

**R.I.T**

**REQUEST FOR PROPOSAL FOR A  
SIMULATION TOOL**

**TCET-760 NETWORK PLANNING & DESIGN**

**Prof. Joseph Nygate**



**Abhishek Sakpal | Rohan Vashi | Omkar Bhalekar**

# **PROJECT REPORT**

## Table of Contents

<b>SR NO.</b>	<b>CONTENTS</b>	<b>PAGE NO.</b>
1	INTRODUCTION	04
2	REQUEST FOR PROPOSAL (RFP)	05
2.1	INTRODUCTION	05
2.2	REQUIREMENTS	05
3	OVERVIEW OF GNS3	12
4	PERFORMANCE ANALYSIS AND RESULTS IN GNS3	13
4.1	RING TOPOLOGY	13
4.2	STAR TOPOLOGY	16
4.3	MESH TOPOLOGY	19
4.4	HYBRID TOPOLOGY	23
5	COMPARISON OF TOPOLOGIES BASED ON GNS3 RESULTS	29
6	NETWORK AUTOMATION USING ANSIBLE	31
7	OVERVIEW OF OPNET	34
8	PERFORMANCE ANALYSIS AND RESULTS IN OPNET	35
8.1	RING TOPOLOGY	35
8.2	STAR TOPOLOGY	40
8.3	MESH TOPOLOGY	45
9	COMPARISON OF TOPOLOGIES BASED ON OPNET RESULTS	50
10	CONCLUSION & RECOMMENDATION	51
11	BIBLIOGRAPHY	52

## 1. INTRODUCTION

### Deliverables

- This Project aims at analyzing the behavior of an emulator by assessing its capabilities and competence in terms of service metrics of a network implemented using this emulator and how these metrics differ for different network topologies
- Creating a Request for Proposal (RFP) which solicits preliminary requirements for a network emulator tool using
- Assessing the deliverables mentioned in the Request for Proposal for designing a network
- Emulating various network topologies like Star, Bus, Tree, Mesh and Hybrid in the proposed network simulation tool
- Automating a network using client-server model with the help of an agentless automation framework leeching on a remote server
- Calculating the service metrics for each of these topologies and comparing the parametric results based on the performance of the network
- Delineating, the statistical results in graphical format and based on the results obtained, inferring which network design stands out in terms of the service metrics obtained after implementation
- Scrutinizing the boons and pitfalls of using this emulator tool following thorough exploitation and the results obtained
- Implementation of mentioned deliverables using network emulator tool GNS3

## 2. Request For Proposal

### 2.1. Introduction

#### Purpose of this document

This RFP aims to resolve computer networking issues by implementing same network on network emulator or simulator. The objective of this RFP is to have healthy competition within the network simulation software market and to give network distributor complete overview of circuitry for fixing potential issues that can arise. The intention of this document is mentioned below:

- a. To provide detailed information for eligible bidders to understand the project objectives.
- b. To provide background and need of this project so that the efforts can meet the requirements.
- c. To make bidders understand about the organization and point of contact for any queries or more insights of the project.

### 2.2. Requirements

(FC=Fully Compliant, PC=Partially Compliant, NC=Non-Compliant)

#### 1. Preliminary Qualifications

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
1.1	Registration of Bidder	M	FC
1.2	No black list record	O	FC
1.3	International Presence	O	PC
1.4	Private Sector Company	M	FC
1.5	Fortune 500 inclusion	O	FC
1.6	Employee size > 1000	M	FC
1.7	Revenue > \$ 50 million	M	FC

#### 2. Supporting Operating System

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
2.1	Supportability with Windows	M	FC
2.2	Supportability with MAC OS X	M	FC
2.3	Supportability with Linux	M	FC

### 3. Hardware Requirements

Sr. No.	Description	Mandatory/Optional	FC/PC/NC
3.1	The Processor of machine should support dual core and superior versions.	M	FC
3.2	Max memory Utilization should not be more than 4 GB.	M	FC
3.3	Maximum available disk space not less than 1 GB.	M	FC
3.4	Minimum 35GB SSD	M	FC
3.5	i7 CPU and 4 or more logical cores AMD V	O	FC

### 4. Purveyor's Name

Sr. No.	Description	Mandatory/Optional	FC/PC/NC
4.1	CISCO	M	FC
4.2	JUNIPER Networks	M	FC
4.3	Aerohive Networks	M	FC
4.4	Huawei	M	FC
4.5	Brocade	O	NC
4.6	Linksys	O	PC
4.7	Asus	O	FC

### 5. Devices

#### 1. Switches

Sr. No.	Description	Mandatory/Optional	FC/PC/NC
5.1.1	Managed Switch	M	FC
5.1.2	Layer 2 IOS Switch	O	FC
5.1.3	Unmanaged Switch	M	FC

## 2. Routers

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
5.2.1	Core Routers	M	FC
5.2.2	Edge Routers	M	FC
5.2.3	Cloud Routers	M	FC
5.2.4	Backbone Routers	O	PC
5.2.5	SDN Internet Routers	O	PC

## 3. Guest Device

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
5.3.1	Centos	M	FC
5.3.2	Kali Linux	M	FC
5.3.3	Network Automation	M	FC
5.3.4	Python, Go, Perl, PHP	M	FC

## 6. Display Requirements

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
6.1	720p (1280*720) HD	M	FC
6.2	1080p (1920*1080) FHD	M	FC
6.3	1440p (2560*1440) QHD	M	FC

## 7. Subscription / Outlay

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
7.1	The license cost per device should be less than \$150.	M	FC
7.2	No upgradation cost	M	FC

## 8. Firewalls

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
8.1	Stateful Filter	M	FC
8.2	Proxy	M	FC
8.3	Deep Packet Inspection	M	FC

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
8.4	Application Level Gateway	O	NC
8.5	Sophos XG Firewall	O	PC

## 9. Configurations

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
9.1	Router Configuration	M	FC
9.2	Security Configuration	M	FC
9.3	VLAN configurations	M	FC
9.4	Port Configuration	M	FC
9.5	Packet Filtering	M	FC
9.6	IP configuration	M	FC
9.7	VPN	M	FC

## 10. Delineation of Topologies

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
10.1	Ring Topology	M	FC
10.2	Star Topology	M	FC
10.3	Mesh Topology	M	FC
10.4	Hybrid Topology	M	FC
10.5	Tree Topology	M	FC

## 11. Availability of Connecting Links

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
11.1	Serial Links (Straight-Through)	M	FC
11.2	Ethernet Link (Crossover)	M	FC
11.3	Twisted Pair	O	PC
11.4	Coaxial Cable	M	FC



## 12. Protocols

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
12.1	VLAN Trunking	M	FC
12.2	RIPv1	M	FC
12.3	RIPv2	M	FC
12.4	OSPF	M	FC
12.5	EIGRP	M	FC
12.6	Spanning Tree Protocol	M	FC
12.7	ATM	O	NC

## 13. Virtualization

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
13.1	Docker	M	FC
13.2	Ansible	M	FC
13.3	Kernel Based Virtual Machine	M	FC

## 14. Incorporation of Network Automation

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
14.1	Ansible supported network management	M	FC
14.2	Network migration and configuration with ansible.	M	FC
14.3	Network containerization	M	FC

## 15. Test & Results

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
15.1	Execution Time should not be more than 10 seconds.	M	FC
15.2	Reverse Compatibility	O	NC
15.3	Forward Compatibility	M	FC
15.4	Graphical & Chart Representation of Results	O	NC

## 16. Technical Support

Sr. No.	Description	Mandatory/ Optional	FC/PC/NC
16.1	24*7 Text based support	M	FC
16.2	24*7 Email based support	M	FC
16.3	Simulation videos with tutorials	M	FC
16.4	Instant query resolution with community chats	M	FC
16.5	Blogs focusing on upcoming releases	O	PC

### 3. Overview of GNS 3

GNS 3 stands for Graphical Network Simulator which basically is a network emulator tool providing discernible working environments for designing, testing and analyzing networks, but without the need of any physical hardware access. Its main function is to provide real-time simulation of networks as it supports a broad range of network devices and paraphernalia based on various platforms.

Dynamips, a script written in C language forms the basis for GNS3 in the back-end wherein this program helps emulate actual routers by following the instructions mapped in the memory virtually in a sequential fashion. On the other hand, Dynagen which runs on Python forms the front-end platform supporting in-built <sup>1</sup>INI configuration files which makes this tool more versatile in terms of the scope to access a particular network device.

Being and user-friendly emulator along with a simulator, GNS 3 offers the freedom to work with real network device images which in essence are the operating systems on which physical routers function and with the help of which we can design various topologies. It also allows the user to simulate networks on virtual platforms such as VMware wherein it comes with a GNS VM bundle providing additional advantages such as support for latest or updated image versions as well as limitless use of network devices along with access to GNS3 via remote servers.

Besides, its ability to imprint data packets in a file on the disk makes it a better affiliate of Wireshark enabling it to sniff and capture packets thereby making packet analysis and categorization easier for reading protocol statistics and summarization of packet conversations.



---

<sup>1</sup> INI config Files -Simple Text based files necessary for initial configuration of a system for e.g.: MS-DOS in Windows

### **PRO'S of GNS3**

- The prime advantage which puts GNS3 in limelight is its ability to emulate real
- In fiscal sense, GNS3 is a full version open source software licensed under GNU GPLv3 available for public access along with its source code on various online resources
- Compatible with all well-known operating systems such as Linux, Mac OS and Windows
- Provides compatibility with plethora of network devices to choose from
- Pre-deployment simulation to test and analyze the behavior of networks for future deployments
- Support for various virtual platforms such VirtualBox, VMware etc. which helps optimal utilization of hardware resources.

### **PITFALLS of GNS3**

- Flip-side of being an emulator is that, it introduces latency due to its function to read router boot process instructions in sequential manner.
- CISCO images need to be purchased for various router series.
- It utilizes CPU resources very heavily.
- Unable to provide graphical representation of service metrics.
- Inaccessibility of virtual end devices which demands manual configuration of routers as a substitute to PCs.
- Inability to analyze data traffic and network behavior in real-time.

## 4. Performance Analysis and Results in GNS3

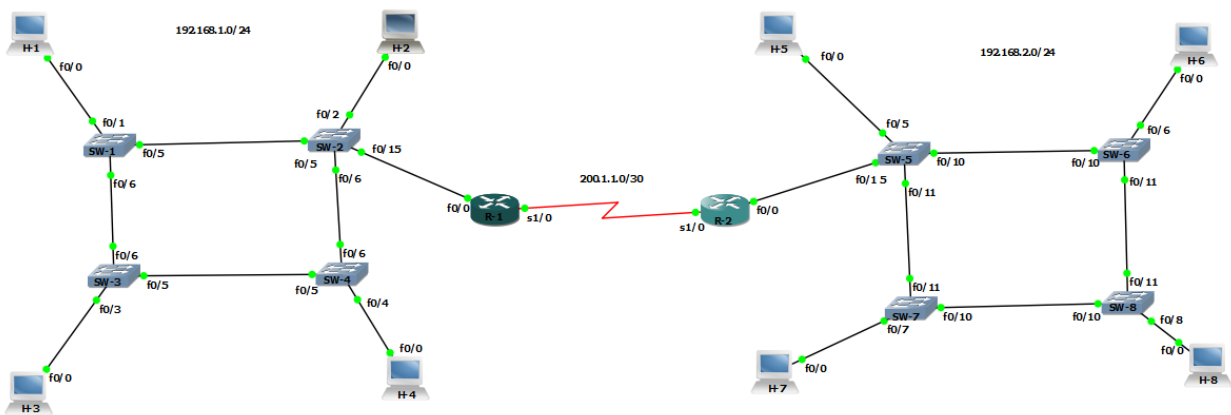
### 4.1. Ring Topology

Ring topology is a circular loop formed by network devices in which devices are connected to two other devices.

In this topology, the traffic forwarded by a source passes through all the devices present between source and destination.

The security and privacy of confidential data is compromised in this architecture.

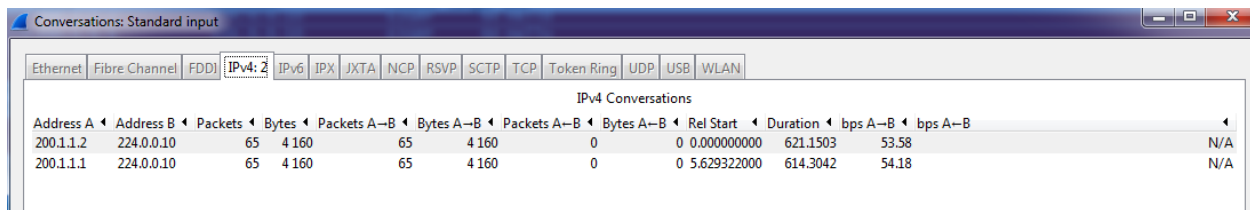
This topology is cheaper than full mesh topology.



Here, the devices are connected in two different rings with each connected via two routers. When one PC wants to communicate with another, it passes through switches and routers to reach other networks. Here, the routers are configured with EIGRP protocol.

## Performance Analysis

This is what packet looks like which generates traffic in the network.



Conversations: Standard input

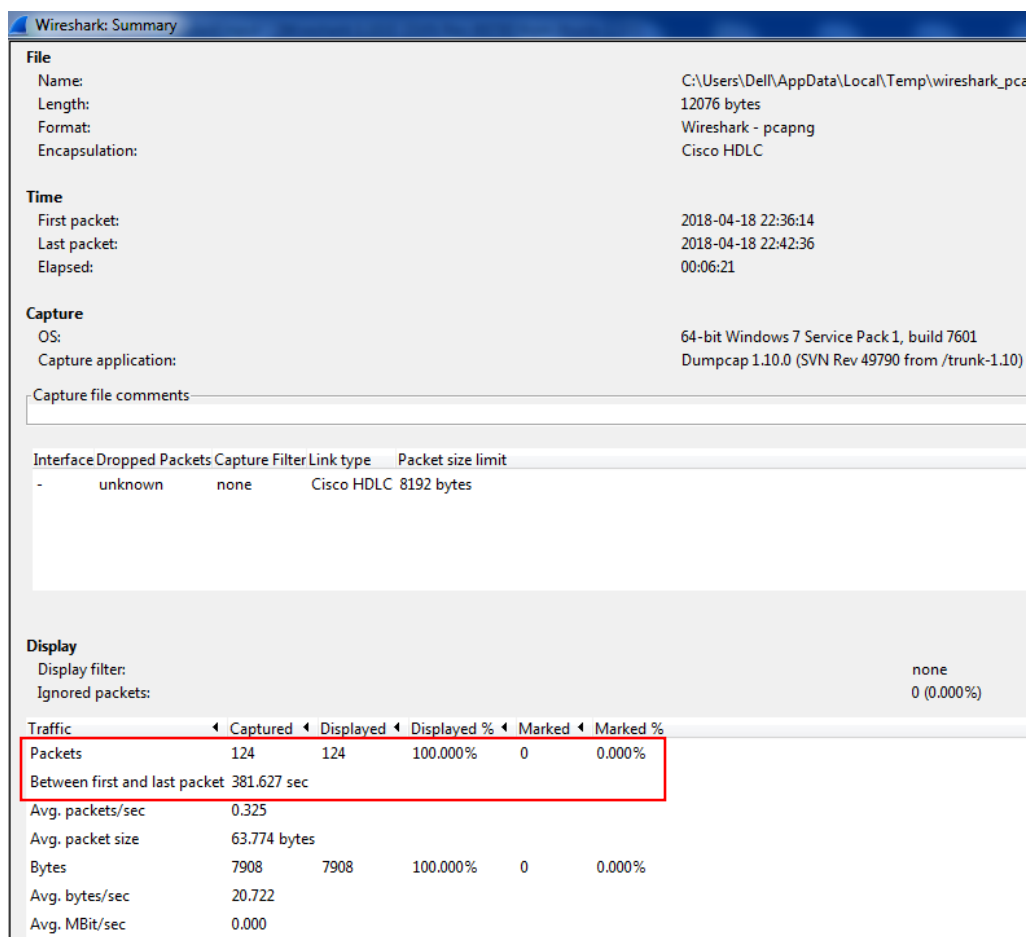
Ethernet Fibre Channel FDDI IPv4: 2 IPv6 IPX JXTA NCP RSVP SCTP TCP Token Ring UDP USB WLAN

IPv4 Conversations

Address A	Address B	Packets	Bytes	Packets A→B	Bytes A→B	Packets B→A	Bytes B→A	Rel Start	Duration	bps A→B	bps B→A
200.1.1.2	224.0.0.10	65	4160	65	4160	0	0	0.000000000	621.1503	53.58	N/A
200.1.1.1	224.0.0.10	65	4160	65	4160	0	0	5.629322000	614.3042	54.18	N/A

As GNS3 is unable to produce graphs, we are showing the results by capturing packets in Wireshark (GNS3 is compatible with Wireshark).

Delay:



Wireshark: Summary

**File**

Name: C:\Users\Delh\AppData\Local\Temp\wireshark\_pca  
Length: 12076 bytes  
Format: Wireshark - pcapng  
Encapsulation: Cisco HDLC

**Time**

First packet: 2018-04-18 22:36:14  
Last packet: 2018-04-18 22:42:36  
Elapsed: 00:06:21

**Capture**

OS: 64-bit Windows 7 Service Pack 1, build 7601  
Capture application: Dumpcap 1.10.0 (SVN Rev 49790 from /trunk-1.10)  
Capture file comments:

Interface	Dropped Packets	Capture Filter	Link type	Packet size limit
-	unknown	none	Cisco HDLC	8192 bytes

**Display**

Display filter: none  
Ignored packets: 0 (0.000%)

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	124	124	100.000%	0	0.000%
Between first and last packet	381.627 sec				
Avg. packets/sec	0.325				
Avg. packet size	63.774 bytes				
Bytes	7908	7908	100.000%	0	0.000%
Avg. bytes/sec	20.722				
Avg. MBit/sec	0.000				

As graph is not present, we will calculate delay for ring topology.

Here, no. of packets are 124 and time between first and last packet is 381.627 seconds.

Therefore, delay can be calculated as  $[381.627 / 124] = 3.07$  seconds

### Throughput:

```
R-1#show ip eigrp traffic
IP-EIGRP Traffic Statistics for AS 123
  Hellos sent/received: 342/248
  Updates sent/received: 3/3
  Queries sent/received: 0/0
  Replies sent/received: 0/0
  Acks sent/received: 2/0
  Input queue high water mark 2, 0 drops
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
  Hello Process ID: 206
  PDM Process ID: 205
```

Here, no. of packets sent and received are 342 and 248 respectively. So, throughput can be calculated as below:

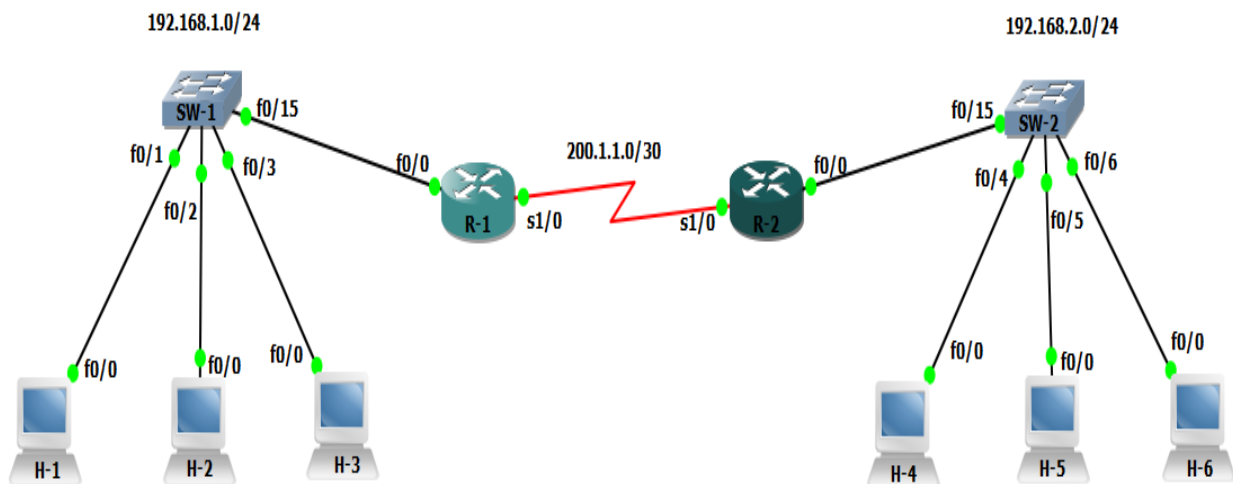
**Throughput for Ring Topology** = [ No. of packets received / No. of packets transmitted]

= [ 248 / 342 ]

= 72.51 %

## 4.2. Star Topology

- Star topology presents a centralized architecture in which all the nodes are connected to a central station which controls the traffic of all its peripheral devices
- The star topology designed below depicts two distinct and centralized star networks interconnected via two routers with two switches in each network
- All the nodes in a network are connected to a central device, switch in this case and the traffic can traverse to other networks using routers
- This design provides limiting impact in an event of a node failure thereby rendering other nodes unaffected
- Additional nodes can be installed or decommissioned easily without affecting traffic of other stations
- Also, the total links in a star topology are same as the total number of devices present in the network





a) This is what a packet looks like which generates traffic in the network

Conversations: Standard input

Ethernet	Fibre Channel	FDDI	IPv4: 2	IPv6	IPX	JXTA	NCP	RSVP	SCTP	TCP	Token Ring	UDP	USB	WLAN
----------	---------------	------	---------	------	-----	------	-----	------	------	-----	------------	-----	-----	------

IPv4 Conversations

Address A	Address B	Packets	Bytes	Packets A→B	Bytes A→B	Packets B→A	Bytes B→A	Rel Start	Duration	bps A→B	bps B→A
200.1.1.1	224.0.0.10	96	6144	96	6144	0	0	0.000000000	439.9958	111.71	N/A
200.1.1.2	224.0.0.10	82	5248	82	5248	0	0	0.575408000	439.0148	95.63	N/A

Delay:

Wireshark: Summary	
<b>File</b>	
Name:	C:\Users\Dell\AppData\Local\Temp\wireshark_pcapng_-_20180418230557_a18452
Length:	16172 bytes
Format:	Wireshark - pcapng
Encapsulation:	Cisco HDLC
<b>Time</b>	
First packet:	2018-04-18 23:01:22
Last packet:	2018-04-18 23:06:02
Elapsed:	00:04:39
<b>Capture</b>	
OS:	64-bit Windows 7 Service Pack 1, build 7601
Capture application:	Dumpcap 1.10.0 (SVN Rev 49790 from /trunk-1.10)
Capture file comments	
Interface Dropped Packets Capture Filter Link type Packet size limit	
-	unknown none Cisco HDLC 8192 bytes
<b>Display</b>	
Display filter:	none
Ignored packets:	0 (0.000%)
Traffic	Captured
Packets	167
Between first and last packet	279.440 sec
Avg. packets/sec	0.598
Avg. packet size	63.617 bytes
Bytes	10624
Avg. bytes/sec	38.019
Avg. MBit/sec	0.000

b) The summary displayed below for a fully functional star topology shows the total number of packets captured by Wireshark to be 167 and the duration between capture of first packet and last packet to be 279.440 seconds which facilitates calculating delay for the packet transmission given as

Time between first and last packet = 279.440seconds

Total Number of packets captured = 167

Delay = (Time between first and last packet) / (Total Number of packets captured)

= 279.440/167

=1.67 seconds

Throughput:

```
R-1#show ip eigrp traffic
IP-EIGRP Traffic Statistics for AS 123
  Hellos sent/received: 157/70
  Updates sent/received: 3/3
  Queries sent/received: 0/0
  Replies sent/received: 0/0
  Acks sent/received: 0/2
  Input queue high water mark 2, 0 drops
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
  Hello Process ID: 205
  PDM Process ID: 204
```

c). The throughput is depicted by the number of hello packets sent and received

Number of hello messages sent = 157

Number of hello messages received = 70

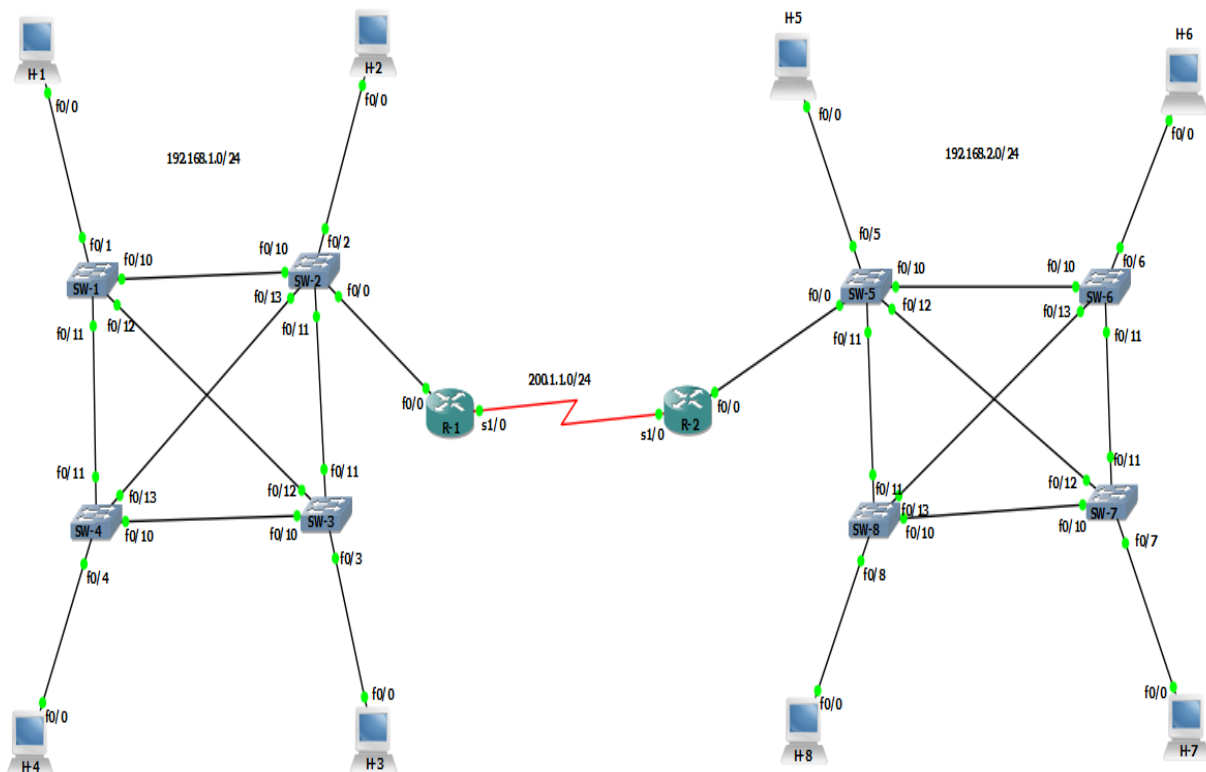
Throughput = Number of hello messages received / Number of hello messages sent

= 70/157

= 44.58%

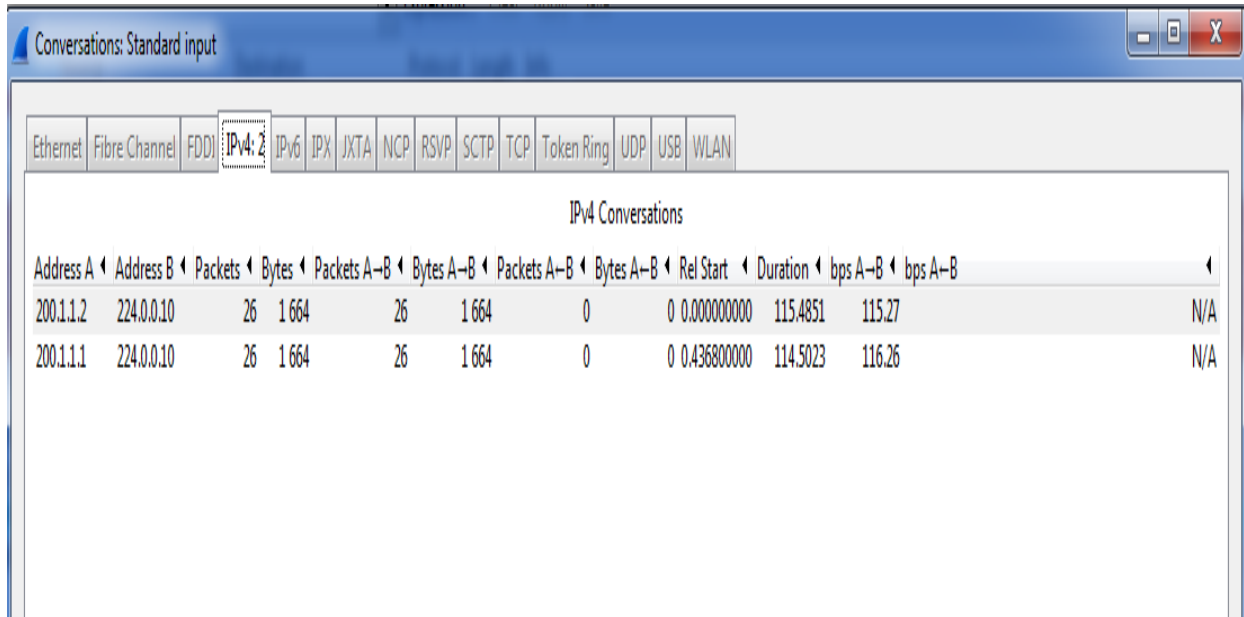
### 4.3. Mesh Topology

- Mesh Topology is designed keeping in mind provision of alternate or redundant routes during link collapse or node breakdowns thereby making it easier and faster to relay data in an efficient manner which adds to the reliability of the network
- The mesh topology designed below depicts two distinct and decentralized mesh networks interconnected via two routers with four switches in each network
- All the switches or nodes in a network are interconnected to every other node forming a mesh structure and the traffic can traverse to other networks using routers
- This design provides robustness in terms of traffic re-routing and addition of new devices in the network without affecting the data exchange amongst other nodes
- In theoretical terms, finding required number of links, given the number of devices can be stated as  $[n \times (n-1)] / 2$  which makes it quite simpler to evaluate the cost of a network
- A ubiquitous routing protocol in networking domain known as EIGRP (Enhanced Interior Gateway Routing Protocol) complementing the features of mesh network is configured onto the mesh topology fulfilling the requirements of quick backup routes



## Performance Parameters

- a) This is what a packet looks like which generates traffic in the network



IPv4 Conversations											
Address A	Address B	Packets	Bytes	Packets A-B	Bytes A-B	Packets A-B	Bytes A-B	Rel Start	Duration	bps A-B	bps A-B
200.1.1.2	224.0.0.10	26	1 664	26	1 664	0	0	0.000000000	115.4851	115.27	N/A
200.1.1.1	224.0.0.10	26	1 664	26	1 664	0	0	0.436800000	114.5023	116.26	N/A

- b) Bundled with GNS3 comes a packet sniffer and analyzer known as Wireshark which helps capture and evaluate packets and based on that provides various metrics pertaining to captured packets.

**Delay:**

Wireshark: Summary

File

Name:

C:\Users\Del\l\AppData\Local\Temp\wireshark\_pcapng\_-\_20180418233414\_a18528

Length:

7828 bytes

Format:

Wireshark - pcapng

Encapsulation:

Cisco HDLC

Time

First packet:

2018-04-18 23:18:38

Last packet:

2018-04-18 23:20:36

Elapsed:

00:01:58

Capture

OS:

64-bit Windows 7 Service Pack 1, build 7601

Capture application:

Dumpcap 1.10.0 (SVN Rev 49790 from /trunk-1.10)

Capture file comments

Interface

Dropped Packets

Capture Filter

Link type

Packet size limit

-

unknown

none

Cisco HDLC

8192 bytes

Display

Display filter:

none

Ignored packets:

0 (0.000%)

Traffic

Captured

Displayed

Displayed %

Marked

Marked %

Packets

79

79

100.000%

0

0.000%

Between first and last packet

118.075 sec

Avg. packets/sec

0.669

Avg. packet size

64.608 bytes

Bytes

5104

5104

100.000%

0

0.000%

Avg. bytes/sec

43.227

Avg. MBit/sec

0.000

- The summary displayed below for a fully functional mesh topology shows the total number of packets captured by Wireshark to be 79 and the duration between capture of first packet and last packet to be 118.075 seconds which facilitates calculating delay for the packet transmission given as

Time between first and last packet = 118.075 seconds

Total Number of packets captured = 79

Delay = (Time between first and last packet) / (Total Number of packets captured)

= 118.075/79

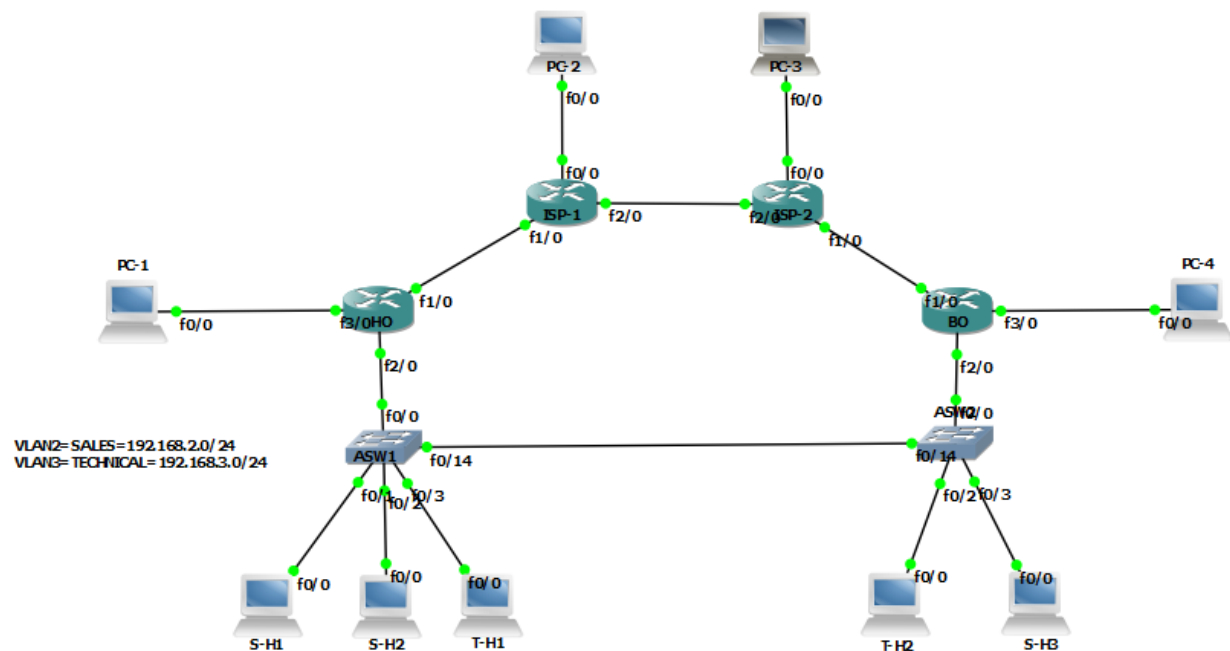
= **1.49 seconds**

Throughput:

```
R-1#show ip eigrp traffic
IP-EIGRP Traffic Statistics for AS 123
  Hellos sent/received: 245/115
  Updates sent/received: 3/3
  Queries sent/received: 0/0
  Replies sent/received: 0/0
  Acks sent/received: 2/1
  Input queue high water mark 2, 0 drops
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
  Hello Process ID: 182
  PDM Process ID: 162
```

- a) The throughput is depicted by the number of hello packets sent and received
- Number of hello messages sent = 245
- Number of hello messages received = 115
- Throughput = Number of hello messages received / Number of hello messages sent
- = 115/245
- = 47%

#### 4.4. Hybrid Topology



This topology is design as an organization setup which include hybrid topology. This is the real-world example of enterprise networking. This topology has two aspect one is the connection of employer machines and second is the quality of service.

Every organization is connected with any internet service provider because of the internet demand so this topology is also showing the connection of ISP to their server and PC.

Organization point of view we configured VPN and VLAN also for local networks in addition to this we also configured encapsulation so that every traffic goes in secure manner.

Topology showing the sales department and technical department employers machine with head office and branch office servers.

Departmental machines are considered as same subnet machine so basically sales department and technical department employers PC's are in same subnet and they are connected by two switches ASW-1 and ASW-2.

As per the connection point of view we configured VLAN-2 and VLAN-3 between technical and sales department employers by making one interface as trunk.

```

interface FastEthernet0/0
  switchport mode trunk
!
interface FastEthernet0/1
  switchport access vlan 2
!
interface FastEthernet0/2
  switchport access vlan 2
!
interface FastEthernet0/3
  switchport access vlan 3

```

```
S-H1#ping 192.168.2.3
```

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.3, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 32/35/44 ms
S-H1#

```

```

interface FastEthernet2/0.1
  encapsulation dot1Q 2
  ip address 192.168.2.100 255.255.255.0
  no snmp trap link-status
!
interface FastEthernet2/0.2
  encapsulation dot1Q 3
  ip address 192.168.3.100 255.255.255.0
  no snmp trap link-status

```

Employer PC are encapsulated with dot1q protocol and they will be giving default gateway, so every department PC will communicate with outer world via this default gateway.

Now after establishing connection between two VLAN and local subnet PCs, going upward the topology and making connection between head office and branch office.

For communication between Head office and Branch office we configured OSPF routing protocol. OSPF is used because it is the reliable and it can communicate up to unlimited hops.

Here is the traffic measured for the OSPF protocol.

```

Rcvd: 4870 total, 0 errors
      4695 hello, 10 database desc, 3 link state req
      102 link state upds, 60 link state acks, 0 invalid
Sent: 6689 total
      6463 hello, 11 database desc, 3 link state req
      144 link state upds, 68 link state acks, 0 invalid

```

For the organization security point of view dot1q encapsulation also implemented.



```

interface FastEthernet2/0.1
 encapsulation dot1Q 2
 ip address 192.168.2.100 255.255.255.0
 no snmp trap link-status
!
interface FastEthernet2/0.2
 encapsulation dot1Q 3
 ip address 192.168.3.100 255.255.255.0
 no snmp trap link-status

```

Another aspect of this topology is quality of service because organization ultimate goal is having a good quality of service which includes packet filtering and then matching the packets by their configured precedence and try to evaluate the performance on the basis of the various parameters.

Quality of service can be configured on cisco devices so here we configured on the head office and branch office router.

```

class-map match-all ospf
 match protocol ospf
class-map match-all OSPF
 match protocol ospf
class-map match-all MATCH_HTTP
 match access-group 105
class-map match-all ICMP_TO_CORE
 match precedence 1
class-map match-all HTTP_TO_CORE
 match precedence 3
class-map match-all MATCH_ICMP
 match access-group 101
!
!
policy-map FROM_HOST
 class MATCH_ICMP
  set precedence 1
 class MATCH_HTTP
  set precedence 3
policy-map TO_CORE
 class ICMP_TO_CORE
  bandwidth 8
  police cir 8000
  conform-action transmit
  exceed-action drop
 class HTTP_TO_CORE
  bandwidth 10000
 class ospf
  set precedence 7
  priority 1000
 class OSPF
  set precedence 7
  priority 1000

```

Now if you see the snapshot various classes are made to match the packets of various services like ICMP, HTTP and OSPF.

For HTTP traffic we configured PC-1 to 4 as a HTTP server and also enabled HTTP's feature for security.

```
ip default-gateway 10.10.1.1
ip http server
ip http authentication local
ip http secure-server
ip classless
```

Security is an important feature of an organization so here an Access control list is also implemented which filters inbound and outbound packets.

```
access-list 101 permit icmp any any
access-list 101 remark "match icmp"
access-list 102 permit ip 10.10.1.0 0.0.0.255 10.10.4.0 0.0.0.255
access-list 105 remark "match http"
access-list 105 permit tcp any any eq www
!
```

Now after configuring every protocol and feature, now we are simulating the topology and try to match the ICMP and HTTP packets.

```

HO#show policy-map interface f1/0
FastEthernet1/0

Service-policy output: TO_CORE

Class-map: ICMP_TO_CORE (match-all)
  10 packets, 1140 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: precedence 1
  Queueing
    Output Queue: Conversation 265
    Bandwidth 8 (kbps) Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0
  police:
    cir 8000 bps, bc 1500 bytes
    conformed 10 packets, 1140 bytes; actions:
      transmit
    exceeded 0 packets, 0 bytes; actions:
      drop
    conformed 0 bps, exceed 0 bps

Class-map: HTTP_TO_CORE (match-all)
  8 packets, 464 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: precedence 3
  Queueing
    Output Queue: Conversation 266
    Bandwidth 10000 (kbps) Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0

```

```

Class-map: ospf (match-all)
  1808 packets, 170772 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: protocol ospf
  QoS Set
    precedence 7
    Packets marked 1811
  Queueing
    Strict Priority
    Output Queue: Conversation 264
    Bandwidth 1000 (kbps) Burst 25000 (Bytes)
    (pkts matched/bytes matched) 69/7322
    (total drops/bytes drops) 0/0

```

In conclusion, this topology is the perfect model of enterprise networking and it is simulated in GNS3 which has everything as a package like VPN configuration, VLAN configuration, Routing protocol, quality of service, ACL and encapsulation. So GNS3 is capable to design large scale network and also it is able to simulate multiple devices at a time. GNS3 also able to configure cisco routers and switches and various protocols including network security protocol and quality of service.

## 5. Comparison of topologies based on GNS3 Results

	Delay (seconds)	Throughput (%)
Ring	3.07	72.51
Star	1.67	44.58
Mesh	1.49	47

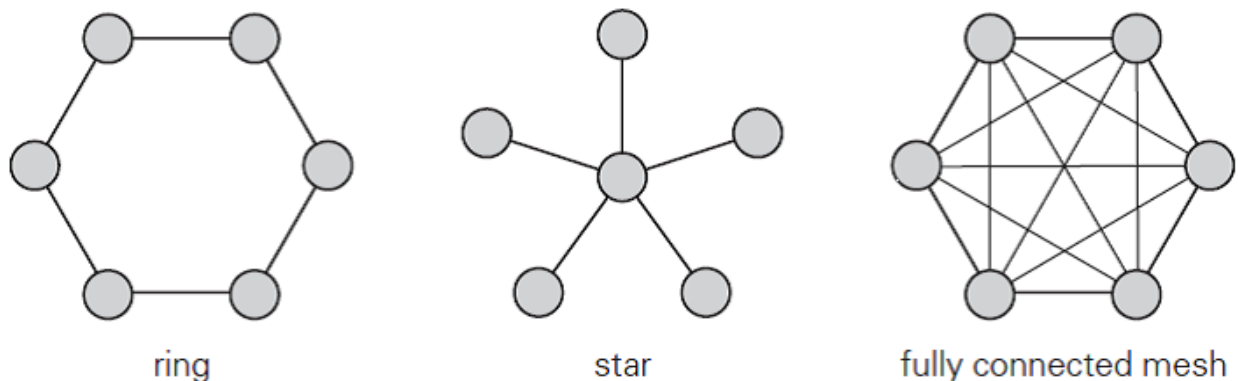
Here, are the results of ring, star and mesh topologies. We can observe the delay and throughput for all the three topologies.

In terms of Delay, **Mesh** topology is better than star and ring respectively.

In terms of throughput, **Ring** topology is better than mesh and star respectively.

### Cost Evaluation:

The cost factor is very important for network architects. Generally, selection of topology is majorly based on cost of infrastructure.



Three topologies represented above are ring, star and mesh. Each topology has 6 network devices. These devices are connected to each other using connecting links. Cost of each topology differs by number of links present in the network.

Topology	Ring	Star	Mesh
No. of links	5	6*	13

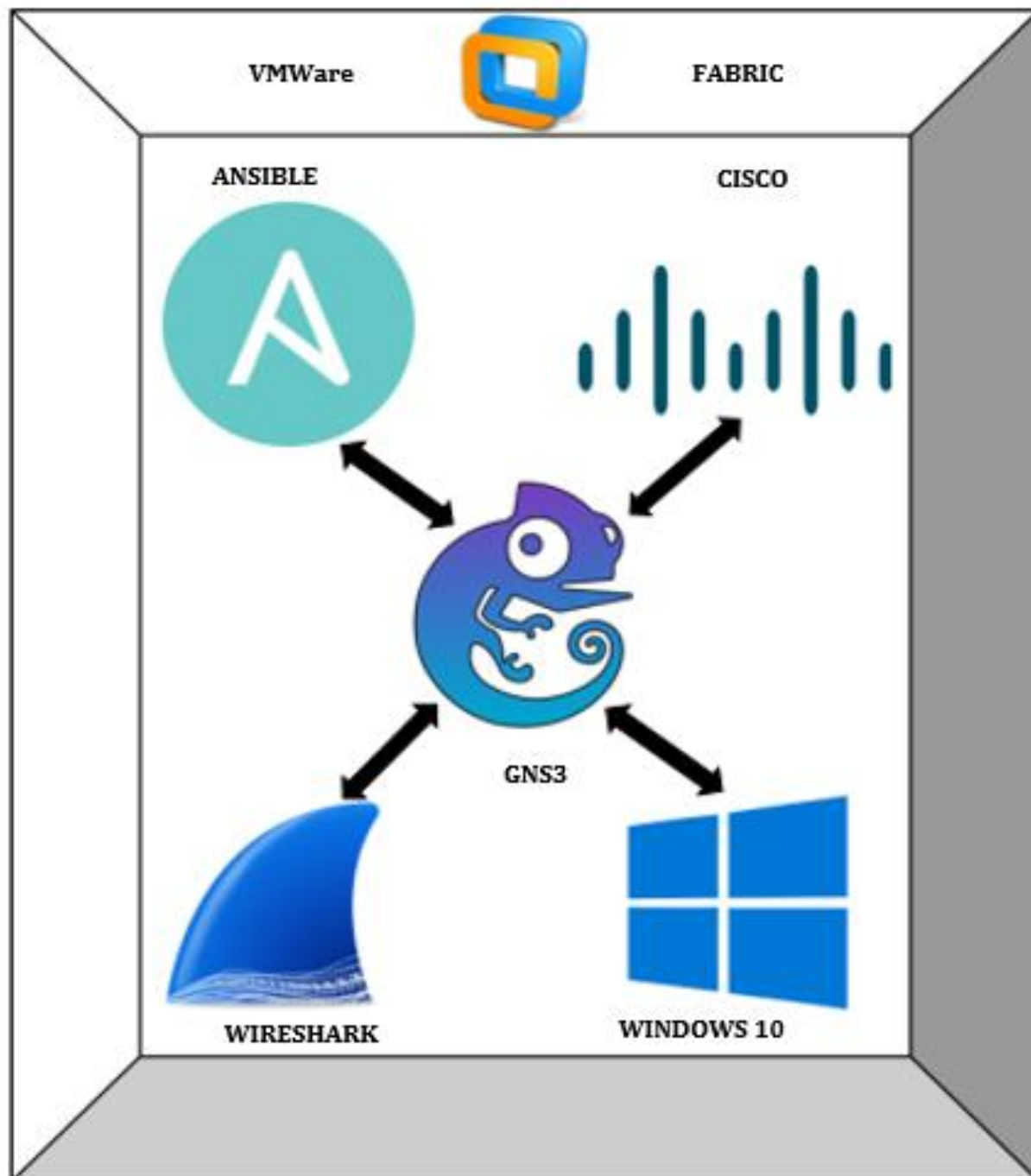
\*Considering Central Unit as hub and rest all networking devices

Assuming cost of each link is same. So, the total cost of mesh topology is highest than all other topologies.

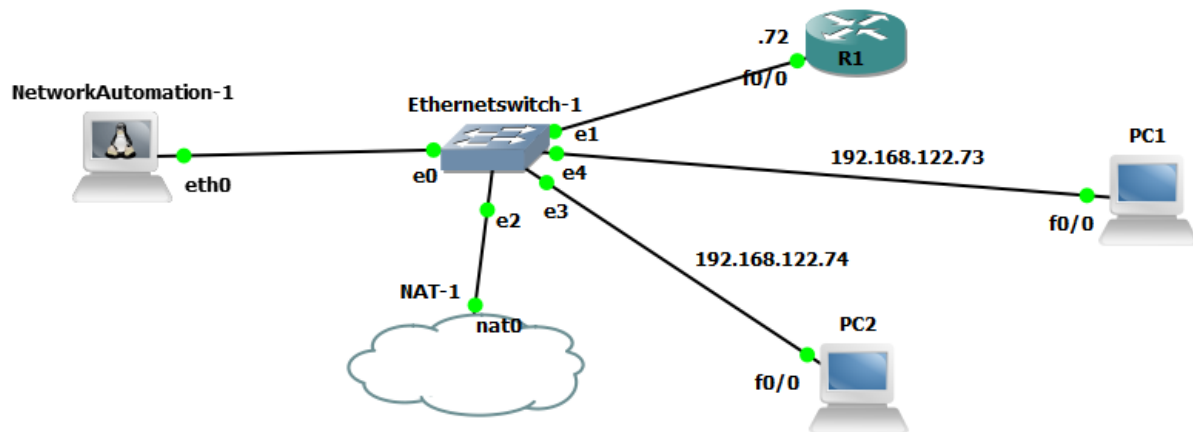
Topology	1 (Best)	2 (Intermediate)	3 (Worst)
Delay	Mesh	Star	Ring
Throughput	Ring	Mesh	Star
Cost	Ring	Star	Mesh

From above result table we can discard star topology as it is beaten by mesh in two factors i.e. delay and throughput. When delay is the criteria, we will select Mesh topology. But, when throughput and cost is the criteria, we will go with Ring Topology.

## Software Bundle for implementation of Network Automation



## 6. Network Automation Using Ansible



### Network Automation using Ansible

Here, the Network Automation-1 is a server with Ansible enabled in it.

- GNS3 software has available appliance which is network automation container. This container can be downloaded from the GNS3 marketplace.
- Network automation container has Linux kernel and it has already configured ansible and docker which are the tools used for network automation.
- Ansible is the most important tool in network automation ansible basically is agentless architecture which runs on Server-client model.
- Ansible clients are also known as node or ansible host. Ansible main advantage is administrator does not require to install ansible in every host machine.
- Ansible runs on SSH protocol so hosts machine should have the same public key that of ansible server.
- Network automation generally performs task which called playbook and task are like getting running configuration information of each and every host machine.
- By taking remote terminal using SSH connection ansible server can configure host machine also.
- Above topology is performing simple network automation task which includes checking connectivity of each machine.
- As per the topology it has one network automation container and it is connected to routers and some other PCs through a Layer-2 Link.

```
NetworkAutomation-1
udhcpd (v1.24.2) started
Sending discover...
Sending discover...
Sending discover...
Sending select for 192.168.122.238...
Lease of 192.168.122.238 obtained, lease time 3600
root@NetworkAutomation-1:~#
root@NetworkAutomation-1:~#
root@NetworkAutomation-1:~#
```

- As per the topology which has NAT cloud. This cloud is like DHCP server who's work is to give IP address of range 192.168.122.0-255/24.
- Now after obtaining IP address now configuring the host resolution process which include IP addresses of hosts.

```
NetworkAutomation-1
Bad owner or permissions on /root/.ssh/config
root@NetworkAutomation-1:~# ping R1
PING R1 (192.168.122.72) 56(84) bytes of data.
64 bytes from R1 (192.168.122.72): icmp_seq=1 ttl=255 time=116 ms
64 bytes from R1 (192.168.122.72): icmp_seq=2 ttl=255 time=7.14 ms
64 bytes from R1 (192.168.122.72): icmp_seq=3 ttl=255 time=7.15 ms
64 bytes from R1 (192.168.122.72): icmp_seq=4 ttl=255 time=5.68 ms
64 bytes from R1 (192.168.122.72): icmp_seq=5 ttl=255 time=7.43 ms
64 bytes from R1 (192.168.122.72): icmp_seq=6 ttl=255 time=11.9 ms
64 bytes from R1 (192.168.122.72): icmp_seq=7 ttl=255 time=3.35 ms
64 bytes from R1 (192.168.122.72): icmp_seq=8 ttl=255 time=11.4 ms
64 bytes from R1 (192.168.122.72): icmp_seq=9 ttl=255 time=4.53 ms
64 bytes from R1 (192.168.122.72): icmp_seq=10 ttl=255 time=1.95 ms
64 bytes from R1 (192.168.122.72): icmp_seq=11 ttl=255 time=11.4 ms
64 bytes from R1 (192.168.122.72): icmp_seq=12 ttl=255 time=10.8 ms
64 bytes from R1 (192.168.122.72): icmp_seq=13 ttl=255 time=11.5 ms
^C
--- R1 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12013ms
rtt min/avg/max/mdev = 1.953/16.262/116.876/29.226 ms
root@NetworkAutomation-1:~#
```

- After Host resolution process now making ansible configuration which includes task and parent directory which has hosts list in addition to this ansible.cfg file contain ssh connection timeout.



```
root@NetworkAutomation-1:~# cat ansible.cfg
[defaults]
hostfile = ./hosts
host_key_checking = false
timeout = 5
```

- Now checking ansible inventory file which includes all the hosts connected to ansible server.

```
root@NetworkAutomation-1:~# ansible --list-hosts all
[DEPRECATION WARNING]: [defaults]hostfile option, The key is misleading as it
can also be a list of hosts, a directory or a list of paths . This feature will
be removed in version 2.8. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
hosts (3):
  R1
  PC-1
  PC-2
```

- Now system ready to get information from its hosts machine.
- But we are facing SSH connection error which include system file :  
/root/.ssh/ssh\_config. And we try to contact the administrator of GNS3 but they are working on this issue so as soon as this errors solve we are able to run ansible modules because all the background process and configuration is up to date.

## 7. Overview of OPNET

OPNET stands for Optimized Network Engineering Tools which is a network simulation tool used for monitoring and management of performance using various networking devices. It helps simulate and analyze the behavior with the help of inbuilt models of various networking devices. OPNET is an open source tool which assists modelling of various protocols along with traffic generation and graphical depiction of the outputs

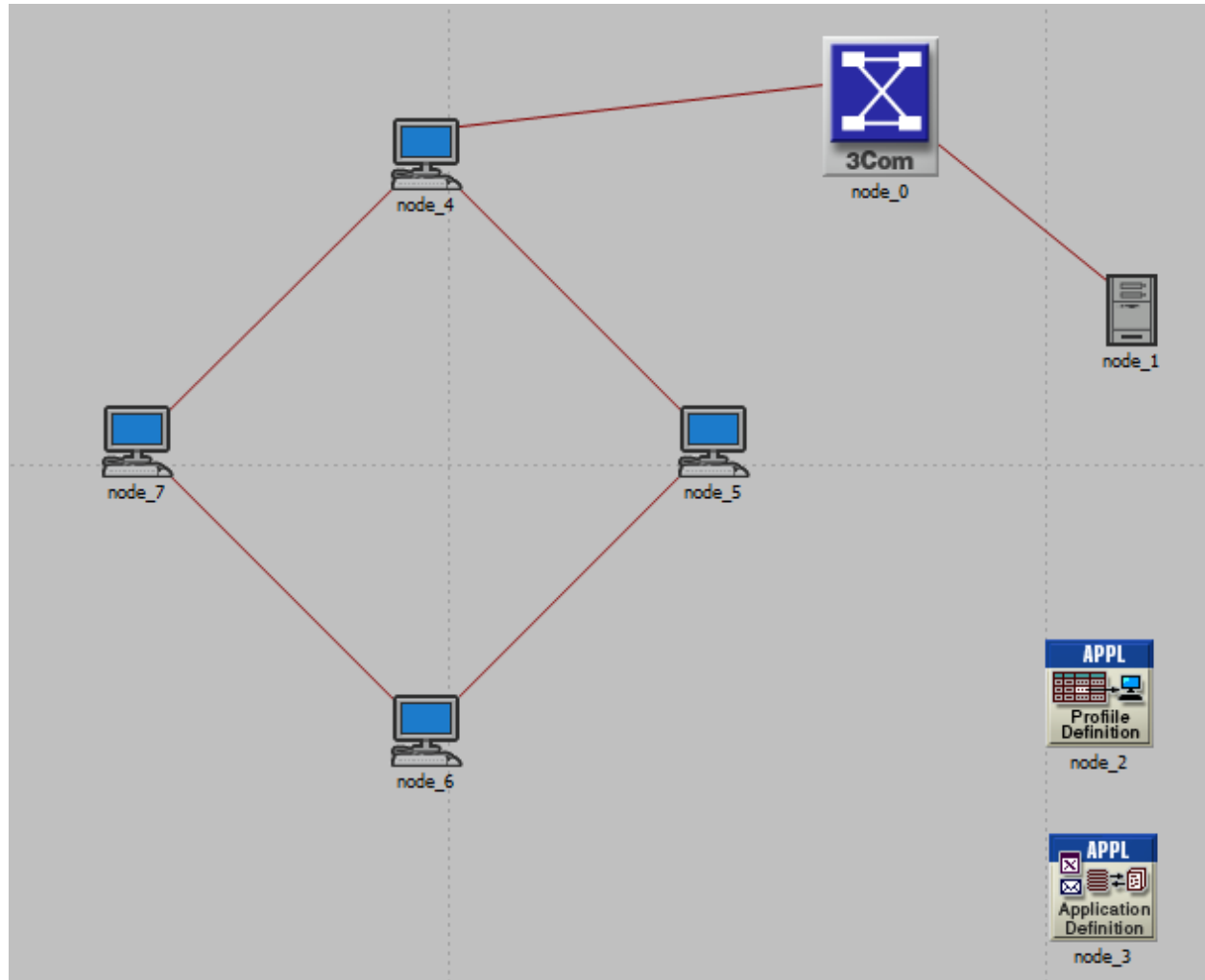
### Features of OPNET

- Since OPNET doesn't act as an emulator, so it performs quicker simulations and no issues with respect to lag and latency arise
- User-friendly due to ease of access to various drag and drop network devices on its graphical interface
- Minimal to no expense to download the tool or any router images due to its open source nature and facility to access various inbuilt routers of varied series
- Minimal hardware and CPU utilization as it doesn't need any virtual image imitation of physical devices which cuts down the tasks of manual configurations
- Various service metrics such as Ethernet Delay, Bandwidth, Utilization, Queuing Delay, Traffic sent and Received, Network Load, Client-server packet filtering
- Compatible with various Virtual Platforms and development tools

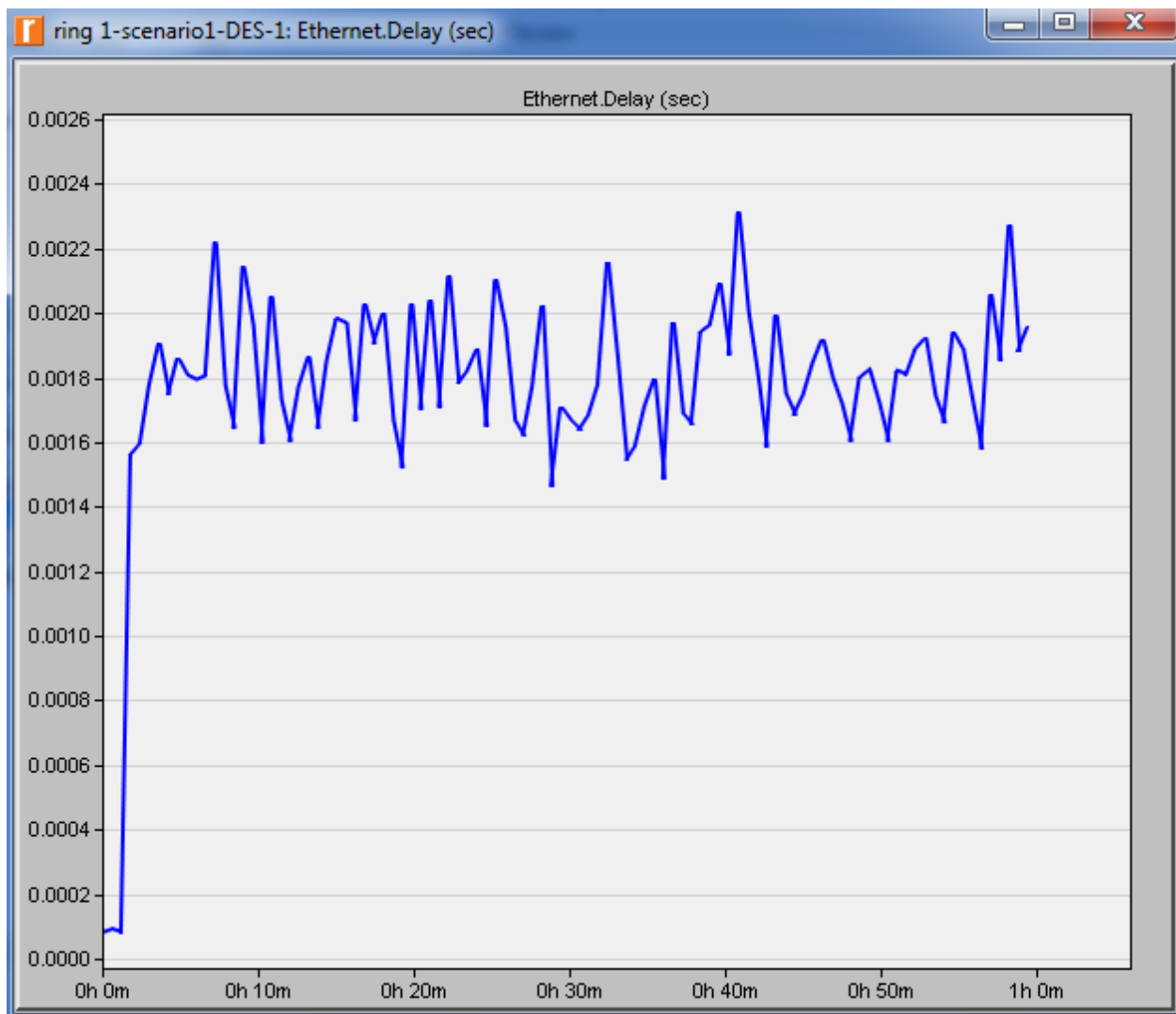


## 8. Performance Analysis and Results In OPNET

### 8.1. Ring Topology

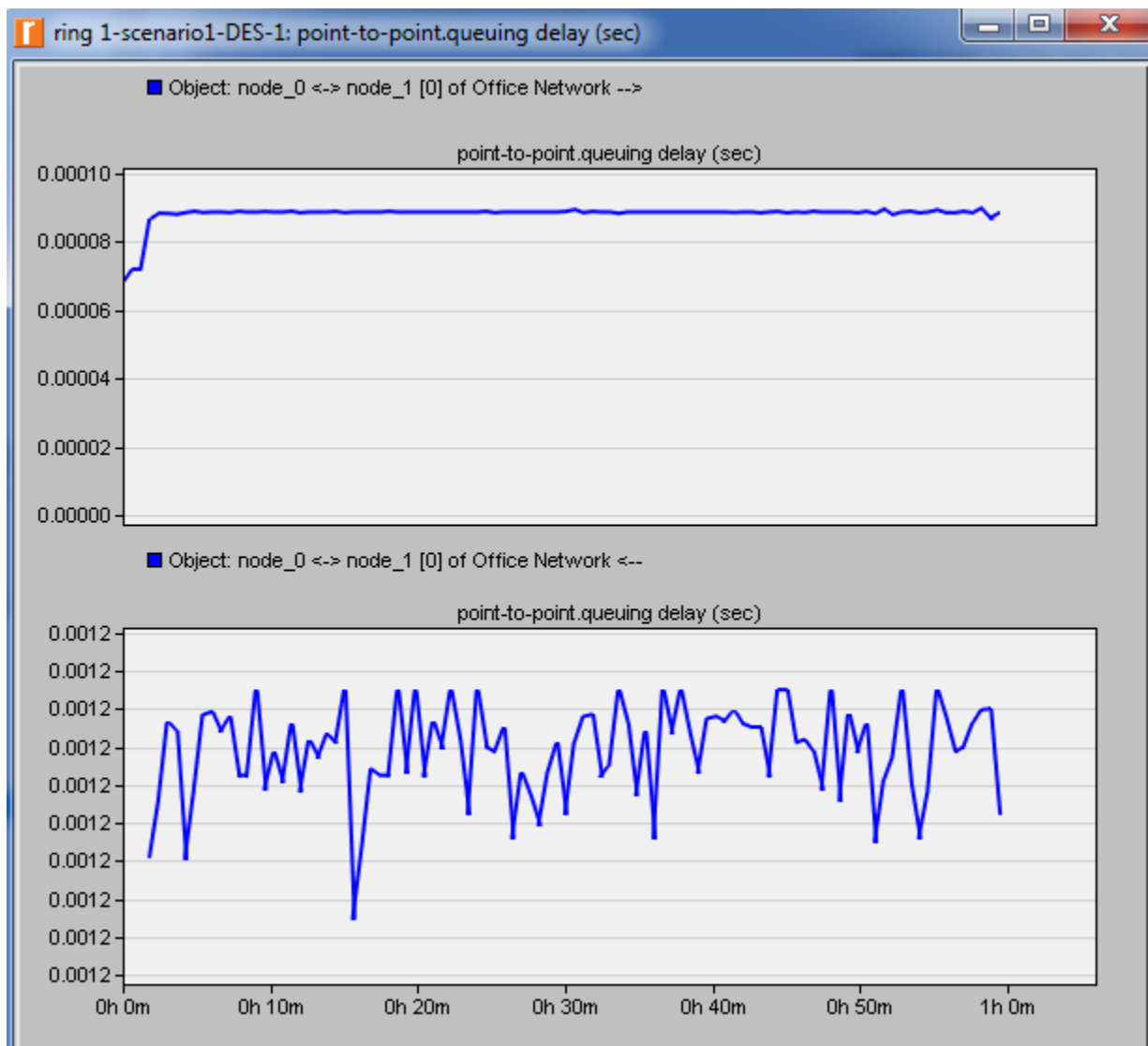


## Delay



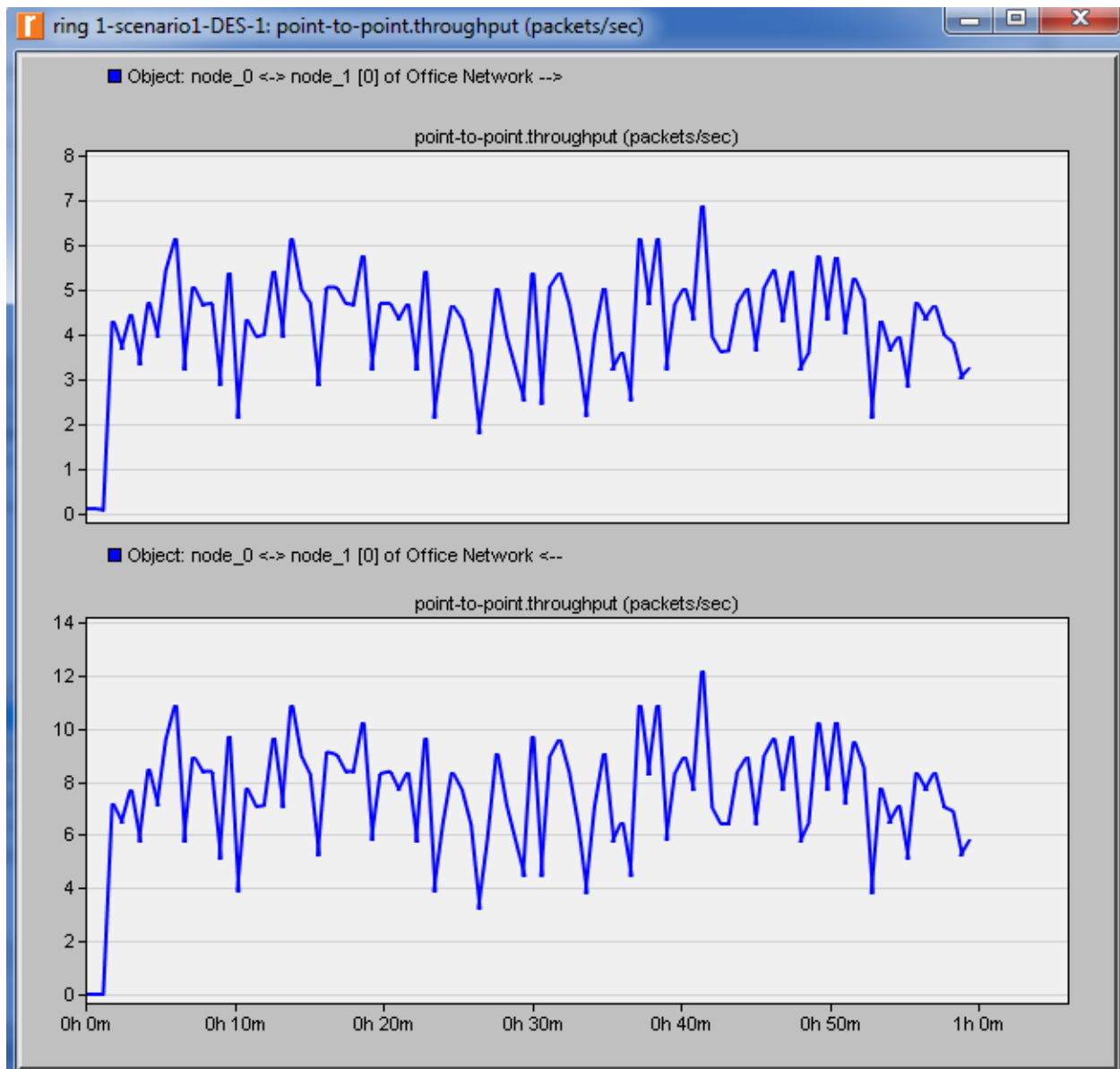
Here, ethernet delay varies with time. For an observation of one hour, the aggregated value of ethernet delay is approximately 0.0019 seconds.

## Queuing Delay:



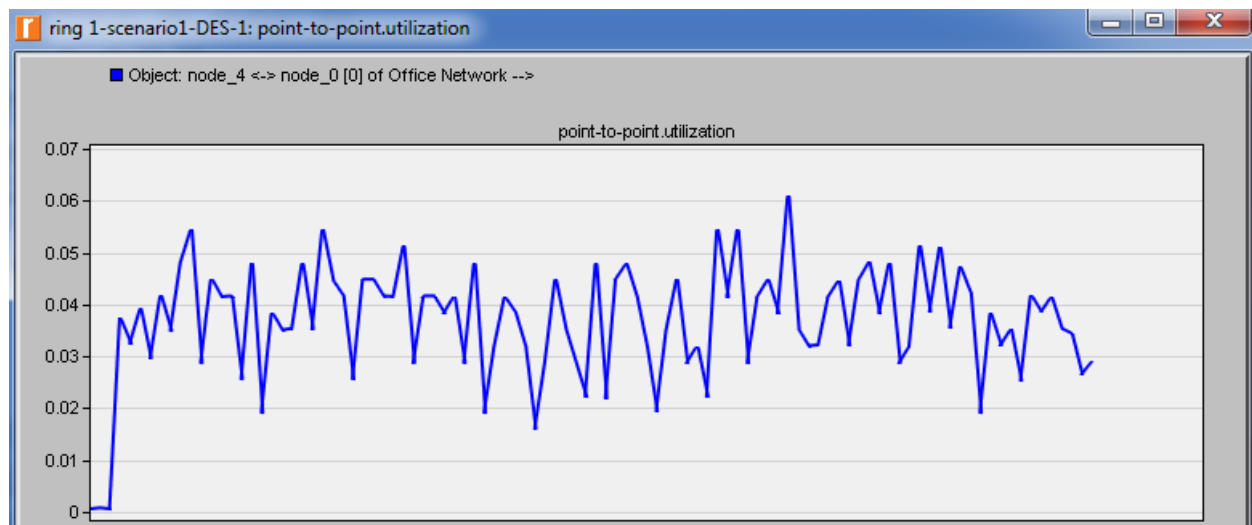
Here, queuing delay is fluctuating and value of it is approximately 0.00009 seconds for packets queued while traversing from host to server and 0.0012 seconds for packets traversing from server to host

## Throughput



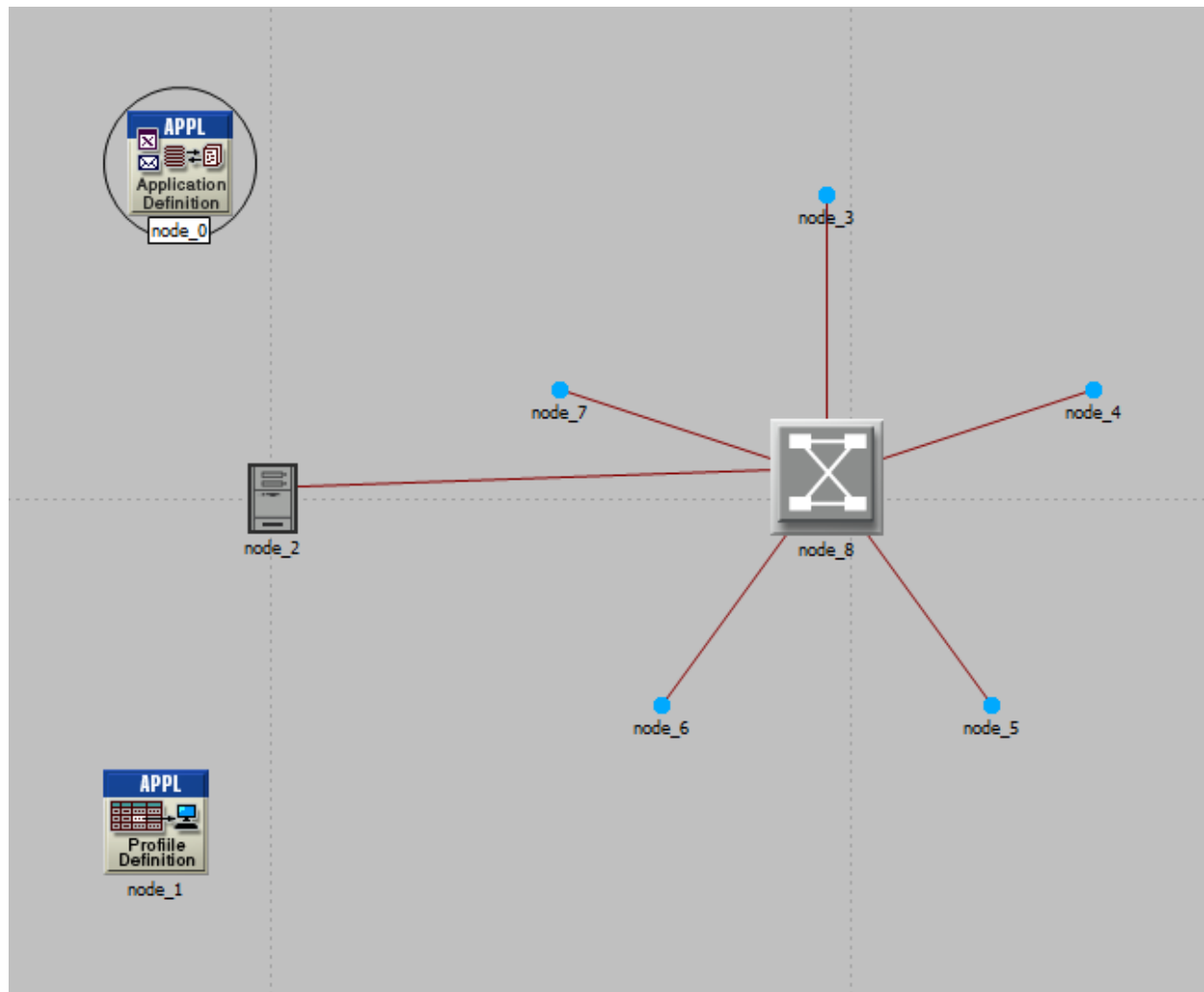
Here, the throughput for ring topology varies with respect to time. For an observation of one hour, the aggregate value of throughput is 7 packets/sec.

## Utilization:



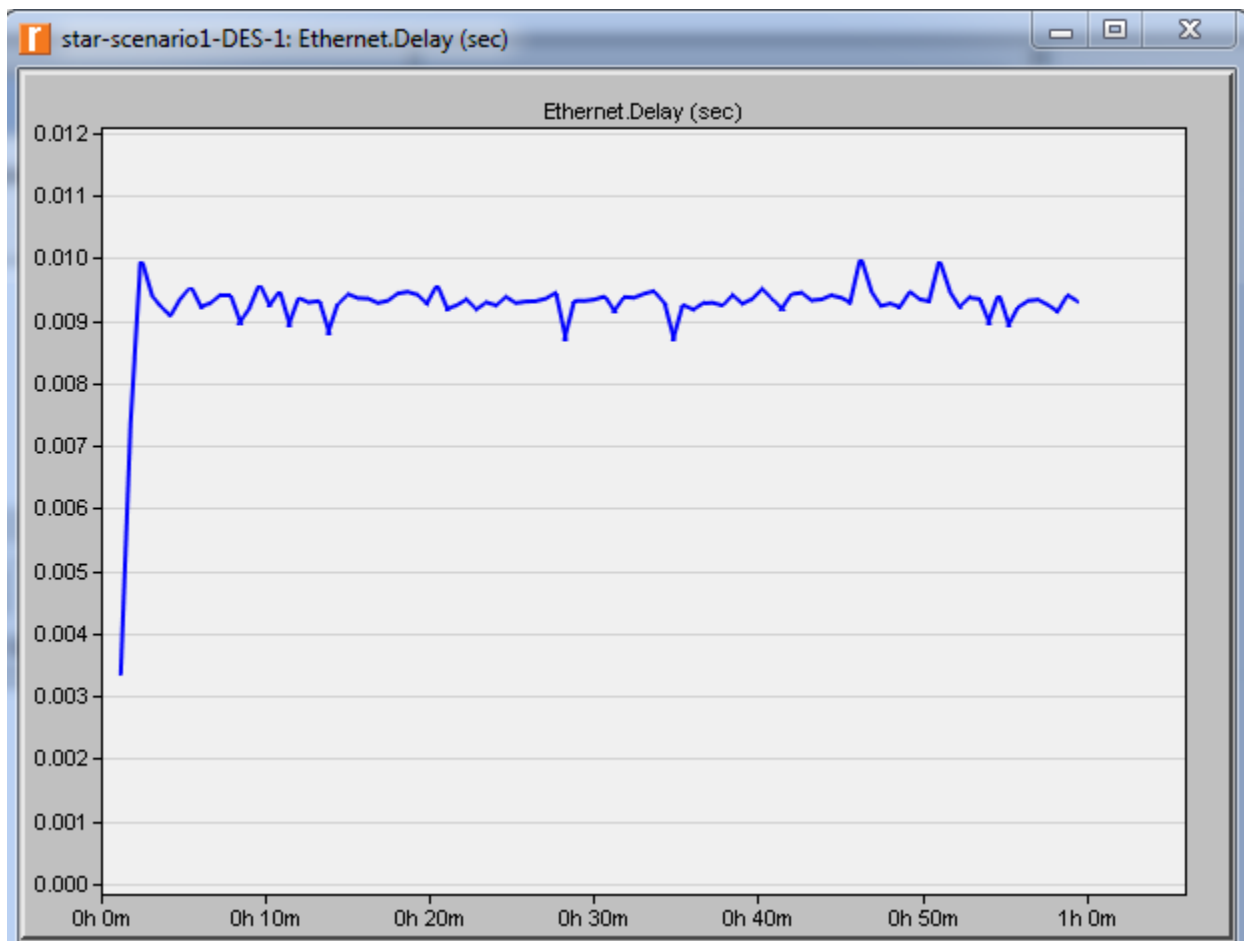
Here, the average utilization of the network measured on switch is about 4.5 %

## 8.2. Star Topology



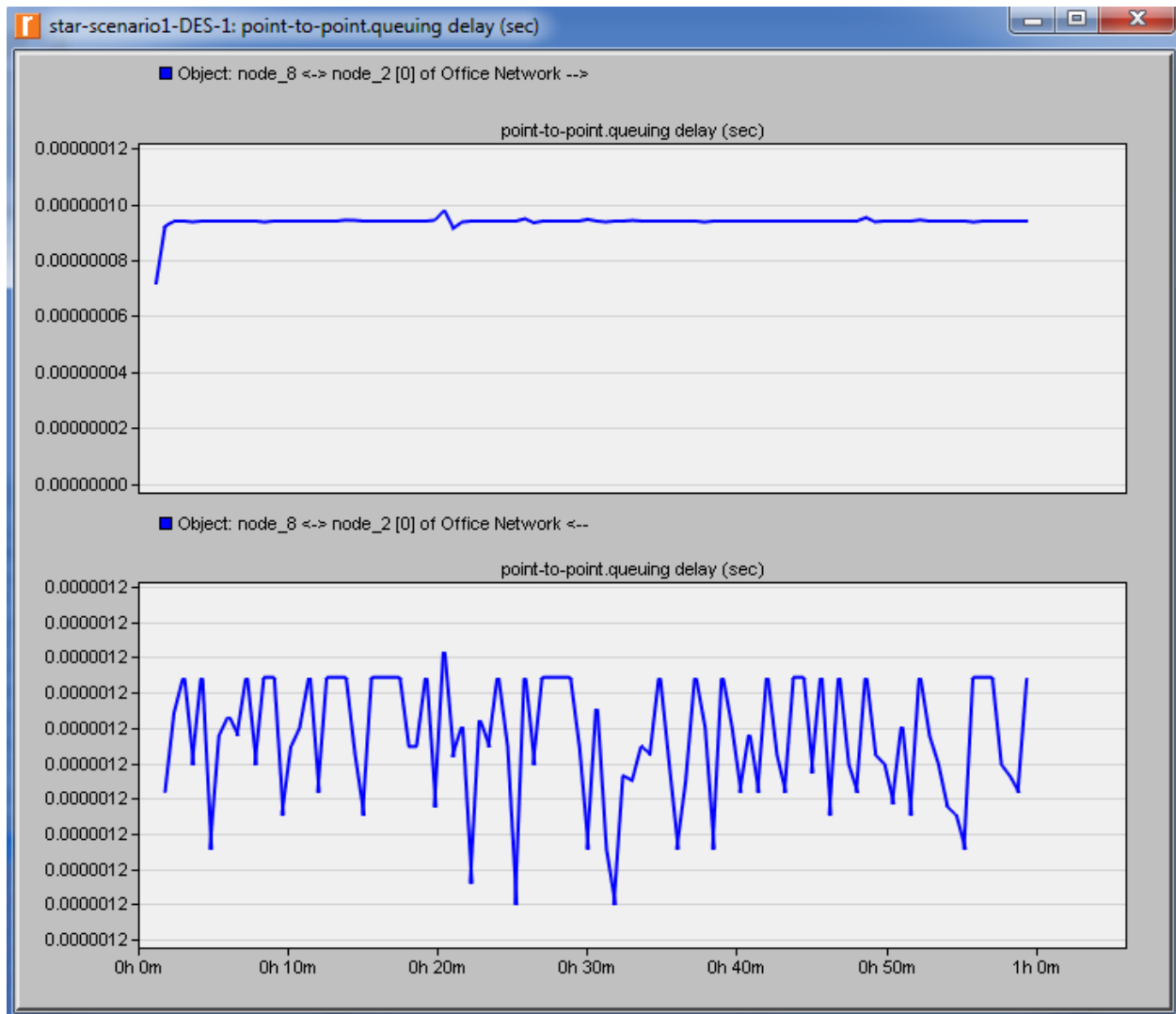


## Delay



Here, ethernet delay varies with time. For an observation of one hour, the aggregated value of ethernet delay is approximately 0.0095 seconds.

## Queuing Delay:



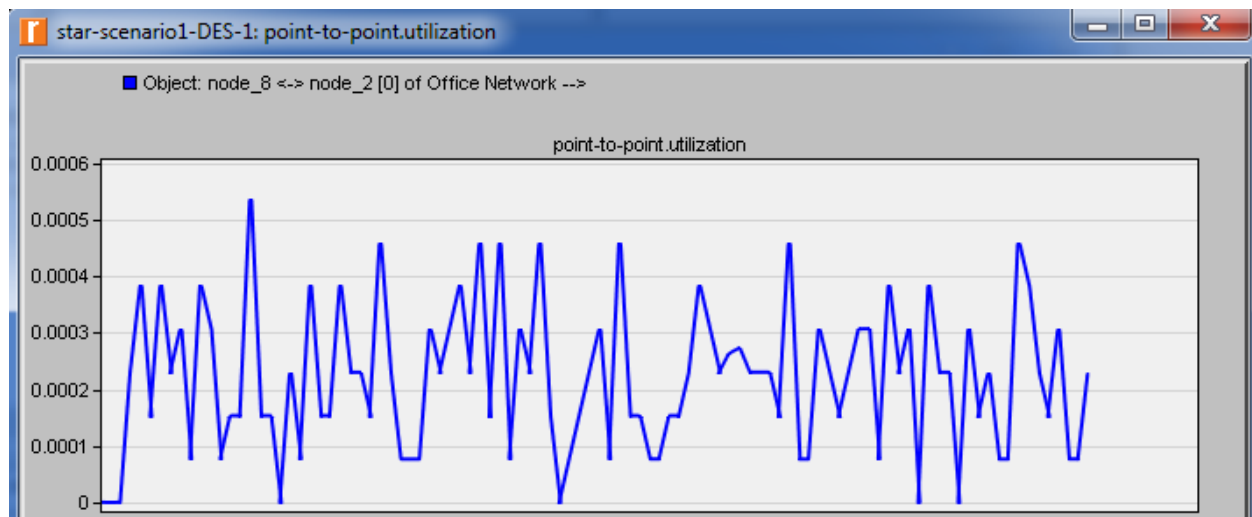
Here, queuing delay is fluctuating and value of it is approximately 0.00000009 seconds for packets queued while traversing from host to server and 0.00000012 seconds for packets traversing from server to host

## Throughput



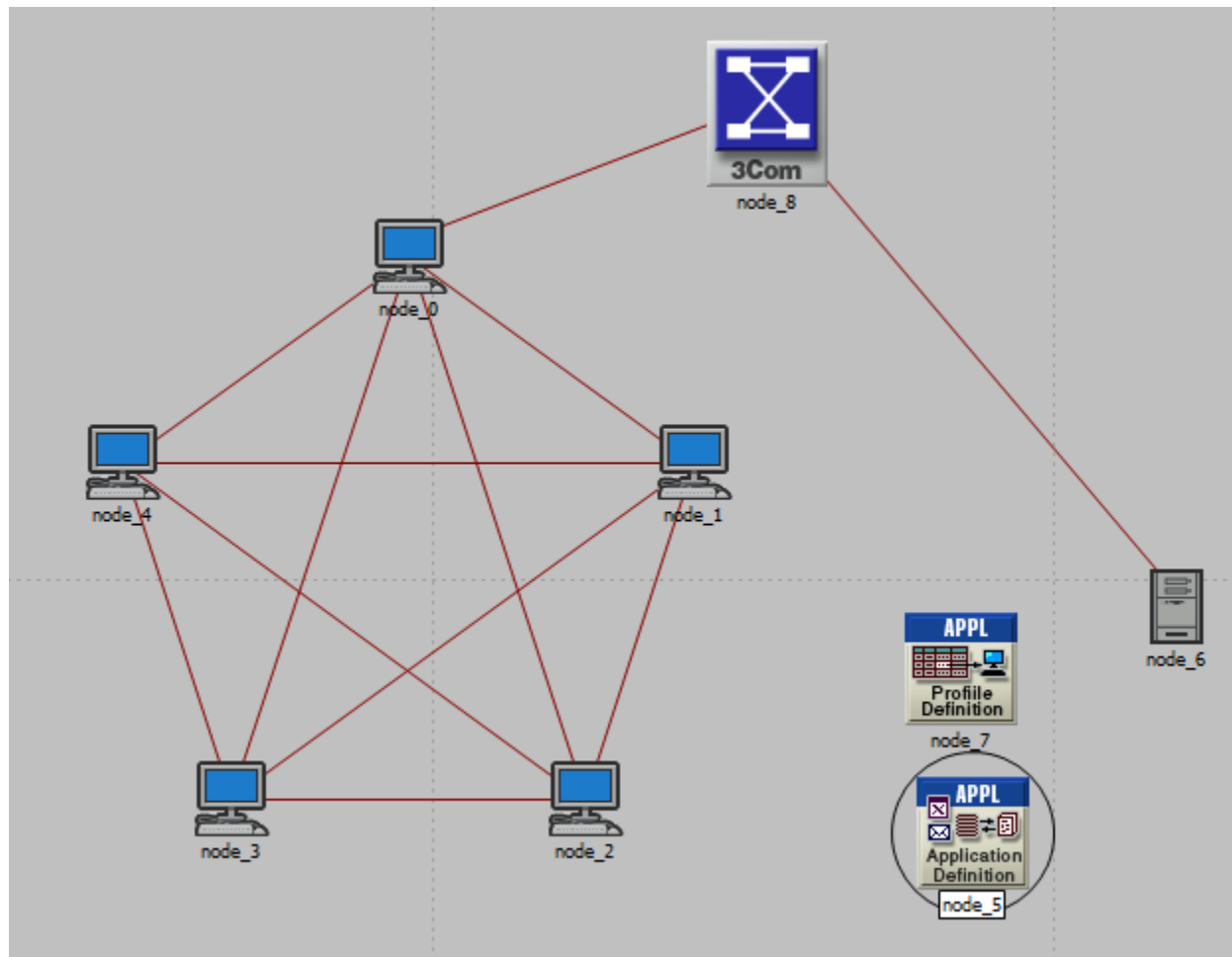
Here, the throughput for star topology varies with respect to time. For an observation of one hour, the aggregate value of throughput is 1250 bits/sec and the packet size is 63.617 bytes which gives us the total packets of  $2.4 \approx 3$  packets/second

## Utilization

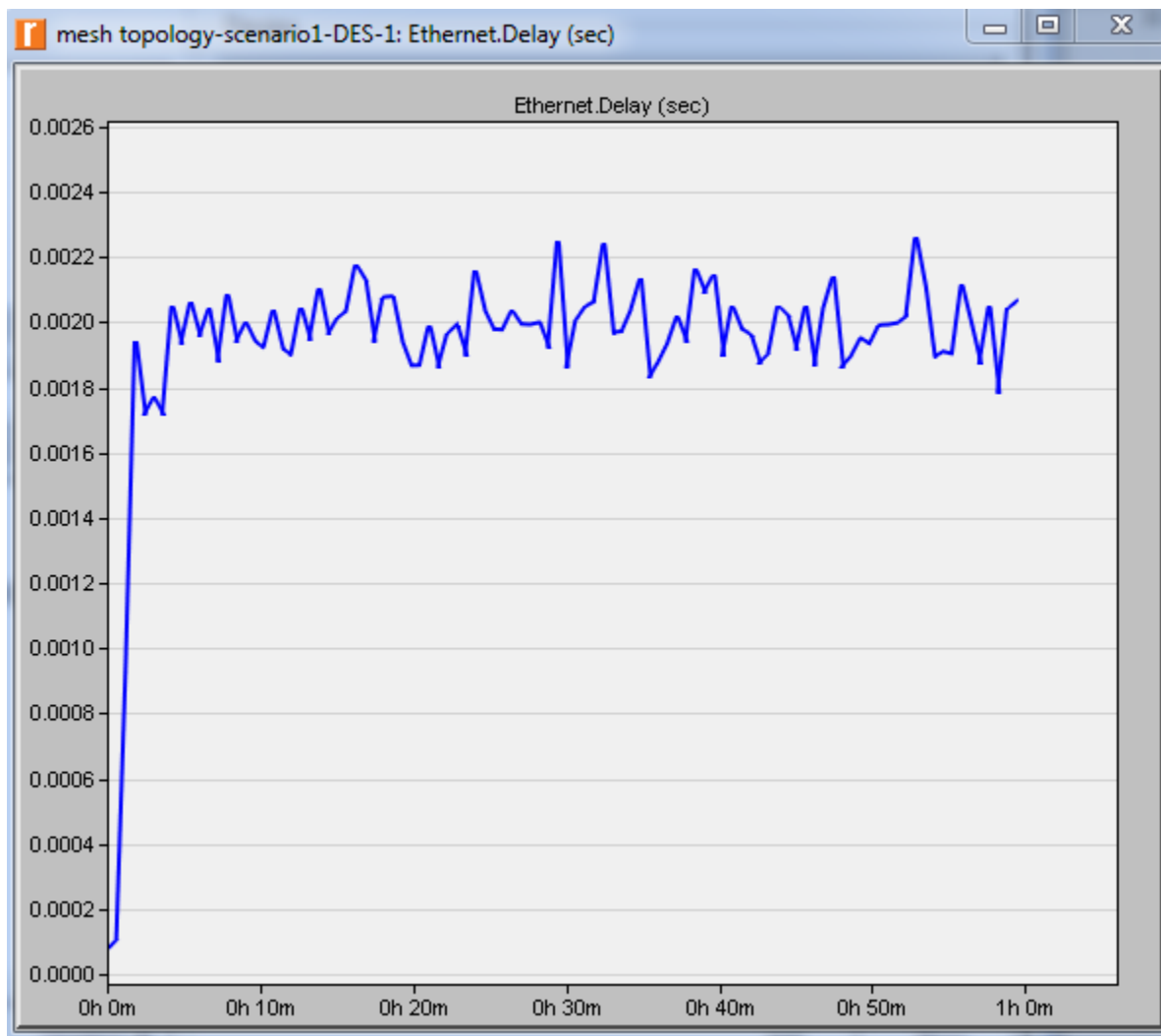


Here, the average utilization of the network measured on switch is about 3 %

### 8.3. Mesh Topology in OPNET

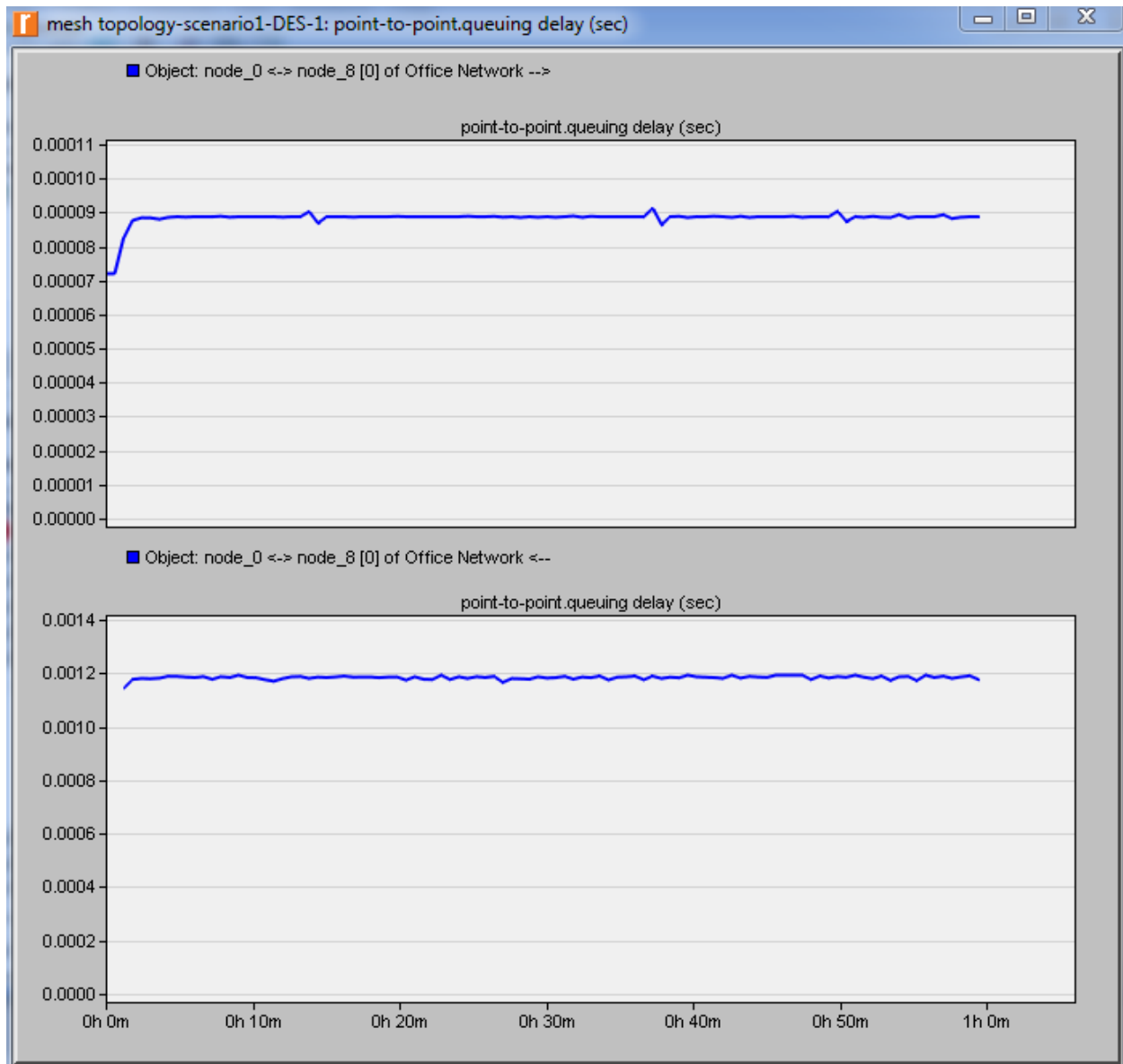


## Ethernet Delay



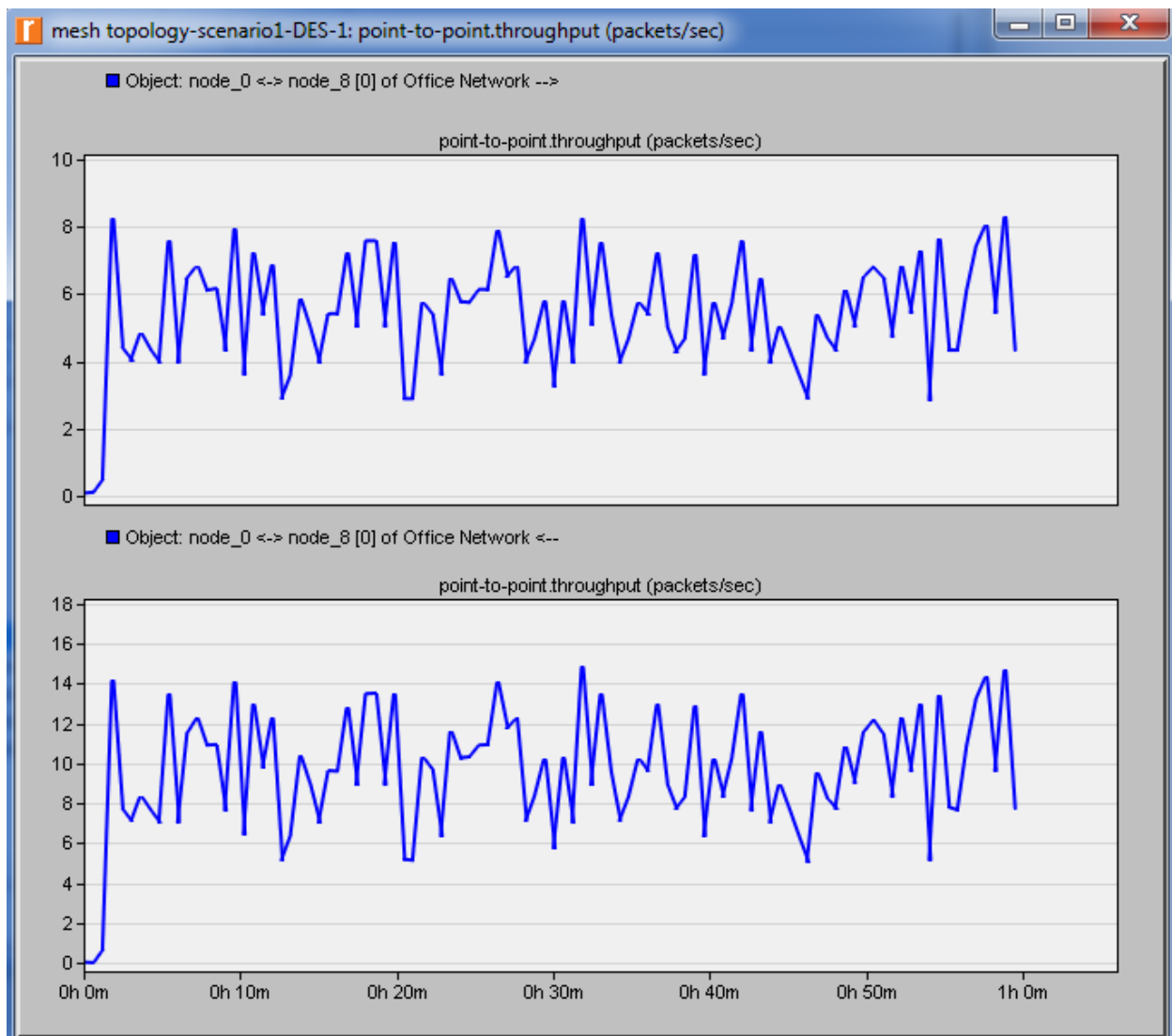
Here, ethernet delay varies with time. For an observation of one hour, the aggregated value of ethernet delay is approximately 0.002 seconds.

## Queuing Delay



Here, queuing delay is fluctuating and value of it is approximately 0.00009 seconds for packets queued while traversing from host to server and 0.0012 seconds for packets traversing from server to host

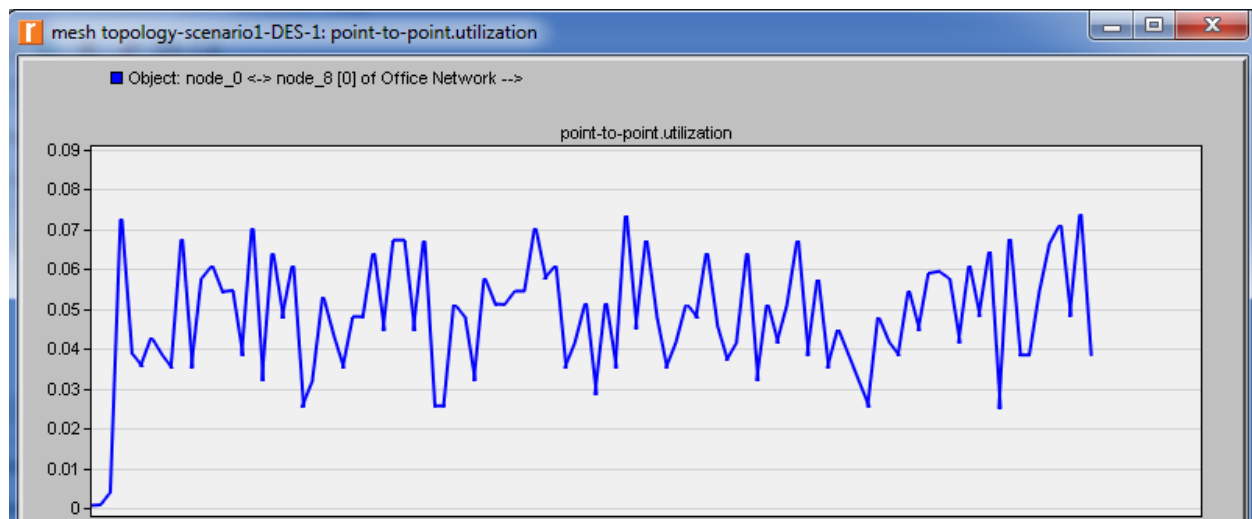
## Throughput



Here, the throughput for mesh topology varies with respect to time. For an observation of one hour, the aggregate value of throughput is 6 packets/sec.



## Utilization



Here, the average utilization of the network measured on switch is about 6 %

## 9. Comparison of Topologies based on OPNET Results

	Delay (seconds)	Queuing Delay (Host-Server) in seconds	Queuing Delay (Server-Host) in seconds	Throughput (pps)
<b>Ring</b>	0.0019	0.00009	0.0012	9
<b>Star</b>	0.0095	0.0000009	0.0000012	3
<b>Mesh</b>	0.002	0.00009	0.0012	6

By analyzing the results obtained using OPNET simulation for various topologies such as Ring, Star and Mesh. Similar to GNS3, we can observe the delay and throughput for all three topologies

In terms of Delay, **Ring** topology is faster and better than Star and Mesh Topology

In terms of throughput, **Ring** Topology is again better than Star and Mesh Topology

### Performance Evaluation

Topology	1 (Best)	2 (Intermediate)	3 (Worst)
<b>Delay</b>	Ring	Mesh	Star
<b>Throughput</b>	Ring	Mesh	Star
<b>Cost</b>	Ring	Star	Mesh

From above result table we can discard star topology as it is beaten by mesh in two factors i.e. delay and throughput. When delay is the criteria, we will select Mesh topology. But, when throughput and cost is the criteria, we will go with Ring Topology.

## 10. Conclusion

After examining two softwares i.e. GNS3 and OPNET, we found out that **OPNET is more user friendly and efficient than GNS3** so it would manifest to be a **better recommendation** as a network simulation tool. GNS3 is an emulator while OPNET is simulator. GNS3 lacks on several fronts like inability to produce performance characteristics like delay, throughput and utilization etc. in graphical format. Also, GNS3 needs user to purchase virtual image of networking devices which are expensive. The GNS3 works on VMWARE when we exploit its functionality (network automation, containerization etc.), it makes the software slower by adding latency during booting process. On contrary, OPNET can process data rapidly and shows performance metrics in graphical format. But, OPNET is not able to incorporate trending technologies like network automation with ansible and docker. Also, OPNET not being an emulator it can't work as real networking device.

Ansible tool is a great tool for network automation which runs on GNS3. This helps to have complete activity log of network. Single activity is reported on Network Automation server. Ansible is predominantly used for network migration and automation. The playbook of ansible has task scripts which can be called to run a function. These all tasks playbooks are stored in Ansible Server. By pinging from server to local host, we can establish connection between the two, and the server is able to have a view of entire network due to network automation by Ansible.

By analyzing three topologies in GNS3 and OPNET, we came to conclusion that Ring topology is best among all. In GNS3, ring topology is superior to mesh and star in terms of throughput and cost. The results of OPNET shows ring topology ahead of others. From industry point of view, ring topology will become winner because of cost of infrastructure is very low.

## 11. Bibliography

1. <https://www.igi-global.com/dictionary/opnet-it-guru/21330>
2. RFP's on My Courses by Prof. Nygate
3. <tps://academy.gns3.com/p/ansible-for-network-engineers-gns3-ansible-cisco-network-automation>
4. <https://www.udemy.com/ansible-for-network-engineers-cisco-quick-start-gns3-ansible/learn/v4/overview>
5. <https://www.gns3.com/marketplace>
6. <https://networklore.com/ansible/>