|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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:** | **20 Minutes** | **Total Marks:** | **15** |
| **Paper Date:** | **26-Nov-2018** | **Weight** | **4 %** |
| **Section:** | **C** | **Page(s):** | **2** |
| **Exam:** | **Quiz 2** | **Reg. No.** |  |
| **Instruction/Notes:** | Solve this Exam on paper | | | |

**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 n1 to n5 🡪 (Data starts at 1.5 and stops at 4.3)  
  Send UDP 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
* Finish the simulation at 12.0

**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

**Start From here**