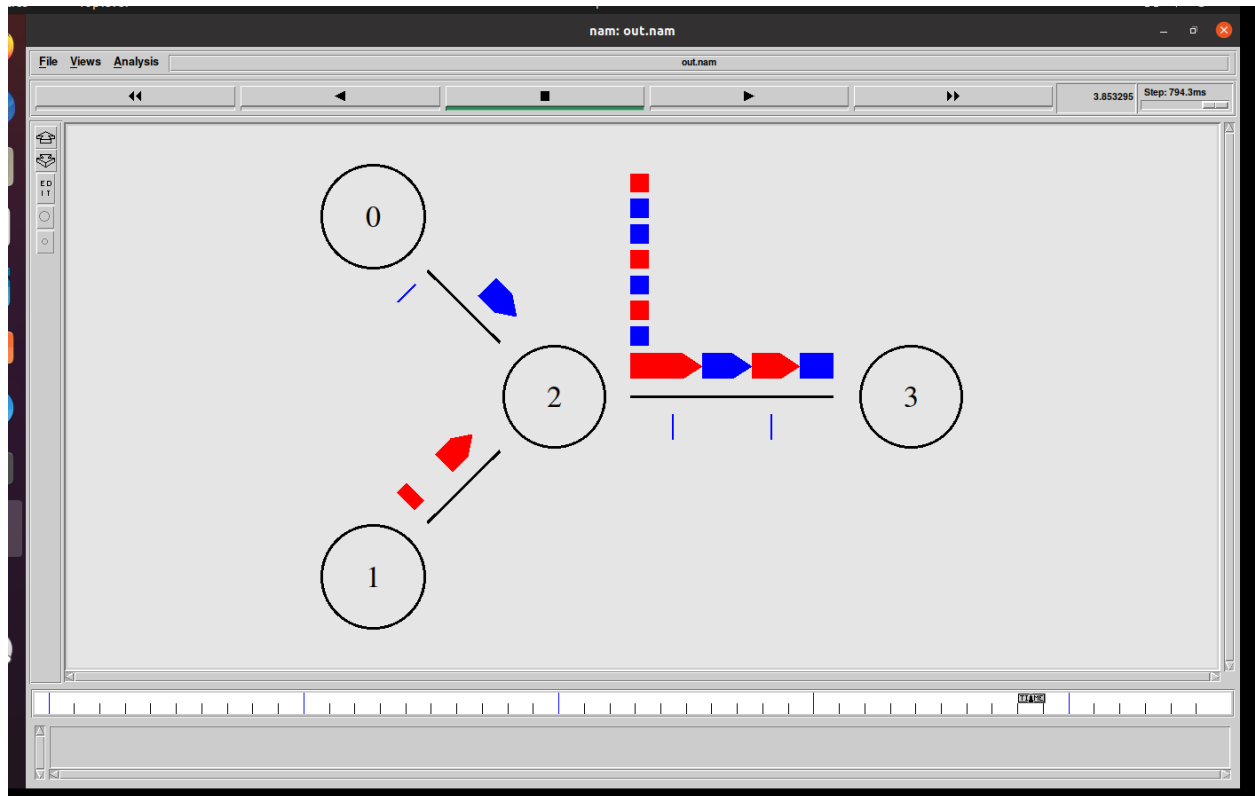
```
manav@manav-virtual-machine:~/Desktop$ ns CBRoverUDP.tcl
CBR packet size = 1000
CBR interval = 0.0080000000000000002
manav@manav-virtual-machine:~/Desktop$ Cannot connect to existing nam instance.
Starting a new one...
```

Conclusion:

We have performed simulation using CBR over UDP.