A PROJECT REPORT

***Submitted by***

BL.EN.U4AIE20041 Nandhitha Ravishankar

BL.EN.U4AIE20047 P Sadhana

***for the course***

**19AIE302 – ADVANCED COMPUTER NETWORKS**

***Guided and evaluated by***

***Mr. Vishwas HN***

***Assistant Professor,***

***Dept. of CSE***



AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

**BANGALORE 560 035**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CHAPTER** | **PAGE NO** |
| 1. | PROBLEM STATEMENT | 3 |
| 2. | INTRODUCTION | 4 |
| 3. | CODE | 6 |
| 4. | SCREENSHOTS | 12 |
| 5. | ANALYSIS | 14 |

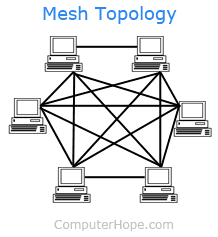
**PROBLEM STATEMENT**

Setup a wired network consisting of 10 nodes in mesh topology and apply RIP routing protocol with TCP. Analyze the percentage of data packets and acknowledgement packets transmitted.

**INTRODUCTION**

1. *Mesh Topology*

A mesh topology is a network setup where each computer and network device is interconnected with one another. This topology setup allows for most transmissions to be distributed even if one of the connections goes down. It is a topology commonly used for wireless networks. Below is a visual example of a simple computer setup on a network using a mesh topology.



1. *RIP Routing Protocol*

Routing Information Protocol (RIP) is a distance-vector routing protocol. Routers running the distance-vector protocol send all or a portion of their routing tables in routing-update messages to their neighbours.

1. *TCP*

Transmission Control Protocol (TCP) is a standard that defines how to establish and maintain a network conversation by which applications can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other.

1. *NS2*

NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks.

**Some of its features are:**

1. It is a discrete event simulator for networking research.

2. It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and DSR.

3. It simulates wired and wireless network.

4. It is primarily Unix based.

5. Uses TCL as its scripting language.

6. Otcl: Object oriented support

7. Tclcl: C++ and otcl linkage

8. Discrete event scheduler

**Basic Architecture:**

NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL

**CODE**

*wired.tcl*

*# Create a simulator object*

set ns [new Simulator]

$ns rtproto DV

*# Create a trace file, this file is for logging purpose*

set tracefile [open "wired.tr" w]

$ns trace-all $tracefile

*# create a animation information or NAM file creation*

set namfile [open "wired.nam" w]

$ns namtrace-all $namfile

*# create 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]

set n6 [$ns node]

set n7 [$ns node]

set n8 [$ns node]

set n9 [$ns node]

*# Creation of links between nodes with DropTail Queue*

*# DropTail means Dropping the tail*

$ns duplex-link $n0 $n1 2Mb 4ms DropTail

$ns duplex-link $n0 $n2 3Mb 2ms DropTail

$ns duplex-link $n0 $n3 4Mb 4ms DropTail

$ns duplex-link $n0 $n4 6Mb 8ms DropTail

$ns duplex-link $n0 $n5 7Mb 4ms DropTail

$ns duplex-link $n0 $n6 3Mb 7ms DropTail

$ns duplex-link $n0 $n7 8Mb 5ms DropTail

$ns duplex-link $n0 $n8 2Mb 1ms DropTail

$ns duplex-link $n0 $n9 8Mb 2ms DropTail

$ns duplex-link $n1 $n2 2Mb 8ms DropTail

$ns duplex-link $n1 $n3 6Mb 4ms DropTail

$ns duplex-link $n1 $n4 7Mb 4ms DropTail

$ns duplex-link $n1 $n5 3Mb 2ms DropTail

$ns duplex-link $n1 $n6 7Mb 1ms DropTail

$ns duplex-link $n1 $n7 2Mb 1ms DropTail

$ns duplex-link $n1 $n8 9Mb 7ms DropTail

$ns duplex-link $n1 $n9 3Mb 4ms DropTail

$ns duplex-link $n2 $n3 2Mb 1ms DropTail

$ns duplex-link $n2 $n4 4Mb 2ms DropTail

$ns duplex-link $n2 $n5 6Mb 7ms DropTail

$ns duplex-link $n2 $n6 8Mb 2ms DropTail

$ns duplex-link $n2 $n7 9Mb 6ms DropTail

$ns duplex-link $n2 $n8 7Mb 1ms DropTail

$ns duplex-link $n2 $n9 2Mb 3ms DropTail

$ns duplex-link $n3 $n4 2Mb 9ms DropTail

$ns duplex-link $n3 $n5 1Mb 4ms DropTail

$ns duplex-link $n3 $n6 8Mb 3ms DropTail

$ns duplex-link $n3 $n7 6Mb 1ms DropTail

$ns duplex-link $n3 $n8 4Mb 7ms DropTail

$ns duplex-link $n3 $n9 3Mb 5ms DropTail

$ns duplex-link $n4 $n5 3Mb 4ms DropTail

$ns duplex-link $n4 $n6 2Mb 6ms DropTail

$ns duplex-link $n4 $n7 7Mb 7ms DropTail

$ns duplex-link $n4 $n8 5Mb 8ms DropTail

$ns duplex-link $n4 $n9 2Mb 2ms DropTail

$ns duplex-link $n5 $n6 2Mb 8ms DropTail

$ns duplex-link $n5 $n7 3Mb 1ms DropTail

$ns duplex-link $n5 $n8 4Mb 2ms DropTail

$ns duplex-link $n5 $n9 5Mb 5ms DropTail

$ns duplex-link $n6 $n7 4Mb 3ms DropTail

$ns duplex-link $n6 $n8 6Mb 5ms DropTail

$ns duplex-link $n6 $n9 8Mb 7ms DropTail

$ns duplex-link $n7 $n8 6b 4ms DropTail

$ns duplex-link $n7 $n9 2Mb 4ms DropTail

$ns duplex-link $n8 $n9 5Mb 8ms DropTail

*# Creation of TCP Agent*

set tcp [new Agent/TCP]

set sink [new Agent/TCPSink]

$ns attach-agent $n2 $tcp

$ns attach-agent $n5 $sink

$ns connect $tcp $sink

*# Creation of application, FTP*

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set type\_ FTP

$ftp set packet\_size\_ 1000

$ftp set rate\_ 1mb

*# Start the traffic*

$ns at 1.0 "$ftp start"

$ns rtmodel-at 2.0 down $n2 $n5

$ns rtmodel-at 3.0 up $n2 $n5

$ns at 4.0 "$ftp stop"

$ns at 5.0 "finish"

*# The following procedure will be called at 10.0 seconds*

proc finish {} {

    global ns tracefile namfile

    $ns flush-trace

    close $tracefile

    close $namfile

    exec awk -f packet.awk wired.tr &

    exit 0

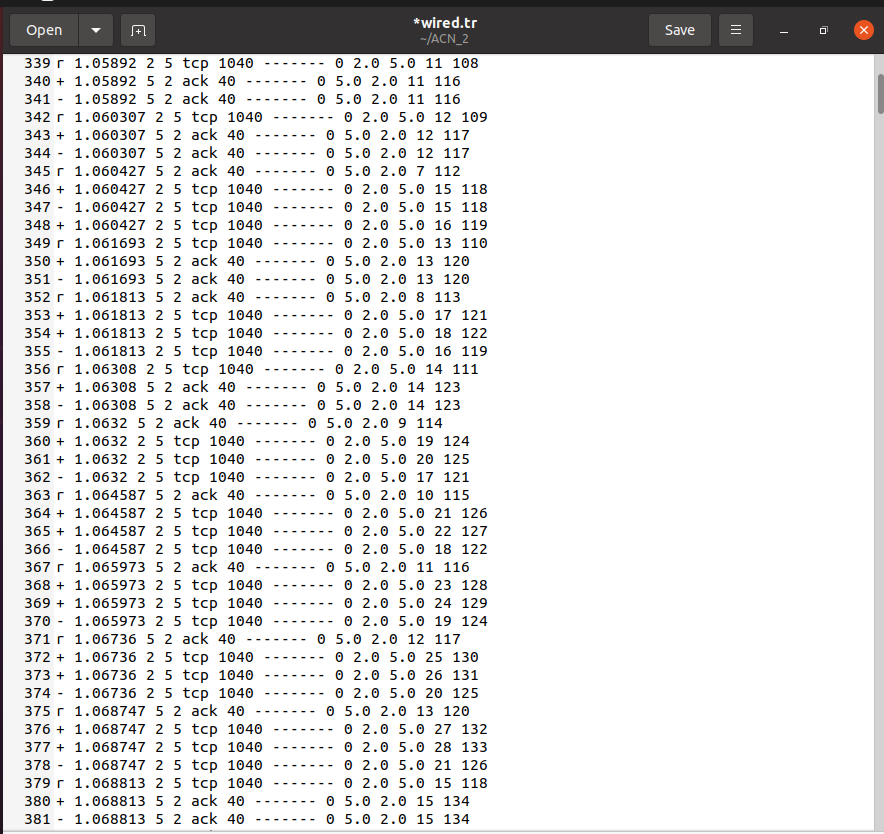
}

puts "Simulation is starting…"

$ns run

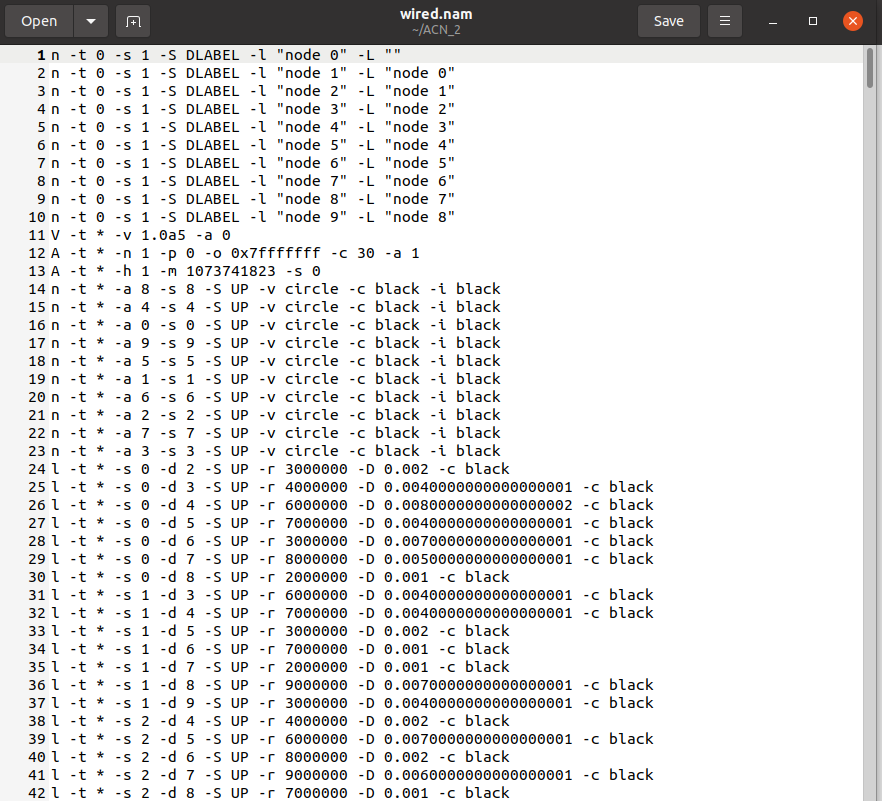
*wired.tr*

The trace file is a file generated by NS2 that contains details regarding the network events that occurs during the simulation. The trace file can be used for various purposes such as analysing the performance of the networks and so on. The trace file obtained during the simulation for this project has been shown below.

**

*wired.nam*

Nam is a Tcl/Tk based animation tool that is used to visualize the ns simulations and real-world packet trace data. The first step to use nam is to produce a nam trace file. The nam trace file should contain topology information like nodes, links, queues, node connectivity etc as well as packet trace information. The nam file obtained during the simulation for this project has been shown below.

**

*packet.awk*

BEGIN

{

    receive=0

    drop=0

    total=0

    ratio=0

}

{

    if($1=="r" && $4==5)

    {

        receive++

    }

    if($1=="d" && $4==5)

    {

    drop++

    }

}

END

{

    total = receive + drop

    ratio = (receive/total)\*100

    printf("\n total packets sent: %d", total)

    printf("\n total acknowledgment packets sent: %f", receive)

}

**SCREENSHOTS**

A picture containing chart

Description automatically generated

Here, the mesh topology with 10 nodes has been depicted.

Diagram

Description automatically generated

The data transmission from the source and the destination node, i.e., from node 2 to node 5 can be seen. The darker line depicts the data transmission and the dotted line shows the acknowledgement data that is transmitted.

Chart, radar chart

Description automatically generated

Here, a link failure is brought between the source and the destination node. Therefore, the packets need to find new routes to reach the destination.

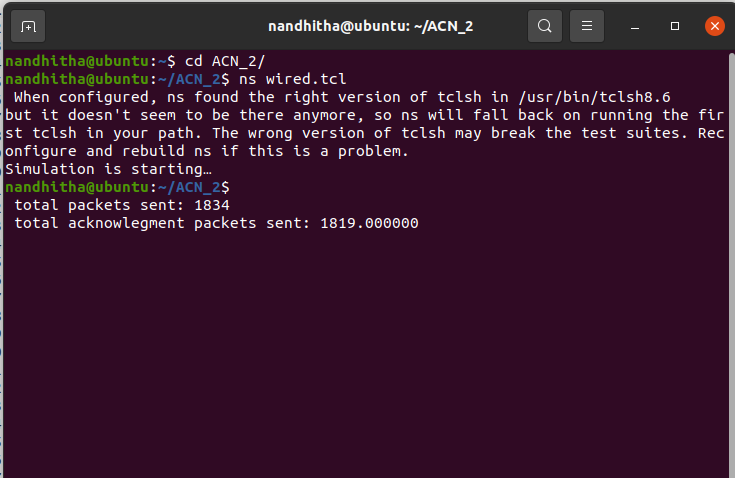
Chart, radar chart

Description automatically generated

Now using RIP protocol which is a distance vector protocol, the short path is found and the data is transmitted as seen in the above picture.

**ANALYSIS**

Analysis of data packets and acknowledgement packets transmitted have been done and the results below have been obtained with the help of packet.awk file.

****

We can observe from the output that the number of packets sent is more than the number of acknowledgments received which means that the designed network is not very efficient in terms of utilising its resources.

This could be due to various reasons such as high error rate of the network, package getting lost and corrupted, low bandwidth of the network. Network congestion.