18IT470-COMPUTER NETWORKS

TERMINAL LAB EXAM

NAME:KEERTHANA R

REG NO:19IT048

DATE:12/07/2021

1.Develop aa concurrent server for finding the determinant of a matrix

Aim:

To implement concurrent server for finding the determinant of a matrix application in Java.

Algorithm:

MatrixClient.java

1. Start
2. Establish connection with server
3. Get start time
4. get input from user
5. create an object for buffer reader
6. Send the data to server
7. Receive the data from sever
8. Get end time
9. Print the data
10. Calculate running time(end time-start time)
11. End

MatrixServer.java

```
1. Start
2. create an object for buffer reader
3. get input from client
4. for(int i = 0; i < 3; i++) {
    for(int j = 0; j < 3; j++) {
        arr[i][j] = sc.nextInt();
        }
        }
5. int z=stub.Determinant(arr);
6. Send the cnt to client
7. End
```

Coding:

MatrixClient.java

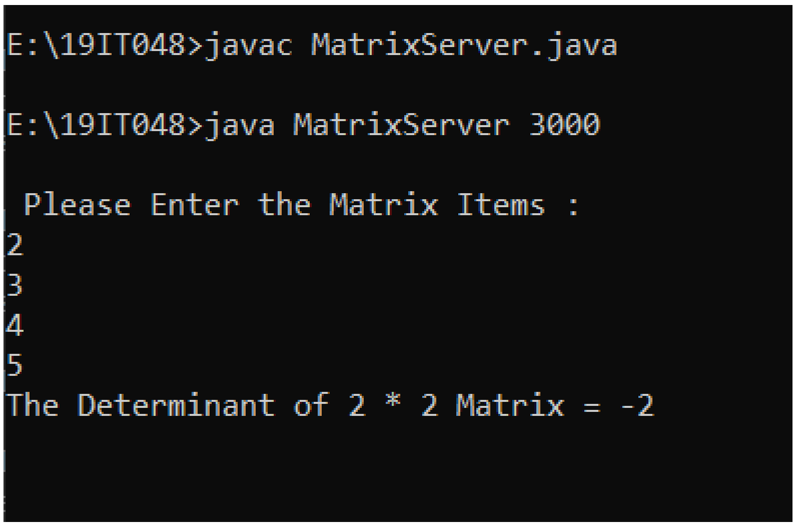
```
import java.io.*;
import java.net.*;
class MatrixClient{
public static void main(String[ ] args){
try{
//get host name
InetAddress serverHost = InetAddress.getByName(args[0]);
int serverPort = Integer.parseInt(args[1]);
//start time
long startTime = System.currentTimeMillis( );
//get number from user
String count = args[2];
Socket clientSocket = new Socket(serverHost, serverPort);
//send the data to server
PrintStream ps = new PrintStream(clientSocket.getOutputStream());
ps.println(count);
//create an object for buffer reader
BufferedReader br = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
//receive the data from the server
int cnt = Integer.parseInt(br.readLine());
//print the output
System.out.println("Determinant of matrix = " + count);
//get end time
long endTime = System.currentTimeMillis();
//print running time
System.out.println("Time consumed for receiving the feedback from the server:" +(endTime-
startTime)+ " milliseconds");
clientSocket.close( );
}
catch(Exception e){e.printStackTrace( );}
}
}
```

MatrixServer.java

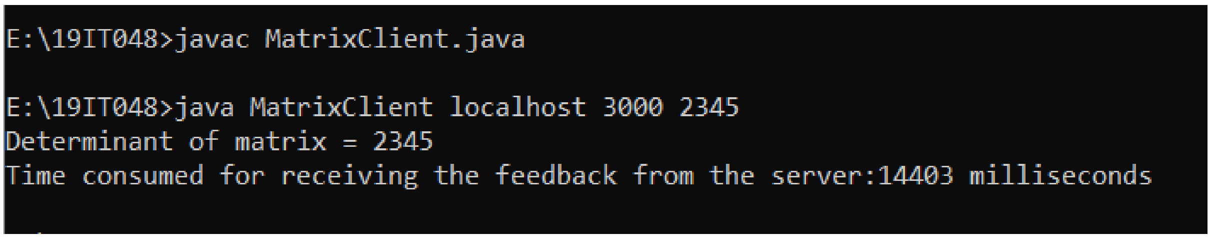
```
import java.net.*;
import java.util.*;
import java.io.*;
class MatrixThread extends Thread{
Socket clientSocket;
MatrixThread(Socket cs){ clientSocket = cs; }
public void run( ){
try{
//create an object for buffer reader
BufferedReader br = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream( )));
//get input from client
int count = Integer.parseInt(br.readLine( ));
{
        int[][] arr = new int[2][2];

                int i, j, determinant = 0;

                Scanner sc = new Scanner(System.in);

                System.out.println("\n Please Enter the Matrix Items :  ");
                for(i = 0; i < 2; i++) {
                        for(j = 0; j < 2; j++) {
                                arr[i][j] = sc.nextInt();
                        }
```

```
                    }

                    determinant = (arr[0][0] * arr[1][1]) - (arr[0][1] * arr[1][0]);

                    System.out.println("The Determinant of 2 * 2 Matrix = " + determinant );
Thread.sleep(200);
}
//send input to client
PrintStream ps = new PrintStream(clientSocket.getOutputStream( ));
ps.println(count);
ps.flush( );
clientSocket.close( );
}
catch(Exception e){e.printStackTrace( );}
}
}
class MatrixServer{
public static void main(String[ ] args){
try{
int serverPort = Integer.parseInt(args[0]);
ServerSocket calcServer = new ServerSocket(serverPort);
while (true){
Socket clientSocket = calcServer.accept( );
MatrixThread thread = new MatrixThread(clientSocket);
thread.start( );
}
}
catch(Exception e){e.printStackTrace( );}
}
}
```

Output:



```
E:\19IT048>javac MatrixServer.java

E:\19IT048>java MatrixServer 3000

 Please Enter the Matrix Items :
2
3
4
5
The Determinant of 2 * 2 Matrix = -2
```



```
E:\19IT048>javac MatrixClient.java

E:\19IT048>java MatrixClient localhost 3000 2345
Determinant of matrix = 2345
Time consumed for receiving the feedback from the server:14403 milliseconds
```

Result:

Concurrent server for finding the determinant of a matrix application is implementedin java.

2. Simulate LAN based on Ethernet using Opnet

Aim:

To simulate a LAN based on Ethernet with a minimum of ten nodes and examine the performance under different load scenarios using OPNET

Procedure:

Create a New Project

To create a new project for the Ethernet network:

1. Start Riverbed Modeler Academic Edition ⇒ Choose New from the File menu.
2. Select Project⇒ Click OK⇒ Nhe project _Ethernet, and the scenario Coax_2⇒ Make sure that the Use Startup Wizard is checked ⇒ Click OK. Local area networks (LANs)are designed to span distances of up to a few thousand meters
3. In the Startup Wizard: Initial Topology dialog box, make sure that Create Empty

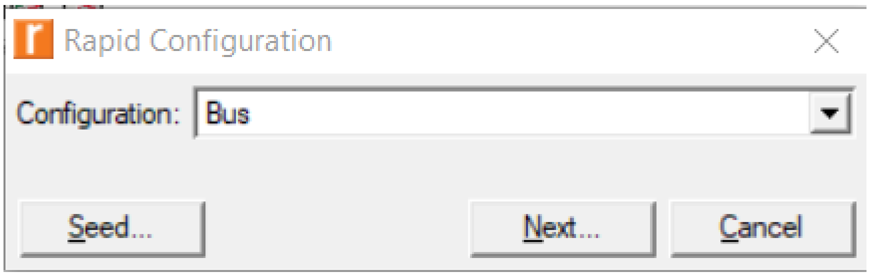
Scenarioisselected ⇒ Click Next ⇒ Choose Office from the Network Scale list ⇒ Click Next ⇒ Assign 200 to X Span and keep Y Span as 100 ⇒ Click Next twice ⇒ Click Finish.

4. Close the Object Tree dialog box.

Create the Network

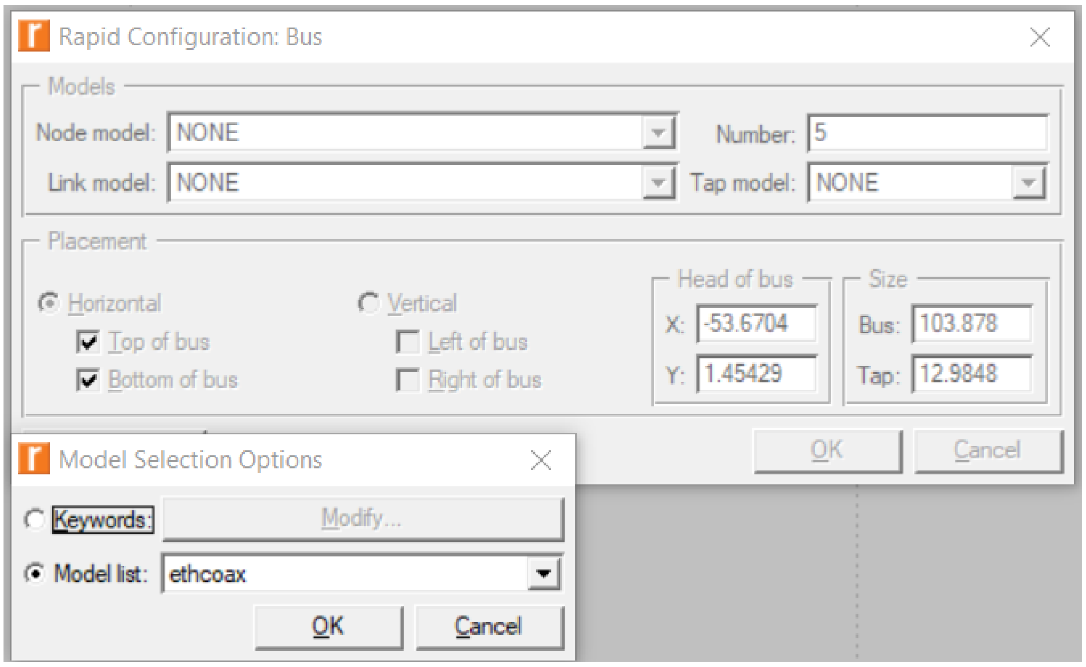
To create our coaxial Ethernet network:

1. To create the network configuration, select Topology ⇒ Rapid Configuration. From thedrop-down menu choose Bus and click Next.



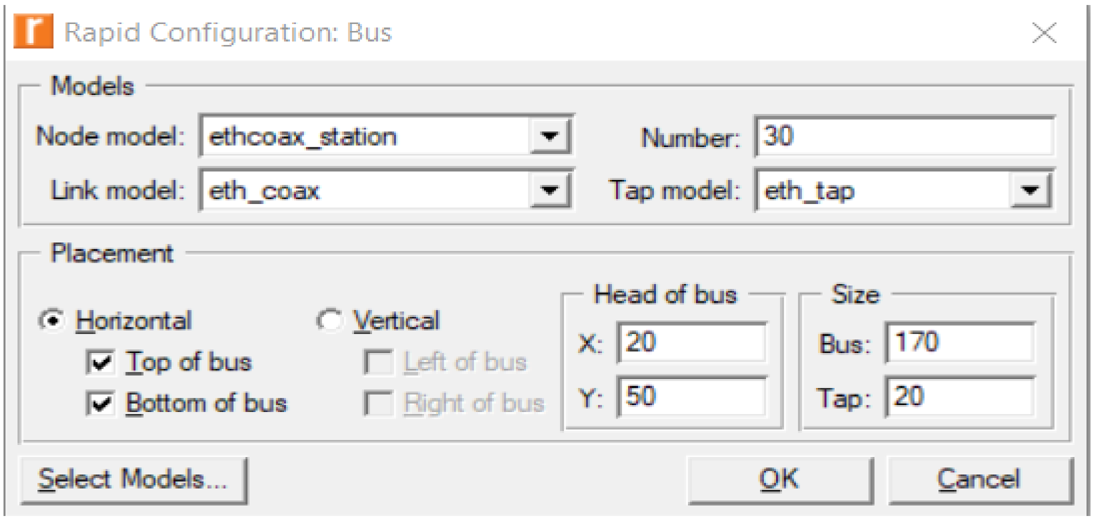
Configuration: Bus

2. Click the Select Models button in the Rapid Configuration dialog box. From the ModelList drop-down menu choose ethcoax and click OK.



Node model: NONE
Number: 5
Link model: NONE
Tap model: NONE
Horizontal [✓]
Top of bus [✓]
Bottom of bus [✓]
Vertical
Left of bus
Right of bus
Head of bus X: -53.6704
Head of bus Y: 1.45429
Size Bus: 103.878
Size Tap: 12.9848
Keywords:
Model list: [✓] ethcoax

3. In the Rapid Configuration dialog box, set the following eight values and click OK



Rapid Configuration: Bus

Models

Node model: ethcoax_station | Number: 30

Link model: eth_coax | Tap model: eth_tap

Placement

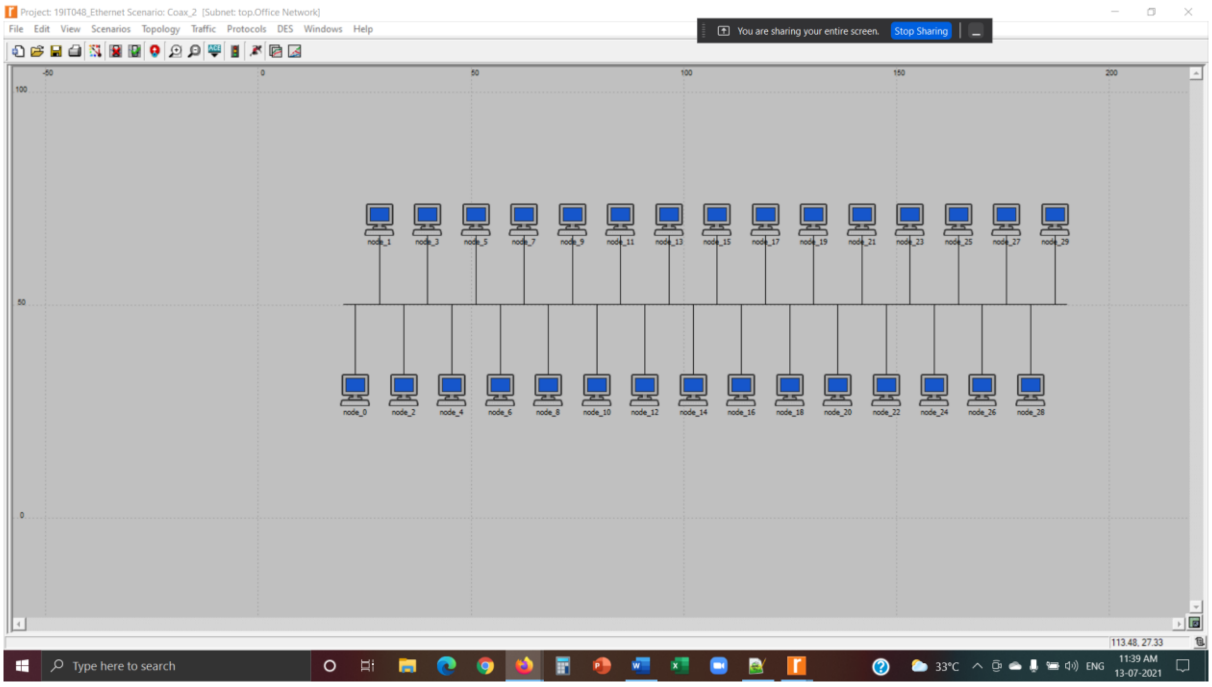
[✓ Horizontal] Vertical

[✓ Top of bus] Left of bus

[✓ Bottom of bus] Right of bus

Head of bus — X: 20, Y: 50

Size — Bus: 170, Tap: 20



Project: 19IT048_Ethernet Scenario: Coax_2 [Subnet: top.Office Network]

4. To configure the coaxial bus, right-click on the horizontal link ⇒ Select Edit Attributes(Advanced) from the menu:

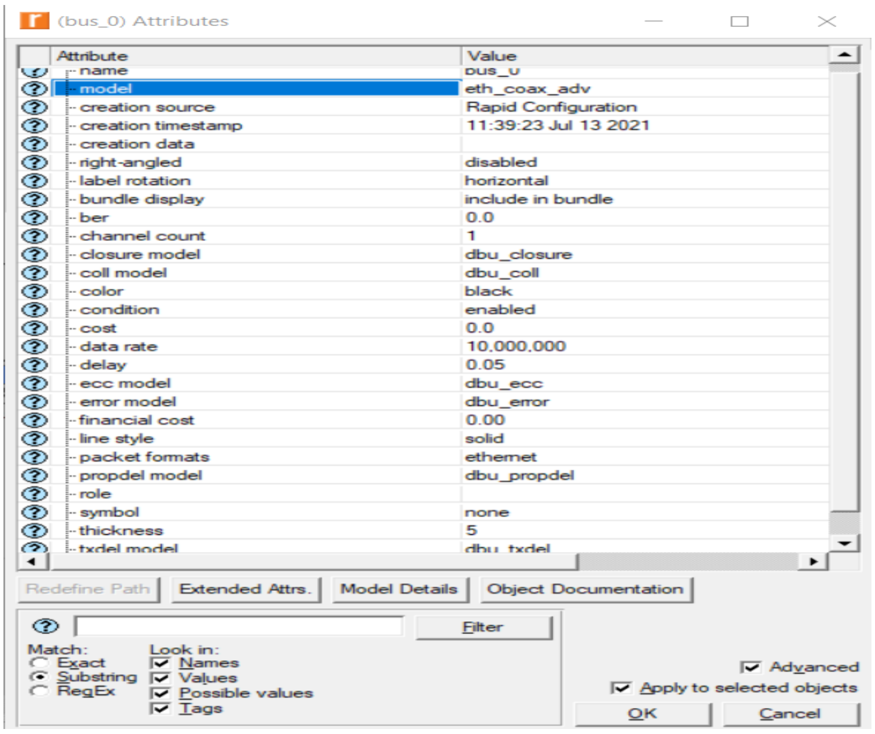
   a. Click on the value of the model attribute ⇒ Select Edit from the dropdown menu ⇒

Choose the eth_coax_adv model.

   b. Assign the value 0.05 to the delay attribute (propagation delay in sec/m).

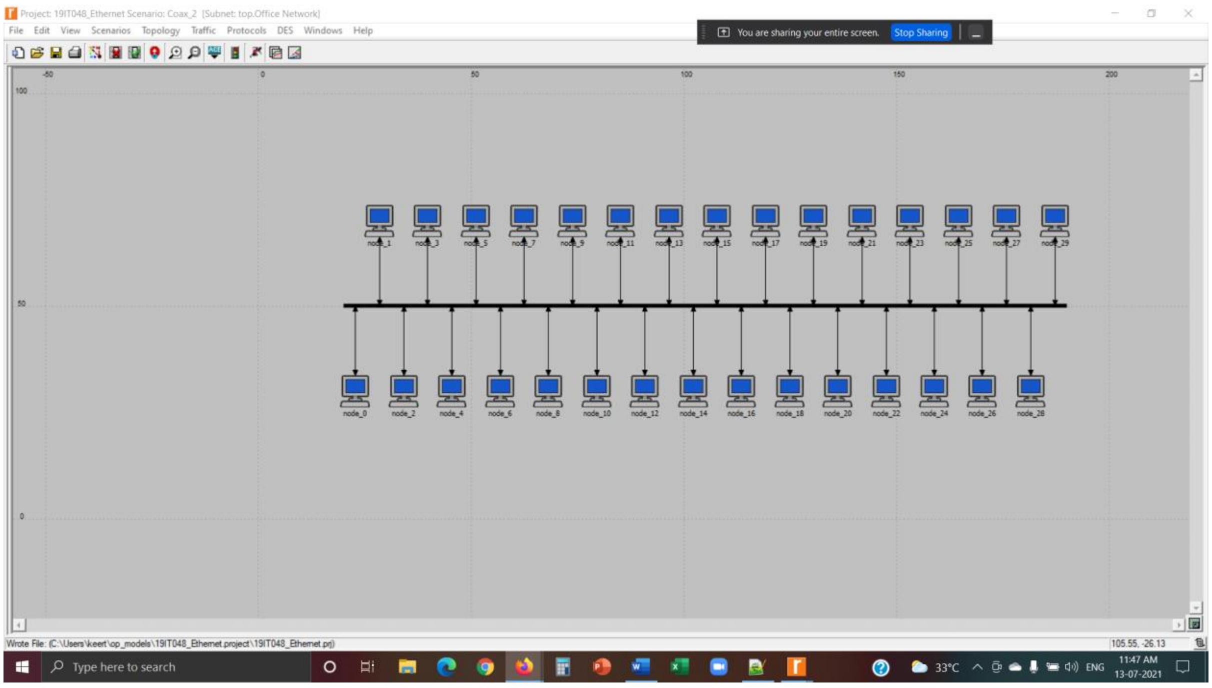
   c. Assign 5 to the thickness attribute.

   d. Click OK.


(bus_0) Attributes

Attribute	Value
name	bus_0
model	eth_coax_adv
creation source	Rapid Configuration
creation timestamp	11:39:23 Jul 13 2021
creation data	
right-angled	disabled
label rotation	horizontal
bundle display	include in bundle
ber	0.0
channel count	1
closure model	dbu_closure
coll model	dbu_coll
color	black
condition	enabled
cost	0.0
data rate	10,000,000
delay	0.05
ecc model	dbu_ecc
error model	dbu_error
financial cost	0.00
line style	solid
packet formats	ethernet
propdel model	dbu_propdel
role	
symbol	none
thickness	5
txdel model	dbu_txdel

5. Now you have created the network. It should look like the illustration below
6. Make sure to save your project.


Project: 19IT048_Ethernet Scenario: Coax_2 [Subnet: top.Office Network]
File Edit View Scenarios Topology Traffic Protocols DES Windows Help
You are sharing your entire screen. Stop Sharing
node_0 node_2 node_4 node_6 node_8 node_10 node_12 node_14 node_16 node_18 node_20 node_22 node_24 node_26 node_28
node_1 node_3 node_5 node_7 node_9 node_11 node_13 node_15 node_17 node_19 node_21 node_23 node_25 node_27 node_29
Wrote File: (C:\Users\keert\op_models\19IT048_Ethernet project\19IT048_Ethernet.prj)
Type here to search
33°C ENG 11:47 AM 13-07-2021

Configure the Network Nodes

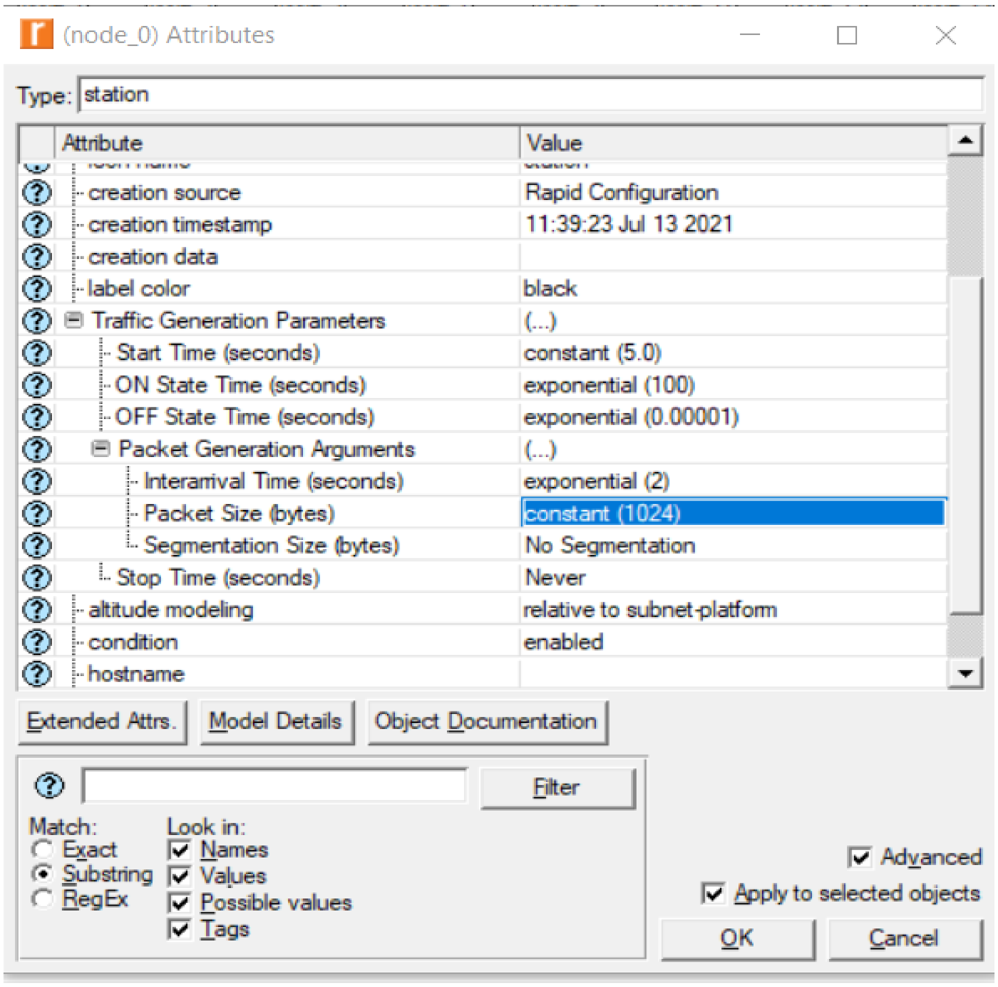
To configure the traffic generated by the nodes:

1. Right-click on any of the 30 nodes ⇒ Select Similar Nodes. Now all nodes in the networkare selected.

2. Right-click on any of the 30 nodes ⇒ Edit Attributes.
3. Check the Apply Changes to Selected Objects check box. This is important to avoidreconfiguring each node individually.
4. Expand the Traffic Generation Parameters hierarchy:

   Change the value of the ON State Time to exponential(100) ⇒ Change the value ofthe OFF State Time to exponential(0.00001). (Note: Packets are generated only in the "ON" state.)
5. Expand the Packet Generation Arguments hierarchy:

   Change the value of the Packet Size attribute to constant(1024) ⇒ Change the valueof the Interarrival Time attribute to exponential(2).


(node_0) Attributes

Type: station

Attribute	Value
creation source	Rapid Configuration
creation timestamp	11:39:23 Jul 13 2021
creation data	
label color	black
Traffic Generation Parameters	(...)
Start Time (seconds)	constant (5.0)
ON State Time (seconds)	exponential (100)
OFF State Time (seconds)	exponential (0.00001)
Packet Generation Arguments	(...)
Interarrival Time (seconds)	exponential (2)
Packet Size (bytes)	constant (1024)
Segmentation Size (bytes)	No Segmentation
Stop Time (seconds)	Never
altitude modeling	relative to subnet-platform
condition	enabled
hostname	

6. Click OK to return back to the Project Editor.
7. Make sure to save your project

Choose the Statistics

To choose the statistics to be collected during the simulation:

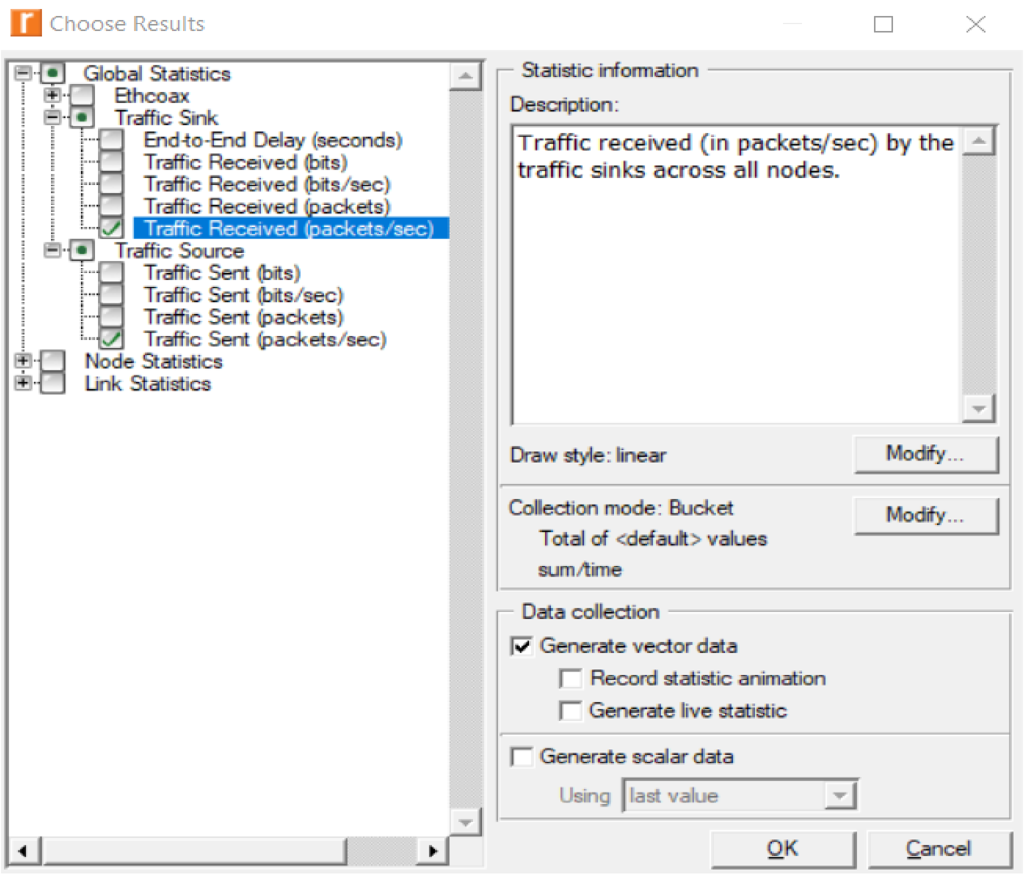
1. Right-click anywhere in the project workspace (but not on one of the nodes or

links) andselect Choose Individual DES Statistics from the pop-up menu ⇒Expand the Global Statistics hierarchy.

a. Expand the Traffic Sink hierarchy ⇒ Click the check box next to Traffic Received(packets/sec) (make sure you select the statistic with units of packets/sec),

b. Expand the Traffic Source hierarchy ⇒ Click the check box next toTraffic Sent(packets/sec).

c. Click OK.


Choose Results

- Global Statistics
  - Ethcoax
  - Traffic Sink
    - End-to-End Delay (seconds)
    - Traffic Received (bits)
    - Traffic Received (bits/sec)
    - Traffic Received (packets)
    - ✓ Traffic Received (packets/sec)
  - Traffic Source
    - Traffic Sent (bits)
    - Traffic Sent (bits/sec)
    - Traffic Sent (packets)
    - ✓ Traffic Sent (packets/sec)
- Node Statistics
- Link Statistics

Statistic information

Description:

Traffic received (in packets/sec) by the traffic sinks across all nodes.

Draw style: linear

Collection mode: Bucket
Total of <default> values
sum/time

Data collection

[✓] Generate vector data

[ ] Record statistic animation

[ ] Generate live statistic

[ ] Generate scalar data

Using: last value

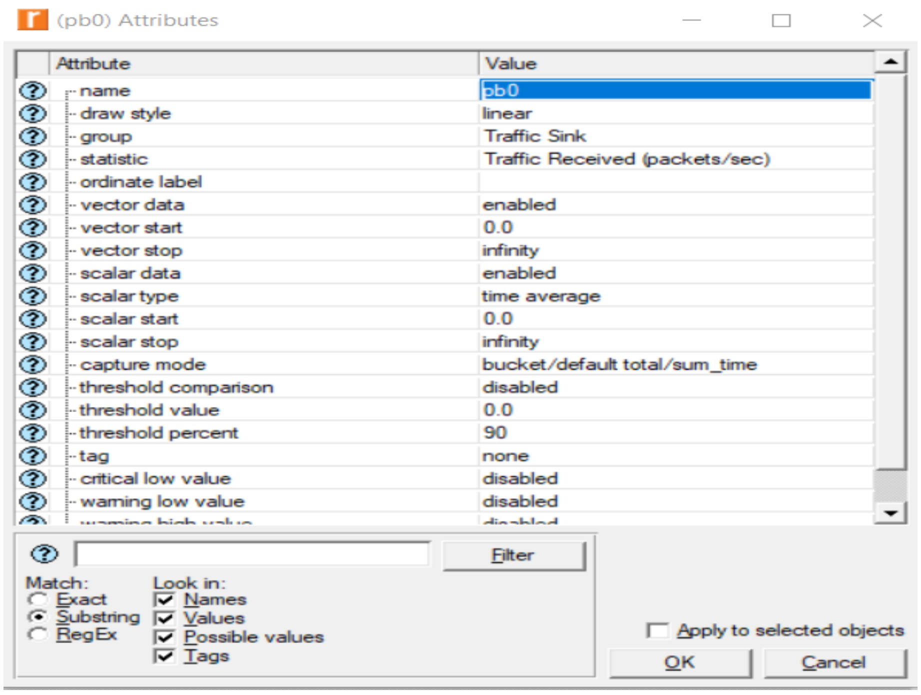
2. Now to collect the average of the above statistics as a scalar value by the end of eachsimulation run:

a. Select Choose Statistics (Advanced) from the DES menu

b. The Traffic Sent and Traffic Received probes should appear under the GlobalStatistic Probes.

c. Right-click on Traffic Received probe ⇒ Edit Attributes. Set the scalar data attribute to enabled ⇒ Set the scalar type attribute to time average ⇒

Compare to thefollowing figure and click OK



(pb0) Attributes

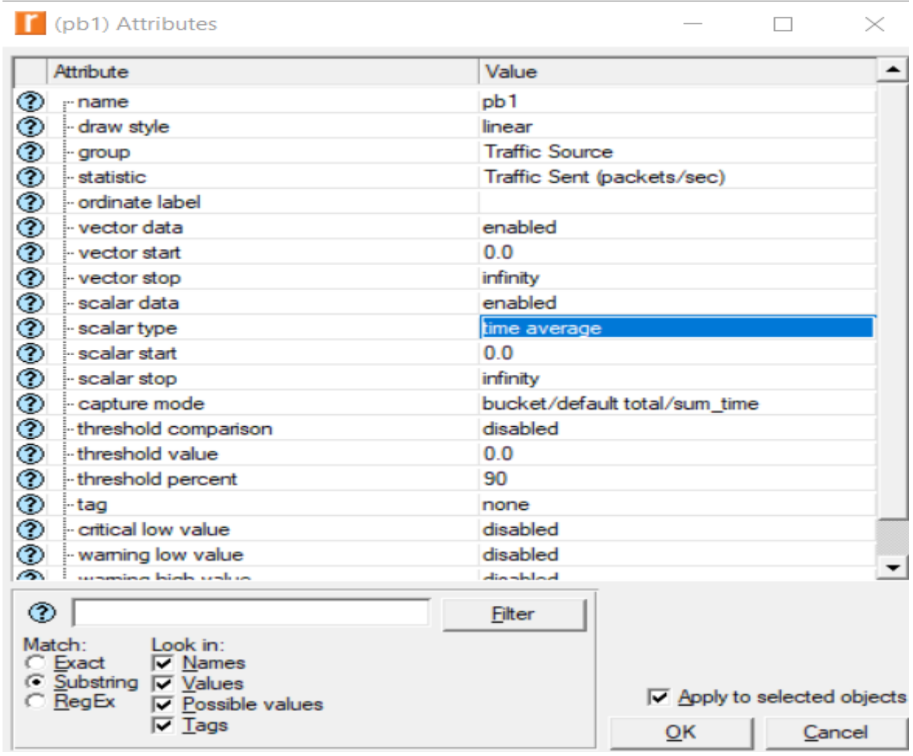
Attribute	Value
name	pb0
draw style	linear
group	Traffic Sink
statistic	Traffic Received (packets/sec)
ordinate label	
vector data	enabled
vector start	0.0
vector stop	infinity
scalar data	enabled
scalar type	time average
scalar start	0.0
scalar stop	infinity
capture mode	bucket/default total/sum_time
threshold comparison	disabled
threshold value	0.0
threshold percent	90
tag	none
critical low value	disabled
warning low value	disabled

Match: Exact [✓ Substring] RegEx

Look in: [✓ Names] [✓ Values] [✓ Possible values] [✓ Tags]

Apply to selected objects

d. Repeat the previous step with the Traffic Sent probe.



(pb1) Attributes

Attribute	Value
name	pb1
draw style	linear
group	Traffic Source
statistic	Traffic Sent (packets/sec)
ordinate label	
vector data	enabled
vector start	0.0
vector stop	infinity
scalar data	enabled
scalar type	time average
scalar start	0.0
scalar stop	infinity
capture mode	bucket/default total/sum_time
threshold comparison	disabled
threshold value	0.0
threshold percent	90
tag	none
critical low value	disabled
warning low value	disabled

Match: Exact [✓ Substring] RegEx

Look in: [✓ Names] [✓ Values] [✓ Possible values] [✓ Tags]

[✓ Apply to selected objects]

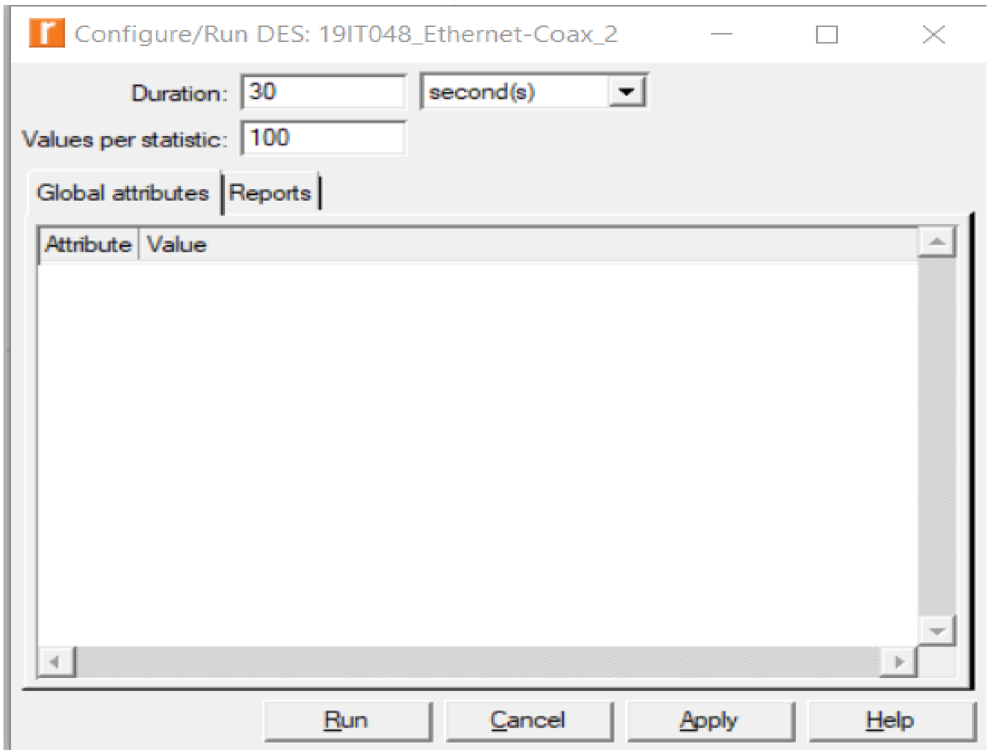
e. Select save from the File menu in the Probe Model window and then close thatwindow.

f. Now you are back to the Project Editor. Make sure to save your project

Run the Simulation

To run the simulation:

1. Click on the Configure/Run Simulation button: ⇒ Assign 30 second(s) (not hours) tothe Duration ⇒ Click Run. Depending on the speed of your processor, this may take several minutes to complete.



Configure/Run DES: 19IT048_Ethernet-Coax_2

Duration: 30 second(s)

Values per statistic: 100

2. After the simulation run is complete, click Close.
3. Save your project.
4. Select Duplicate Scenario from the Scenarios menu.
5. Name the new scenario Coax_1.
6. Right-click on any of the 30 nodes ⇒ Select Similar Nodes. Now all nodes in thenetwork are selected.

View the Results

To view and analyze the results:

1. Click on the View Results button: Now the Results Browseris open.

2. Select the DES Parametric Studies tab.

3. From the Results for drop-down menu, select Current Project.

4. Uncheck and check again the results for your project in order to check all the results.

5. Uncheck Coax_0025

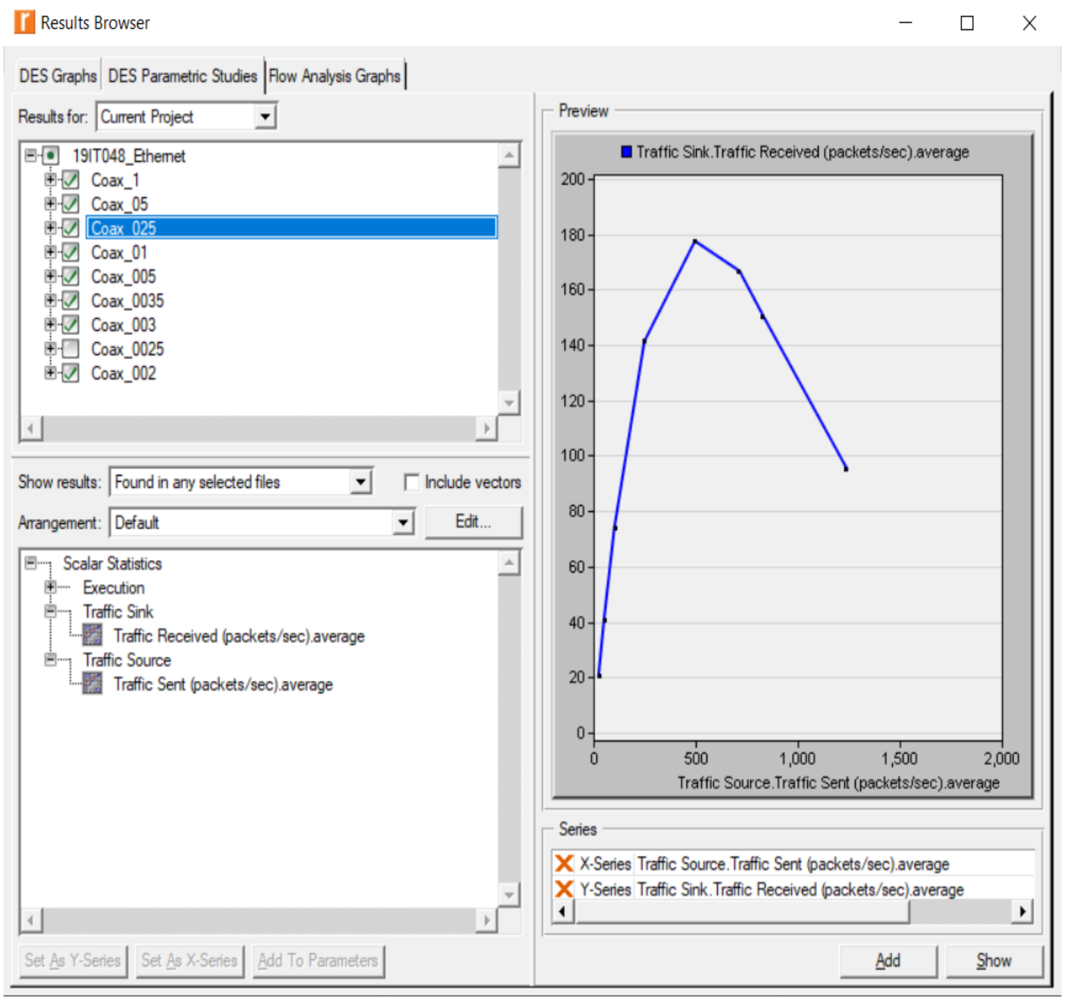
6. Uncheck Include vectors.

7. Expand the Scalar Statistics ⇒ Expand the Traffic Sink and Traffic Source.

8. Right click on Traffic Received and select Set as Y-Series

9. Right click on Traffic Sent and select Set as X-Series

10. The resulting graph should resemble the one below:



Results for: Current Project

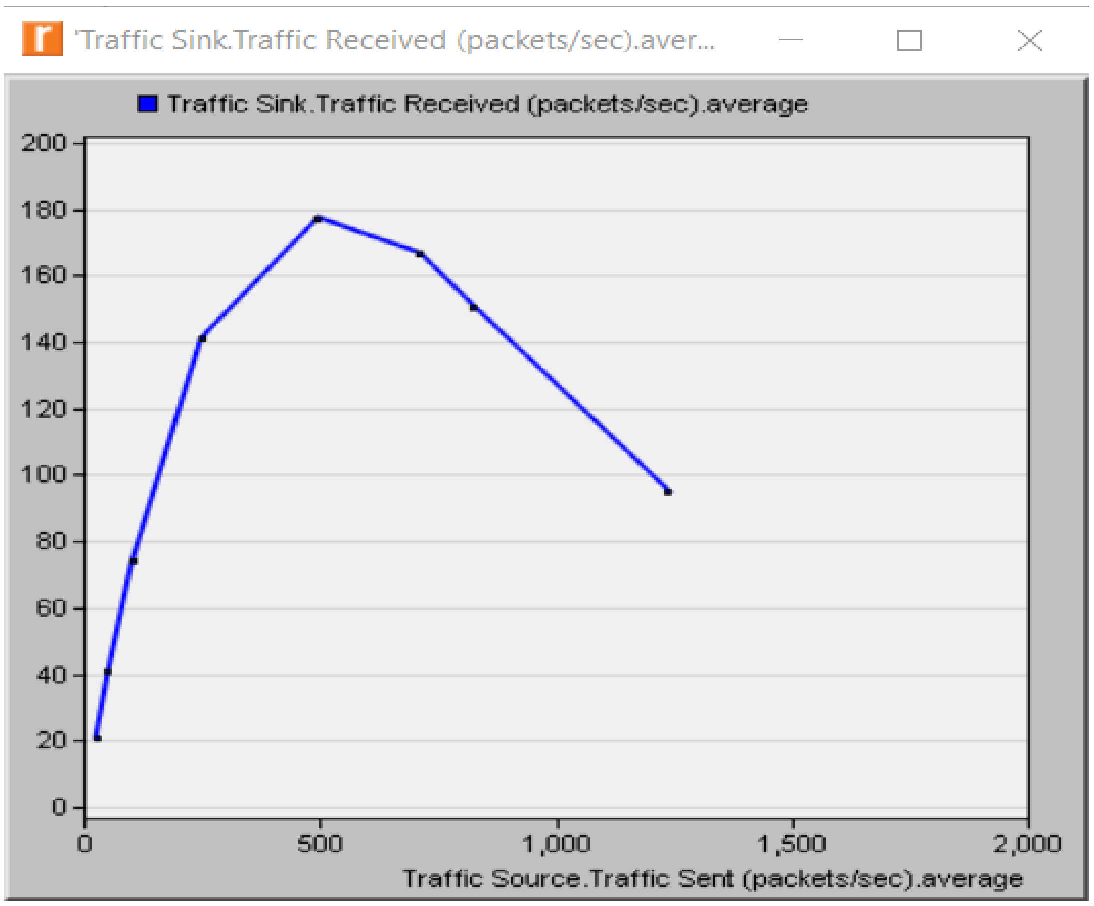
19IT048_Ethernet
- ✓ Coax_1
- ✓ Coax_05
- ✓ Coax_025
- ✓ Coax_01
- ✓ Coax_005
- ✓ Coax_0035
- ✓ Coax_003
- ☐ Coax_0025
- ✓ Coax_002

Show results: Found in any selected files ☐ Include vectors

Arrangement: Default

Scalar Statistics
- Execution
- Traffic Sink
  - Traffic Received (packets/sec).average
- Traffic Source
  - Traffic Sent (packets/sec).average

X-Series Traffic Source.Traffic Sent (packets/sec).average
Y-Series Traffic Sink.Traffic Received (packets/sec).average



Result:
Thus LAN based on Ethernet with a minimum of ten nodes is successfully simulatedand the performance under different load scenarios using OPNET are examined