|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| final design | **Course:** | **Computer Networks** | **Course Code:** | **CL-307** |
| **Program:** | **BS (Computer Science)** | **Semester:** | **Fall 2018** |
| **Duration:** | **30 Minutes** | **Total Marks:** | **25** |
| **Paper Date:** | **19-Nov-2018** | **Weight** | **5 %** |
| **Section:** | **A** | **Page(s):** | **6** |
| **Exam:** | **Quiz 2** | **Reg. No.** |  |
| **Instruction/Notes:** |  | | | |

**PART 1 (Marks: 15)**

You will have to create following topology as given in the diagram below using statements in correct format from NS2 to implement the **Distance vector routing** protocol. Assume all the devices in the following topology as nodes and all the wires as **duplex links** with a DropTail queue mechanism. You **must orient the nodes** as shown in the topolgy below.

5Mb, 15ms

2Mb, 20ms

5Mb, 15ms

2Mb, 20ms

4

3

10Mb, 40ms

5Mb, 15ms

2Mb, 20ms

**Note: Implement the task in less number of statements to get the full credit**

* Send TCP Data from n0 to n7 🡪 (Data starts at 1.5 and stops at 4.3)  
  Send TCP Data from n2 to n5 🡪 (Data starts at 2.5 and stops at 7.3)  
  Send UDP Data from n1 to n6 🡪 (Data starts at 2.0 and stops at 6.5)
* UDP data with a rate of **33000 packets per 66 seconds** with a single packet having a size of **3KB.**
* Bring the link down between **n4** and **n6** at 1.3 and bring it back up at 2.5
* Limit the queue from n3 to n4 to only 50 packets
* Finish the simulation at 12.0

**PART 2 (Marks: 10)**

**Select an appropriate answer for each of the following questions.**

1. \_\_\_\_\_\_\_\_ tool that allows visualization of nodes.
2. Network animator
3. Trace Files
4. Logs
5. None
6. To take input in a variable **\_\_\_\_\_\_** is used.
7. puts stdin
8. set
9. gets stdin
10. execute
11. \_\_\_\_\_\_\_ procedure is call at the end of the simulation.
12. proc
13. finish
14. ns
15. none
16. \_\_\_\_\_\_\_ sets up simulation by assembling and configuring the objects as well as scheduling discrete events.
17. C++
18. C
19. OTcl
20. Nam
21. In the given statement what does 10ms and 10Mb represents.

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

1. Capacity, Propagation Delay
2. Propagation delay, Capacity
3. Queue size, Transmission Delay
4. Transmission delay, Capacity
5. Is it possible to send different packets to different nodes through one single node using Ns2?
6. Yes
7. No
8. \_\_\_\_\_\_ queueing mechanism divides equal traffic over a network or cause equal packet loss over a limited number of queues.
9. Exactly Fair Queue
10. Drop Tail
11. SFQ
12. Round Robin
13. In distance vector routing, a router advances traffic to the destination router using which mechanism:
14. Counting No. of Hubs
15. Counting No. of Routers
16. Counting No. of Switches
17. None
18. Data flow in a computer ring network topology without the Distance Vector protocol is
19. Uni-directional
20. Bi-directional
21. Simplex
22. Duplex
23. A router using distance-vector routing protocol gets the new network information only when the link is broken/damaged:
24. True
25. False

**NS2 Syntax:**

**Create Simulation:** set ns [new Simulator]

**Trace Files for NAM**: set nf [open out.nam w]

$ns namtrace-all $nf

**Finish Procedure:** proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

}

**Routing Algorithm**: $ns rtproto <protocol\_name>; <protocol\_name>: DV

**Node creation**: set <node\_name> [$ns node]

**Links Creation**: $ns <link\_type> <node1> <node2> <Bandwidth> <Delay> <queue\_type>

<link\_type>: simplex-link, duplex-link; <queue\_type>: DropTail, SFQ

**Graphical Settings (NAM)**: $ns <type> <node1> <node2> <option> <args>

<type> : simplex-link-op, duplex-link-op; <option> : orient, queuePos

**Limiting Queue**: $ns queue-limit <node\_name> <node\_name> <no. of packets>

**Transport Layer**: set <layer\_name> [new Agent/<agent\_type>]

<agent\_type>: UDP,TCP,Null,TCPSink

**Attaching Transport layer:** $ns attach-agent <node\_name> <layer\_name>

**Connecting Transport layer:** $ns connect <layer\_name> <layer\_name>

**File Transfer Protocol:** set <ftp\_name> [new Application/FTP]

**FTP Attach Agent:** <ftp\_name> attach-agent <layer\_name>

**Constant Bit Rate:** set <cbr\_name> [new Application/Traffic/CBR]

**CBR Attach Agent:** <cbr\_name> attach-agent <layer\_name>

**CBR Parameters:** <cbr\_name>set <parameter> <parameter\_value>

<parameter>: packetSize\_, interval\_, rate\_

**Event Scheduling:** $ns at <time\_frame\_value> “<cbr\_name>/<ftp\_name> <time\_event>”

<time\_event>: start, stop

**Ending Simulation:** $ns at <time\_frame\_value> “finish”

**Run Simulation:** $ns run

**Link Up/Down:** $ns rtmodel-at <time\_frame\_value> <function> <node1> <node2>

<function>: up,down