**ECEN 602**

**NETWORK SIMULATOR 2 FAMILIRISATION**

**TEAM 17**

**Mohammad Faisal Khan**

**Amiya Ranjan Panda**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**README**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**TITLE : NS 2 Familirisation**

**INTRODUCTION :**

**This code is a part of the Network simulation Assignment for ECEN 602 at Texas A&M University.**

**It has been sucessfully compiled, executed and tested on Network Simulator - 2 environment.**

**Ns is a discrete event simulator targeted at networking research. Ns provides substantial support**

**for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite)**

**networks. NS2 consists of two key languages: CCC and Object-oriented Tool Command Language (OTcl).**

**While the CCC defines the internal mechanism (i.e., a backend) of the simulation, the OTcl sets up**

**simulation by assembling and configuring the objects as well as scheduling discrete events**

**(i.e., a frontend). The CCC and the OTcl are linked together using TclCL. After simulation, NS2**

**outputs either text-based simulation results. To interpret these results graphically and interactively,**

**tools such as NAM (Network AniMator) and XGraph are used. To analyze a particular behavior of the network,**

**users can extract a relevant subset of text-based data and transform it to a more conceivable presentation.**

**This step implements the design in the first step. It consists of two phases:**

**• Network configuration phase: In this phase, network components (e.g., node,**

**TCP and UDP) are created and configured according to the simulation design.**

**Also, the events such as data transfer are scheduled to start at a certain**

**time.**

**• Simulation Phase: This phase starts the simulation which was configured in the**

**Network Configuration Phase. It maintains the simulation clock and executes**

**events chronologically. This phase usually runs until the simulation clock reaches**

**a threshold value specified in the Network Configuration Phase.**

**TASK:**

**Use the NS-2 simulator to build the following configuration:**

**• Two routers (R1, R2) connected with a 1 Mbps link and 5ms of latency**

**• Two senders (src1, src2) connected to R1 with 10 Mbps links**

**• Two receivers (rcv1, rcv2) connected to R2 with 10 Mbps links**

**• Application sender is FTP over TCP**

**Run 400s simulations for the following variable parameters:**

**TCP version = (TCP SACK | TCP VEGAS)**

**Case 1:**

**•src1-R1 and R2-rcv1 end-2-end delay = 5 ms**

**•src2-R1 and R2-rcv2 end-2-end delay = 12.5 ms**

**Case 2:**

**•src1-R1 and R2-rcv1 end-2-end delay = 5 ms**

**•src2-R1 and R2-rcv2 end-2-end delay = 20 ms**

**Case 3:**

**•src1-R1 and R2-rcv1 end-2-end delay = 5 ms**

**•src2-R1 and R2-rcv2 end-2-end delay = 27.5 ms**

**USAGE:**

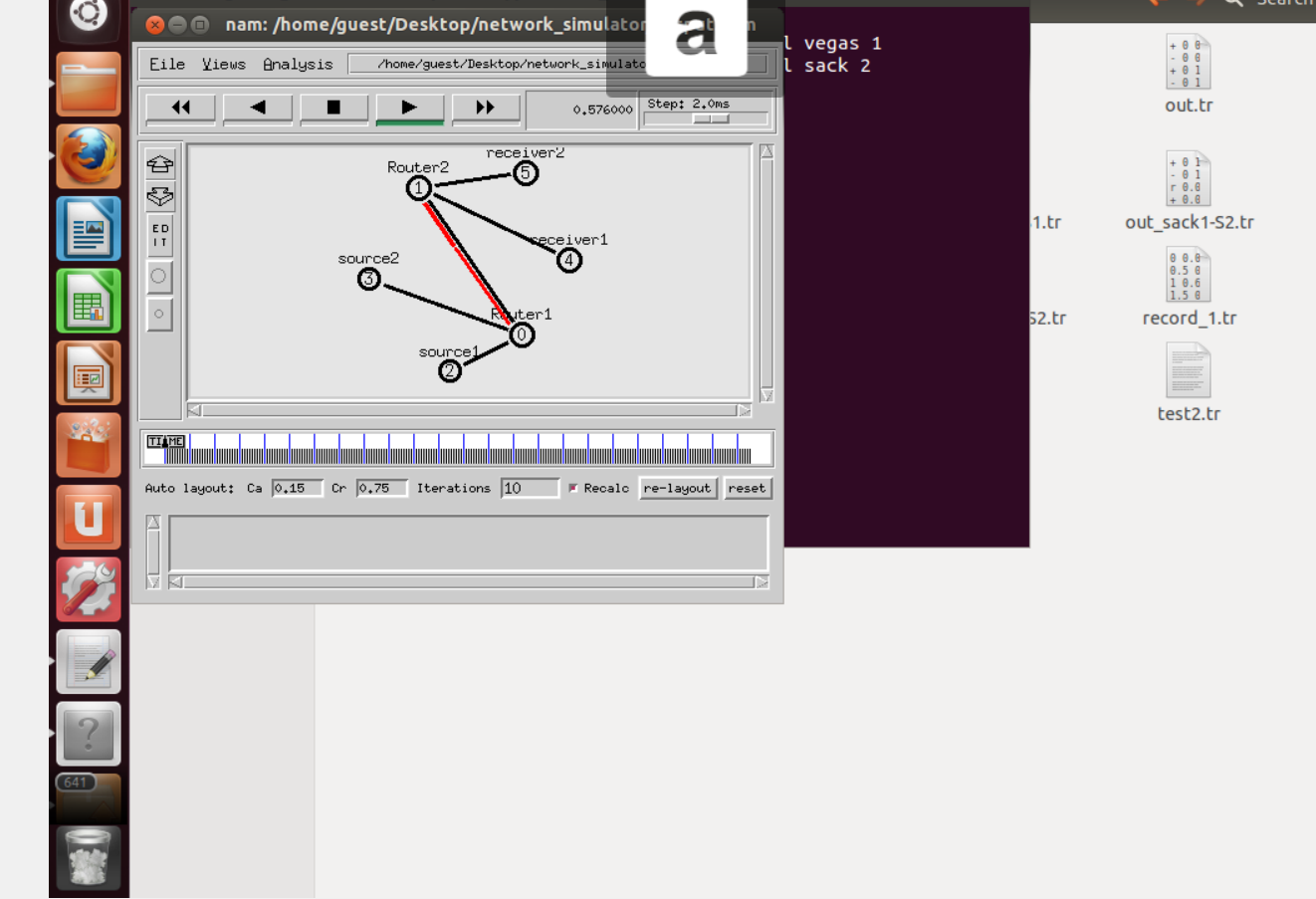
**1. open bash window.**

**2. Enter input in the below given format:**

**ns ns2.tcl <tcp type> <case no>**

**wherein, <tcp type> - "vegas" or "sack"**

**<case no> - "1", "2", "3"**

****