ANNA UNIVERSITY, GUINDY, CHENNAI:: 600 025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course Code: CS6111

Course Name: Computer Networks

Date :18.10.2024

NS2 SIMULATION

EXPERIMENT 1: SIMULATE HTTP, FTP AND DBMS ACCESS IN NETWORKS

ALGORITHM:

1. Create a simulator object
2. Open a nam trace file and define finish procedure then close the trace file, and execute nam on trace file.
3. Create two nodes that forms a network numbered from 0 to 1
4. Create duplex links between the nodes n(0) to n(1)
5. Setup TCP Connection between n(0) and n(1)
6. Apply FTP Traffic over TCP.
7. Schedule events and run the program

THEORY:

HTTP:

HTTP stands for Hyper Text Transfer Protocol.

It is a protocol used to access the data on the World Wide Web (www).

The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.

This protocol is known as HyperText Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.

HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files.

HTTP is used to carry the data in the form of MIME-like format.

FTP:

FTP stands for File transfer protocol.

FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.

It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.

It is also used for downloading the files to computer from other servers.

DBMS:

Data is the cornerstone of any modern software application, and databases are the most common way to store and manage data used by applications. With the explosion of web and cloud technologies, databases have evolved from traditional relational databases to more advanced types of databases such as NoSQL, columnar, key-value, hierarchical, and distributed databases. Each type has the ability to handle structured, semi-structured, and even unstructured data.

On top of that, databases are continuously handling mission-critical and sensitive data. When this is coupled with compliance requirements and the distributed nature of most data sets, managing databases has become highly complex. As a result, organizations require robust, secure, and userfriendly
tools to maintain these databases. This is where database management systems come into play—by offering a platform to manage databases. Let’s take a look.
A database management system (DBMS) is a software tool that enables users to manage a database easily. It allows users to access and interact with the underlying data in the database.

These actions can range from simply querying data to defining database schemas that fundamentally affect the database structure. Furthermore, DBMS allow users to interact with a database securely and concurrently without interfering with each user and while maintaining dataintegrity.

PROGRAM:

```
set val(stop) 10.5
#Create a ns simulator
set ns [new Simulator]
#Open the NS trace file
set tracefile [open httpex.tr w]
$ns trace-all $tracefile
#Open the NAM trace file
set namfile [open httpex.nam w]
$ns namtrace-all $namfile
#Create 6 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
#Createlinks between nodes
$ns duplex-link $n0 $n2 100.0Mb 10ms SFQ
$ns queue-limit $n0 $n2 50
$ns duplex-link $n3 $n2 100.0Mb 10ms SFQ
$ns queue-limit $n3 $n2 50
$ns duplex-link $n1 $n2 100.0Mb 10ms SFQ
$ns queue-limit $n1 $n2 50
$ns duplex-link $n3 $n4 100.0Mb 10ms SFQ
$ns queue-limit $n3 $n4 50
$ns duplex-link $n3 $n5 100.0Mb 10ms SFQ
$ns queue-limit $n3 $n5 50
#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n3 $n2 orient left
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n3 $n5 orient right-down
#Setup a TCP connection
set tcp0 [new Agent/TCP]
```

```
$ns attach-agent $n0 $tcp0
set sink3 [new Agent/TCPSink]
$ns attach-agent $n5 $sink3
$ns connect $tcp0 $sink3
$tcp0 set packetSize_ 1500
#Setup a TCP connection
set tcp1 [new Agent/TCP]
$ns attach-agent $n4 $tcp1
set sink2 [new Agent/TCPSink]
$ns attach$ns connect $tcp1 $sink2
$tcp1 set packetSize_ 1500
#Setup a UDP connection
set udp4 [new Agent/UDP]
$ns attach-agent $n2 $udp4
set null5 [new Agent/Null]
$ns attach-agent $n5 $null5
$ns connect $udp4 $null5
$udp4 set packetSize_ 48
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 10.0 "$ftp0 stop"
#Setup a FTP Application over TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 1.0 "$ftp1 start"
$ns at 10.0 "$ftp1 stop"
#Setup a CBR Application over UDP connection
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp4
$cbr2 set packetSize_ 48
$cbr2 set interval_ 50ms
$cbr2 set random_ null
$ns at 1.0 "$cbr2 start"
$ns at 10.0 "$cbr2 stop"
#Define a 'finish' procedure
proc finish {} {
global ns tracefile namfile
$ns flush-trace
close $tracefile
close $namfile
exec nam httpex.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
```

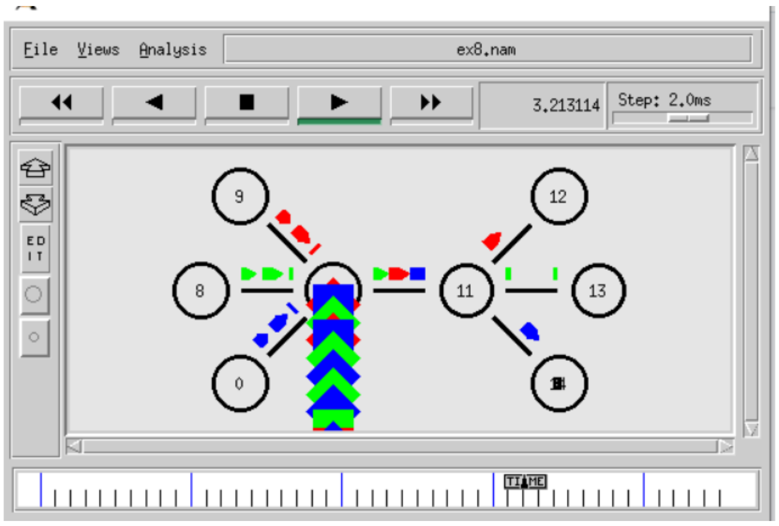
Experiment II : Simulation of Congestion Control Algorithm (RED) using Network Simulation tool (ns2)

PROGRAM:

```
#Create a simulator object
set ns [new Simulator]
set nr [open ex8_red.tr w]
$ns trace-all $nr
set nf [open ex8.nam w]
$ns namtrace-all $nf
proc finish { } {
global ns nr nf
$ns flush-trace
close $nf
close $nr
exec nam ex8.nam &
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
$ns duplex-link $n0 $n3 1Mb 10ms RED
$ns duplex-link $n1 $n3 1Mb 10ms RED
$ns duplex-link $n2 $n3 1Mb 10ms RED
$ns duplex-link $n3 $n4 1Mb 10ms RED
$ns duplex-link $n4 $n5 1Mb 10ms RED
$ns duplex-link $n4 $n6 1Mb 10ms RED
$ns duplex-link $n4 $n7 1Mb 10ms RED
$ns duplex-link-op $n0 $n3 orient right-up
$ns duplex-link-op $n3 $n4 orient middle
$ns duplex-link-op $n2 $n3 orient right-down
$ns duplex-link-op $n4 $n5 orient right-up
$ns duplex-link-op $n4 $n7 orient right-down
$ns duplex-link-op $n1 $n3 orient right
$ns duplex-link-op $n6 $n4 orient left
$ns duplex-link-op $n0 $n3 orient right-up
$ns duplex-link-op $n3 $n4 orient middle
```

```
$ns duplex-link-op $n2 $n3 orient right-down
$ns duplex-link-op $n4 $n5 orient right-up
$ns duplex$ns duplex-link-op $n1 $n3 orient right
$ns duplex-link-op $n6 $n4 orient left
set udp0 [new Agent/UDP]
$ns attach-agent $n2 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n5 $null0
$ns connect $udp0 $null0
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
set null0 [new Agent/Null]
$ns attach-agent $n6 $null0
$ns connect $udp1 $null0
set udp2 [new Agent/UDP]
$ns attach-agent $n0 $udp2
set cbr2 [new Application/Traffic/CBR]
$cbr2 set packetSize_ 500
$cbr2 set interval_ 0.005
$cbr2 attach-agent $udp2
set null0 [new Agent/Null]
$ns attach-agent $n7 $null0
$ns connect $udp2 $null0
$udp0 set fid_ 1
$udp1 set fid_ 2
$udp2 set fid_ 3
$ns color 1 Red
$ns color 2 Green
$ns color 3 blue
$ns at 0.1 "$cbr0 start"
$ns at 0.2 "$cbr1 start"
$ns at 0.5 "$cbr2 start"
$ns at 4.0 "$cbr2 stop"
$ns at 4.2 "$cbr1 stop"
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"
$ns run
$ns at 0.1 "$cbr0 start"
$ns at 0.2 "$cbr1 start"
$ns at 0.5 "$cbr2 start"
$ns at 4.0 "$cbr2 stop"
$ns at 4.2 "$cbr1 stop"
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"
```

$ns runOUTPUT:


File
Views
Analysis
ex8.nam
3.213114
Step: 2.0ms
EDIT
9
12
8
11
13
0
TIME