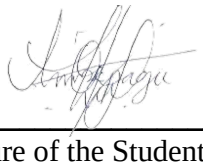
Individual Project

Evaluation and Analysis of the Routing Protocols IS-IS and OpenFabric using CORE

Submitted by:	Arinze Okechukwu Okpagu
Matriculation number:	1324112
First examiner:	Prof. Armin Lehman
Second examiner:	-
Date of start:	November 7, 2023
Date of submission:	April 11, 2023

Statement

I confirm that I have written this thesis on my own. No other sources were used except those referenced. Content which is taken literally or analogously from published or unpublished sources is identified as such. The drawings or figures of this work have been created by myself or are provided with an appropriate reference. This work has not been submitted in the same or similar form or to any other examination board.

April 11, 2023 

Date, Signature of the Student

Content

1 Introduction

Routing protocols are used by routers to determine the best paths for data to take across a network. They are responsible for exchanging information between routers, such as network topology, link costs, and neighbor availability. Depending on the type of protocol, the data exchanged can be used to build routing tables, to find the best path for a packet to take, and to update the routing tables when the network topology changes. Routing protocols help maintain the integrity of the network by creating efficient and reliable communication paths between network devices.

The most commonly used routing protocols are the Interior Gateway Protocols (IGPs) and the Exterior Gateway Protocols (EGPs). The IGPs are used within an autonomous system and include RIP, OSPF, EIGRP, IS-IS and Openfabric. The EGPs are used for communication between autonomous systems and include BGP. Each of the routing protocol can be further segmented depending on the algorithm it uses to select the best path to a destination. These include Distance vector protocols, link state protocols and path vector protocols.

Distance vector protocols are based on the Bellman-Ford Algorithm and use a distance vector (or hop count) to determine the shortest path path to a destination. These protocols exchange information with their directly-connected neighbors, including the cost (or distance) to each destination. Here, each router maintains a routing table that contains the distance (in hops) to all destinations in the network. Link state protocols are based on the Dijkstra Algorithm, also known as the Shortest Path First algorithm, and use a link state (or topology) database to determine the best path to a destination. In link-state routing protocols, each router maintains a database of the entire network topology, which includes information about the status and cost of each link in the network. Each router then uses this information to calculate the shortest path to all other routers in the network using the SPF (Shortest Path First) algorithm. Path vector protocols calculate the path to a destination based on a set of policies that are defined by network administrators. These policies may include factors such as cost, performance, and security.

Both Intermediate System-to-Intermediate System (IS-IS) and Openfabric routing protocols are link-state routing protocols. They differ from distance vector and path vector protocols in that they maintain a complete view of the network topology and use this information to calculate the shortest path to a destination network, taking into account factors such as link bandwidth, delay, and cost. This approach enables more efficient path selection, faster convergence times and allows for rapid adjustment to changes in the network structure such as link failures or network congestion. They are typically deployed in large-scale, complex networks such as Service Provider networks, Enterprise networks, and data center networks.

The objective of this document is to evaluate and analyze the IS-IS and OpenFabric routing protocols. The document covers the theoretical foundations of both protocols, compares their respective strengths and weaknesses, and provides practical demonstrations of the protocols in action using tools such as Common Research Network Emulator (CORE) and FRRouting (FRR). Additionally, the document highlights the use of FRR, an open source Internet routing protocol suite for Linux.

2 Theoretical Background

Routing protocols are a set of rules and algorithms that determine how data is transmitted between network devices. These protocols enable routers and switches to dynamically share network topology and routing information and determine the best path for data to travel across a network.

2.1 Intermediate System-to-Intermediate System (IS-IS)

The IS-IS protocol was originally developed by a team of people at Digital Equipment Corporation as part of DECnet Phase V in 1985 as DECnet Phase V Routing. In 1988, it was adopted by the International Organization for Standardization and renamed IS-IS (Intermediate System-to-Intermediate System). Currently, IS-IS is an international standard within the Open Systems Interconnection (OSI) reference model, defined in ISO/IEC 10589:2002 (ISO, 2002). It was originally designed to work with the connectionless-mode network service (ISO 8473) and the End system to Intermediate system routing exchange protocol. However, it has been extended to support IP by RFC 1195. Other RFCs such as RFC 5308, RFC 5301, RFC 6232, RFC 8667 and RFC 9377 to mention but a few, have contributed additional enhancements and guidelines for the protocol.

IS-IS, as an interior gateway protocol, is designed for use within an administrative domain. IS-IS is a link-state routing protocol, such that each router maintains a database of the entire network topology, including information about the status and cost of each link in the network. This information is shared between routers using link-state advertisements (LSAs) or link-state packets (LSPs). Each router then uses this information to calculate the shortest path to all other routers in the network using the SPF algorithm, also referred as Dijkstra's algorithm. The Dijkstra's algorithm finds a shortest path tree from a single source node, by building a set of nodes that have minimum distance from the source. It is used both for routing in IP networks (such as Internet Protocol or IPv4 and IPv6) and for routing in CLNS (Connectionless Network Service) networks, which are used for the OSI protocol.

ISO/IEC 10589:2002 standard defines the format of the packets used for routing information exchange in IS-IS, the algorithms used to compute shortest paths, and the rules for distributing and updating routing information. It also specifies the mechanisms for managing and maintaining the routing information database, and the rules for handling network topologies and addressing. The standard also defines the mechanisms for authentication and security, which are important for ensuring the integrity of the routing information exchange.

IS-IS has been deployed successfully in service provider, data center and enterprise networks, particularly in large-scale and complex networks because it provides speed, efficient routing, high scalability, fast convergence, hierarchical routing and support for multi-protocol environments.

2.1.1 IS-IS Network

An IS-IS network is a single autonomous system (AS), also called a routing domain, that consists of end systems and intermediate systems.

- End Systems (ES): End systems refer to the devices that are connected to the network and communicate with each other through the intermediate systems. ESs can be hosts, servers, routers, or any other devices that generate or consume network traffic. ES devices do not participate in the routing of network traffic.
- Intermediate System (IS): Intermediate system is the OSI term for a router. ISs are actively involved in the routing process. They particularly relay network traffic between end systems. Within IS-IS, a single autonomous system (AS) can be partitioned into smaller groups known as areas. An area is a group of routers and networks that share the same topology information. Communication between the areas is arranged in a hierarchical manner, where a domain can be split into smaller administrative areas to simplify network management and reduce the amount of information that must be flooded throughout the entire network. Intermediate systems are subdivided into levels to cater for these administrative areas as follows:

    - Level 1 ISs: These systems deliver and receive network traffic from other systems, and relay network traffic from other source systems to other destination systems. These systems are intra-area systems and route directly to other Level 1 ISs within their own area. In a case where the destination system

is located in a different area, Level 1 IS would route the traffic towards a level 2 IS within its own area. Level 1 IS creates a level 1 link state database and SPF tree specific for the area.

- Level 2 ISs: These systems could be viewed as inter-area systems. Systems in the level 2 subdomain route towards a destination area, or another ASs. A level 2 IS creates a level 2 link-state database and SPF tree based on its adjacencies. Level 2 IS do not need to know the topology within any level 1 area, except to the extent that a level 2 router may also be a level 1 router within a single area.
- Level 1-2 ISs: These systems can act as both Level 1 and Level 2 routers, sharing intra-area routes with other Level 1 ISs and interarea routes with other Level 2 ISs. A Level 1-2 IS creates a separate level 1 and 2 link-state database and two SPF trees, one for each database.

2.1.2 IS-IS Terminology

Some terminologies associated with IS-IS include:

- Area: A group of ISs that are interconnected within a single AS and share a common set of routing protocols.
- Domain: A collection of contiguous areas, which are interconnected through a backbone area.
- Protocol Data Unit (PDU): ISO packets are called network protocol data units (PDUs). This refers to the basic unit of exchange between entities that communicate using a specified networking protocol. IS-IS uses several types of PDUs to exchange information between routers and maintain data about the network.
- Hello Packet (IIH PDU): This is used to establish and maintain neighbor relationship between ISs
- Adjacencies: These are the relationships between directly connected IS-IS routers. Adjacencies are formed using the Hello PDU and are necessary for exchanging routing information.
- Level: A single IS-IS domain can be divided into multiple levels, where each level includes a subset of the routers in the domain.
- Link-state PDUs (LSPs): These are the messages used by IS-IS routers to exchange information about the network topology. LSPs contain information such as the router's system ID, the interfaces on the router, and the state of those interfaces.
- LSP flooding: This is the process by which LSPs are distributed throughout the network. Each router floods its own LSP to its neighbors, who in turn flood it to their neighbors, until all routers in the network have received the LSP.
- Sequence number packets (SNPs): These form a vital component in the IS-IS routing protocol, as they help to manage the distribution and synchronization of link-state packets across routers within a particular routing area. By using sequence numbers, routers can ensure that the most current information is being shared among all devices, which is essential for accurate and efficient routing. Essentially, SNPs help to maintain a consistent and up-to-date view of the network topology among all routers in the IS-IS network.
- Routing Information Base (RIB): The database that contains the routing information for all destinations within an IS-IS domain. Each router in an IS-IS network maintains a local RIB, which contains information about the network topology and routing information. The RIB comprises of the link-state database and the forwarding database.
- Topology Dissemination: The process used by IS-IS to distribute information about the network topology to all routers within an area or domain. This information is communicated using Link-State Packets (LSPs).
- Metric: This is a value that is associated with each link or routing path in the network, which is used to determine the best path for forwarding data packets. The metric can be based on a variety of factors, such as the cost of the link, the bandwidth available, or the delay or latency of the link.
- Dijkstra's Shortest Path First (SPF) Algorithm: After receiving link state information from other routers, each router runs the Dijkstra's SPF algorithm to calculate the shortest path to every other node in the network. The result of the SPF algorithm is used to populate the router's RIB.

2.1.3 IS-IS Packets

IS-IS packets have three categories:

- Hello Packets
- Link-state Packets
- Sequence Number Packets

Based on Figure 1, the different categories are divided into smaller groups that can be recognized within the network by their PDU type number.

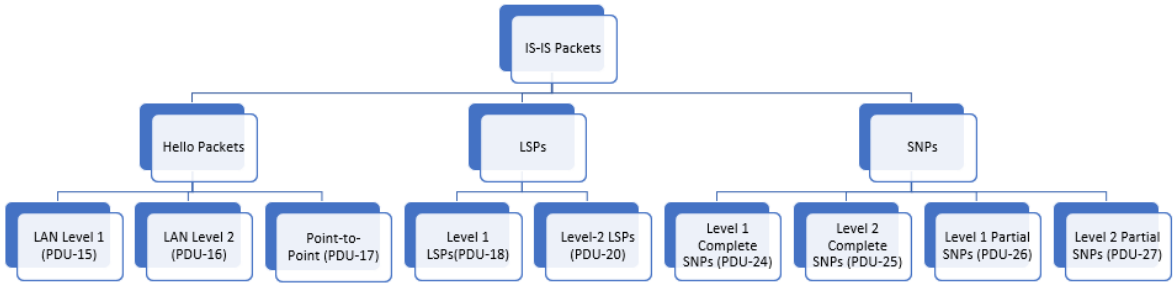


Figure 1: IS-IS Packet Types

IS-IS packets consist of a header and several optional variable-length fields with a type label that explains the content. The content is made up of multiple repeated blocks with a specific length specified in a 1-byte length field. The type, length, and value of each variable-length field form a tuple (TLV). Numeric code values specify the types of variable-length fields.

Although the composition of header fields slightly varies for different IS-IS packet types, the first eight fields of each type are one byte in length.

Figure 2 shows the generic packet format and the common fields shared by all IS-IS packets.

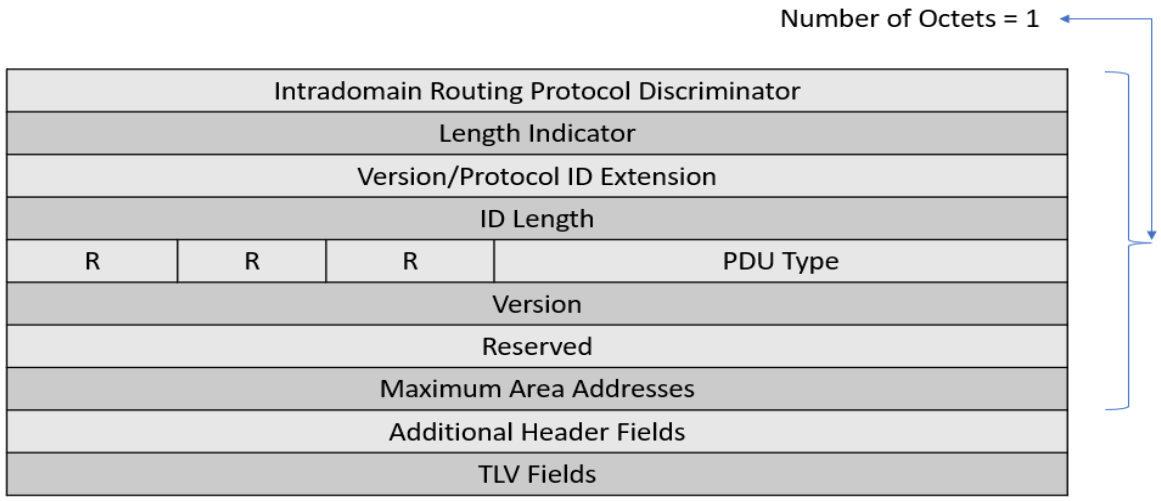


Figure 2: IS-IS Packet Format

The TLV fields are appended to the header to make up the entire packet.

- Length Indicator: This indicates length of the packet header fields in octets. This field specifies the length of the fixed part of the packet header excluding the variable-length fields.
- Version: This field is used to indicate the version number of the IS-IS protocol being used. Currently, the version number for IS-IS is 1.
- ID Length: This is a 1-byte field that indicates the length of the source ID field, also known as the System ID (SysID) field, in bytes. The value of the ID length field can be between 0 and 255. For values between 1 and 8, the ID length field indicates the actual length of the SysID field in bytes. This allows for flexibility in the length of the SysID field, depending on the specific implementation of IS-IS being used.
- PDU Type: Specifies the type of IS-IS packet.
- Version: The value is 1.
- Reserved: Unused bits. Set to 0s.

- Maximum Area Addresses: Values are between 1 and 254. A value of 0 implies a maximum of three addresses per area.
- TLV: Type, T implies the number code for a specific TLV. The length, L specifies the total length of the TLV field. Value, V indicates the content of the TLV

2.1.4 IS-IS Network Address Structure

The IS-IS hierarchical address structure enables the efficient routing of data in large and complex networks by allowing them to be partitioned into smaller, more manageable subnetworks, each with its own area or domain. It provides a method of organizing and partitioning network addresses. This reduces the amount of routing information that needs to be exchanged between routers and improves overall network performance.

The address used for routing is called the Network Entity Title (NET). NETs can have different forms based on network requirements, and they are represented in hexadecimal format ranging from 8 to 20 octets in length. The format generally includes an Authority and Format Identifier (AFI), a Domain ID, an Area ID, a System Identifier, and a Selector. The NET can also be viewed as having two parts: the Initial Domain Part (IDP) and the Domain Specific Part (DSP).

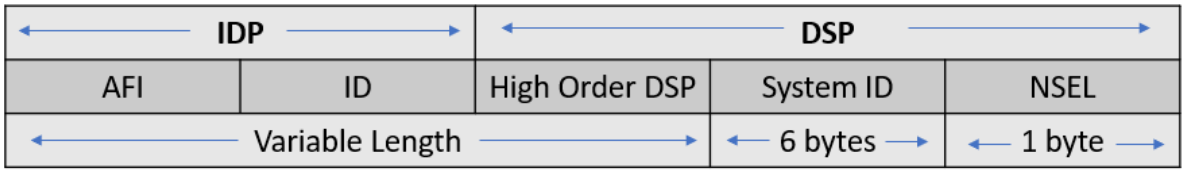


Figure 3: IS-IS Address Format

The IDP is standardized by the ISO and specifies the format and authority responsible for assigning the rest of the address. The DSP is assigned by the addressing authority specified by the IDP. The DSP is further divided into a High Order DSP (HO-DSP), a system identifier (ID), and a Network Service Access Point (NSAP) selector. The HO-DSP may use any format desired by the authority identified by the IDP. Together, the combination of IDP and HO-DSP identify both the routing domain and the area within the routing domain. This combination is called the Area Address. The System ID is a unique identifier assigned to each IS in the network. no two IS in an area may use the same ID value. The area address is used for intra-area routing, while the system ID is used for inter-area routing.

The NSAP Selector is used to identify the network service access point (NSAP) for a particular router. To adhere to the addressing regulations of IS-IS, it is not allowed for any two systems in an area to have identical addresses in all fields, except for the SEL field. This means that the NETs assigned to ISs within the same area must have distinctive addresses in all fields except for the N-selector field. Additionally, all systems that belong to the same routing domain are required to have NETs or NSAP addresses with ID fields that are of the same length. The diagram in Figure 4 further describes the format of a NET address for instance; 49.0001.1921.6800.1001.00, used to identify a system in an IS-IS network.

	ID + HO-DSP		
AFI	Area ID	System ID	NSEL
49	0001	1921.6800.1001	00

Figure 4: NET Address instance

2.2 OpenFabric

The OpenFabric routing protocol is a modified version of the IS-IS protocol that aims to make the configuration of each Intermediate System (IS) in a network simpler. This protocol is specifically designed for spine and leaf

network, with the goal of improving, scaling efficiency and providing a comprehensive view of the network topology from a single point.

Spine and Leaf architecture is designed to provide a more scalable, highly available, flexible, and high-bandwidth network by eliminating the traditional three-tier hierarchical network design and replacing it with a flatter, two-tier design. It is a data center network design that separates the access layer (leaf) and the core layer (spine) of the network. In this architecture, the spine switches are responsible for routing traffic between the leaf switches and are connected to every leaf switch in a full mesh topology. The leaf switches, in turn, are responsible for connecting end devices such as servers, storage, and virtual machines, to the network.

The architecture of a leaf-spine network is inspired by the fat-tree topology, where a number of leaf switches are interconnected with several spine switches in a non-blocking configuration. This design allows each leaf switch to communicate with any other leaf switch in the network through one or more spine switches, without causing any one spine switch to become a performance bottleneck. The architecture allows for easy growth and expansion of the network, as well as the ability to add or remove leaf switches without disrupting network operations. Figure 5 shows a sample of the leaf spine architecture.

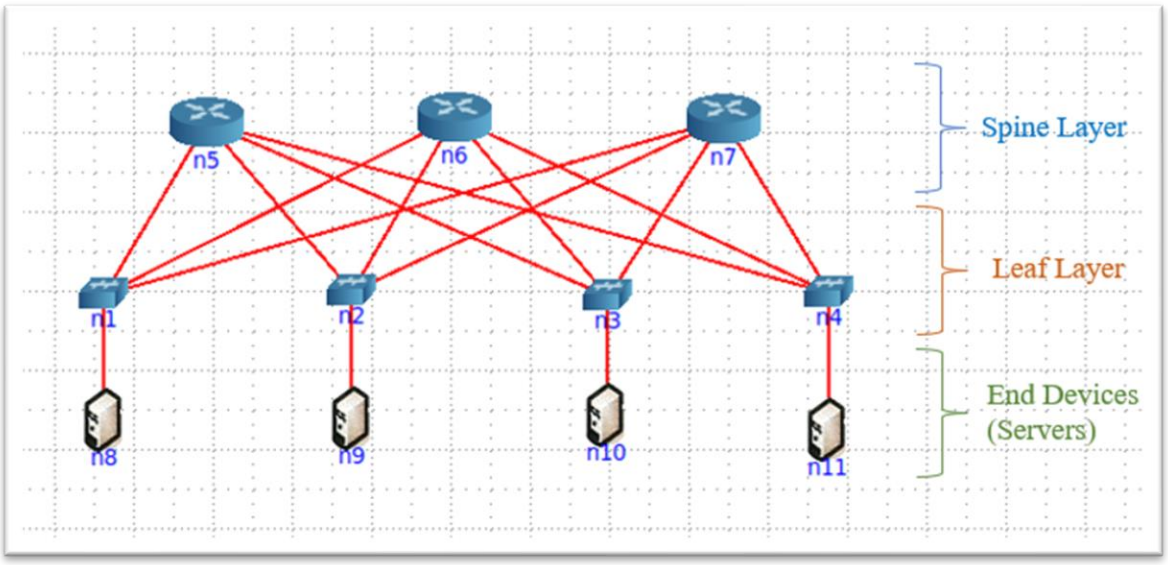


Figure 5: Spine-Leaf Architecture

OpenFabric is a link-state routing protocol, same as IS-IS, and implements the SPF algorithm for path computation. IS-IS was basically stripped of unneeded functions according to the internet draft (White, Zandi, 2018) to implement the OpenFabric routing protocol. Theoretically, OpenFabric follows the standard specified in the ISO/IEC 10589:2000 for IS-IS with couple of modifications to the following areas:

- Adjacency formation
- Advertisement of Reachability information
- Flooding Optimization
- Transit Link Reachability

OpenFabric operates in a tightly controlled data center environment. Due to the intent to increase efficiency, optimize operation, and simplify the protocol, some of the burdensome procedures encountered within the IS-IS operation were stripped down to achieve its target.

2.3 Comaprisons between IS-IS and OpenFabric

To understand the differences between IS-IS and OpenFabric routing protocols, it is best to focus on the improvements made to the IS-IS protocol in order to create OpenFabric.

2.3.1 Adjacency Formation

Due to the need for simplicity and to improve efficiency in a tightly controlled data center environment demanded for OpenFabric, the following modifications were considered for IS-IS neighbor formation process:

- IIH PDU 17 (level 2 point-to-point circuit hello) should be the only IIH PDU type transmitted. The circuit type field is set to 2 to implement this.
- Removal of support for IIH PDU 15 (level 1 broadcast hello).
- Removal of support for IIH PDU 16 (level 2 broadcast hello)
- Removal of support for broadcast link types, pseudocode, and designated IS processing.
- Support for three-way handshake to ensure two-way connectivity is established before synchronizing the link state database.

Overall, OpenFabric takes into account an efficient method of forming adjacencies that enables a new IS added to the fabric to transmit a complete table only once, resulting in minimal transmission of information for initial synchronization and reducing the time required to synchronize a new IS.

2.3.2 Advertisement of Reachability Information

The following modifications are still being considered or already implemented to further revised the IS-IS original mechanisms:

- The level 2 LSPs and the scoped flooding PDU are the only PDU types used to carry link state information.
- It is possible to remove processing related to the level 1 LSPs.
- It was revised that neighbor reachability MUST BE carried in TLV type 22.
- It was revised that IPv4 reachability SHOULD BE carried in TLV type 135, or TLV type 235 in respect to multi-topology deployment. Additionally, IPv6 reachability SHOULD BE carried in TLV type 236 or TLV type 237 for multi-topology.
- Support for processing neighbor reachability TLV (type 2) was to be removed.
- Processing related to the narrow metric IP reachability TLV (types 128 and 130) SHOULD BE removed ensuring OpenFabric network framework only offers support for wide metrics.

2.3.3 Flooding Optimization

To reduce the flooding of link state information in the form of LSPs. OpenFabric leverages the link state protocol's existing information such as the neighboring intermediate system's list of neighbors and the calculated locality of the fabric. The architecture further implements additional optimization techniques to minimize the impact of flooding on network performance. These techniques make use of local packet filtering, which prevents unnecessary packets from being forwarded. The following tables were introduced to optimize the flooding process:

- Neighbor List (NL) list: The set of neighbors
- Neighbor's Neighbors (NN) list: These are group of intermediate systems that are two hops away, identified by conducting a truncated SPF.
- Do Not Reflood (DNR) list: The set of neighbors who should have LSPs who should not reflood LSPs. At the beginning, DNR list is always empty.
- Reflood (RF) list: This includes the set of neighbors who should flood LSPs (or fragments) to their immediate neighbors to ensure synchronization. Same as DNR, RF tables are initially empty.

The OpenFabric architecture also includes mechanisms to handle potential flooding failures that may occur if a reflooder becomes disconnected from the necessary links required for flooding, which could result in incomplete LSP flooding. The process for populating the listed table were discussed extensively in the draft (White, Zandi, 2018).

2.3.4 Transient Link Reachability

The OpenFabric architecture recommends against advertising reachability for transit links, which may remain unnumbered since the IS-IS protocol does not require layer 3 IP addresses. Instead, each IS should have a single loopback address assigned an IPv6 address to provide reachability to intermediate systems in the fabric. By adopting these recommendations, the amount of control plane state carried across the network is reduced.

2.3.5 Determining and Advertising location within the Fabric

The connection of an IS to a specific tier is beneficial for enabling automatic configuration of intermediate systems on the fabric and minimizing unnecessary broadcast of traffic. The value of the tier increases as the IS gets further away from the edge of the fabric. The tiers serve as unique identifiers known as Fabric IDs, which are used to identify the location of the device within the network. Fabric IDs are typically determined using a combination of physical location information and network topology information, which allows for the creation of a hierarchical addressing scheme that can be used to efficiently route packets within the network. ISs connected to end devices, like PCs, are assumed to be in tier 0. The determination and advertisement of location information is critical for the proper operation of the OpenFabric architecture.

3 Requirements Analysis

3.1 General Objectives

The objective of this work is to illustrate the behavior of the IS-IS and OpenFabric routing protocols. To achieve this, it is necessary to select an appropriate topology that exemplifies the behavior of both protocols. Within the topology, intermediate systems in the form of routers, as well as end systems like remote PCs and HTTP servers, will be deployed. The network activity will be observed through a network monitoring tool and documented accordingly. Understanding the impact of interface and link parameters, such as metrics, delays, and bandwidth, will be critical in comprehending how both protocols negotiate between source and destination systems.

3.2 Software

The network topology implementation and emulation were facilitated by utilizing the following tools:

- CORE (Common Open Research Emulator) 8.2.0
- FRRouting (FRR)
- Ubuntu OS 20.04
- VMware Workstation 16 Player

3.2.1 CORE 8.2.0

The Common Open Research Emulator (CORE) is an open-source software tool for network emulation. It provides a way to emulate complex networks, including routers, switches, and other networking devices, using virtualization technologies.

CORE is designed to be highly configurable and flexible, allowing researchers to experiment with a variety of network topologies, protocols, and configurations. It includes a graphical user interface (GUI) that makes it easy to create, configure, and manage network topologies, as well as a Python API for programmatic control. CORE supports a wide range of popular network protocols and technologies, including IPv4 and IPv6, OSPF, IS-IS, OpenFabric, VLANs, and more. It also provides a range of tools for network monitoring and analysis, including packet capture and analysis, traffic generation and manipulation, and performance monitoring. Its open-source nature and flexible design make it a valuable tool for exploring and testing new network technologies and protocols.

CORE operates based on the following key components:

- core-daemon: This manages emulated sessions of nodes and links using Linux namespaces and bridges. It manipulates packets with traffic control and provides a gRPC API.
- core-gui: The core-gui communicates with the daemon via gRPC and provides a drag-and-drop interface for creating nodes and links, launching terminals, and saving/opening scenario files.
- vnoded: This provides s a command line utility for creating CORE node namespaces.
- vcmd: This allows a command line utility for sending shell commands to nodes.

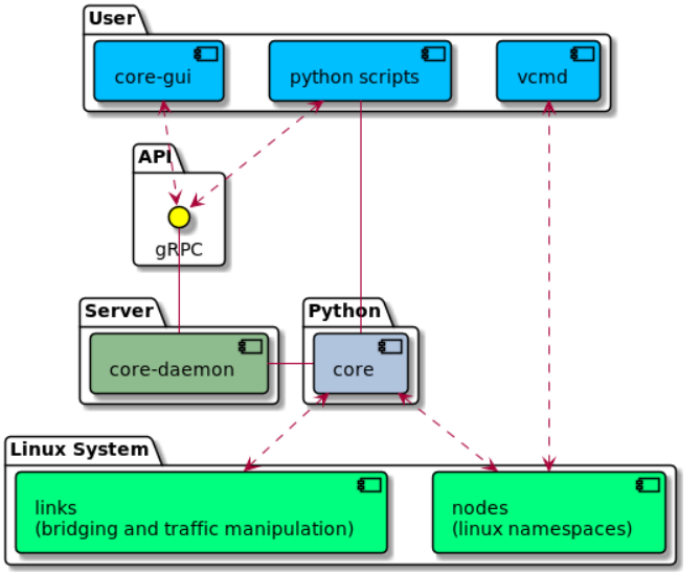


Figure 6: CORE Framework

The CORE framework allows users to create node containers on Linux by utilizing Linux namespacing and linking them together using virtual interfaces and bridging.

3.2.2 FRRouting (FRR)

FRRouting is a free and open-source dynamic routing protocol suite for Unix and Linux operating systems. It is designed to provide a flexible and scalable routing solution for various network environments. FRRouting includes support for multiple routing protocols such as OSPF, IS-IS, OpenFabric, RIP, BGP, and more. It also includes advanced features like policy routing, VPN routing, and IPv6 support.

3.2.3 Ubuntu 20.04

The latest LTS (Long-Term Support) version of the Ubuntu operating system, Ubuntu 20.04 Focal Fossa OS was used for this work. Ubuntu OS is a free, open-source operating system based on the Debian Linux distribution. For this work, Ubuntu 20.04 comes with several new features and improvements, including an updated kernel (5.4), and support to run CORE.

3.2.4 VMWare Workstation 16 Player

VMware Workstation is a popular software application that allows users to run multiple virtual machines on a single physical computer. With VMware Workstation, users can create and manage virtual machines, which are complete computer systems that run within their host operating system. This enables users to run multiple operating systems on a single computer simultaneously, without the need to partition their hard drive or reboot the system.

VMware Workstation supports a wide range of guest operating systems, including Windows, Linux, macOS, and various versions of Unix. It also provides a range of features and tools to help users manage their virtual machines, such as snapshots, cloning, and virtual networking.

4 Project Implementation

The upcoming section will outline the steps taken to create the virtual network and explore its evaluation using IS-IS and OpenFabric routing protocols in more detail.

4.1 IS-IS Network Emulation with CORE

4.1.1 Network Design

A network topology was defined to reflect the end devices, ISs and network administrative areas typical of the IS-IS framework. Figure 7 shows a snapshot of the network layout implemented for this evaluation.

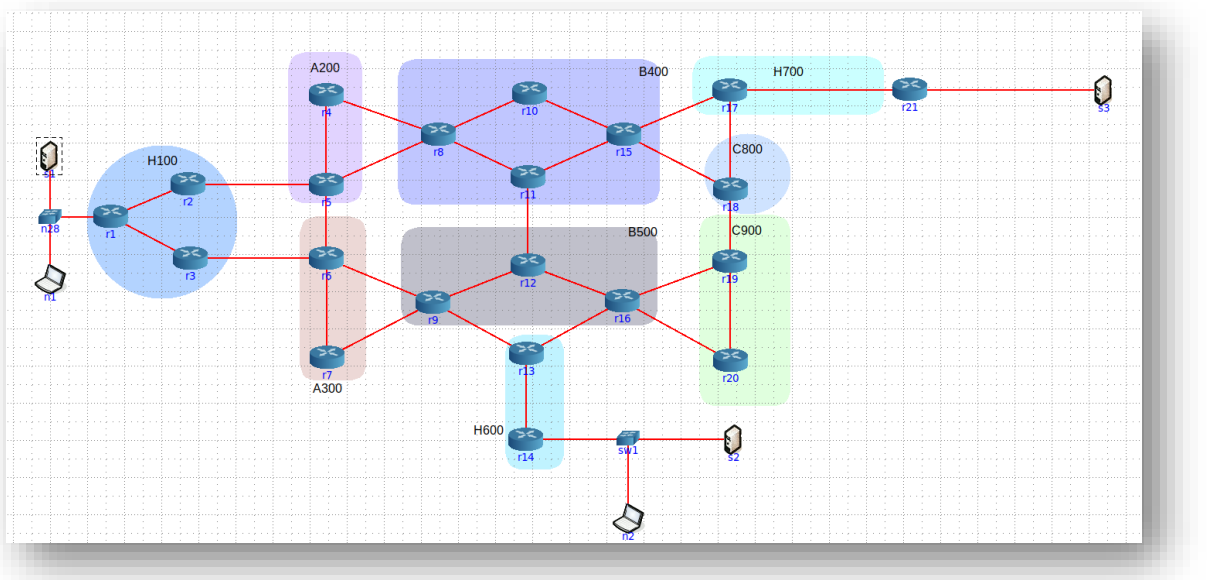


Figure 7: IS-IS Network Topology

From Figure 7, the following nodes and components were defined using the CORE GUI:

- End Systems
  - Remote PCs: n1, n2
  - HTTP Servers: s1,s2, s3
  - Switches: n28, sw1
- Routers: r1-r21
- Areas: A200, A300, B400, B500, C800, C900, H100, H200, H700
- Links

4.1.2 Configuring FRR IS-IS

The below sample configuration was deployed at the routers per interface to enable interaction with the other neighbor routers within the network.

```
interface eth0
  ip address 10.0.18.1/24
  ipv6 address 2001:18::1/64
  ip router isis B500
  ipv6 router isis B500
```

```
6       isis circuit-type level-2-only
7       isis network point-to-point
8       isis metric level-2 30
```

Line 2 and 3 respectively assigns the router's interface ipv4 and ipv6 addresses to be used with the network. Line 4 and 5 respectively defines active IP address format allowed within the network in addition to the router's Area label which could ben used to identify its administrative subdomain. Line 6 defines the circuit-type level to be used for the specific interface depending on the nature of its administrative area.

According to line 8, *isis metric level-2 30* defines the metric value used at the *eth0* interface of the *r6* IS. IS-IS network uses a default metric value of 10 but this can be reassigned per circuit to represent its capacity for handling traffic within the network. Higher value means lower capacity. The impact of these values would come to effect when calculating the best path to a destination node using the SPF algorithm.

To ensure the FRR IS-IS daemon, *isisd* must be set to yes on the daemons configuration file else the daemon will not be started if the network goes live:

```
1     isisd=yes
```

Once configurations are resolved for all the active interfaces within the network, network connections could be tested for reachability of all the nodes and system deployed using *vtysh* terminal, FRR's integrated shell that provides a combined interface to all the FRR router daemons in a single session on a Linux system.

4.1.3 Intermediate Systems Configuration

4.1.3.1 Showing active daemons, r9

```
Hello, this is FRRouting (version 7.2.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

r9# show daemons
 zebra isisd
r9#
```

Figure 8: FRR IS-IS daemon, isisd

4.1.3.2 Route Summary, r9

```
Hello, this is FRRouting (version 7.2.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

r9# show daemons
 zebra isisd
r9# show ip route summary
Route Source         Routes               FIB  (vrf default)
connected            4                    4
isis                 32                   28
------
Totals               36                   32

r9#
```

Figure 9: Route summary from router, r9

4.1.3.3 Route Summary, r15

```
Hello, this is FRRouting (version 7.2.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

r15# show ip route summary
Route Source         Routes               FIB  (vrf default)
connected            4                    4
isis                 32                   28
------
Totals               36                   32

r15#
```

Figure 10: Route summary from router, r15

Figure 9 and 10, confirms that r9 and r15 has the same view of the network topology. This confirms the link-state protocol algorithm which means that each router maintains a complete and consistent view of the network topology.

4.1.3.4 Discovered Routes, r15

```
r15# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

I>* 10.0.0.0/24 [115/20] via 10.0.3.2, eth1, 00:21:04
I>* 10.0.1.0/24 [115/50] via 10.0.3.2, eth1, 00:21:03
I   10.0.2.0/24 [115/60] via 10.0.2.1, eth0 inactive, 00:21:03
C>* 10.0.2.0/24 is directly connected, eth0, 00:21:33
I   10.0.3.0/24 [115/20] via 10.0.3.2, eth1 inactive, 00:21:04
C>* 10.0.3.0/24 is directly connected, eth1, 00:21:32
I   10.0.4.0/24 [115/20] via 10.0.4.2, eth2 inactive, 00:21:02
C>* 10.0.4.0/24 is directly connected, eth2, 00:21:31
I   10.0.5.0/24 [115/40] via 10.0.4.2, eth2, 00:21:01
C>* 10.0.5.0/24 is directly connected, eth3, 00:21:30
I>* 10.0.6.0/24 [115/50] via 10.0.3.2, eth1, 00:21:03
I>* 10.0.7.0/24 [115/30] via 10.0.3.2, eth1, 00:21:03
I>* 10.0.8.0/24 [115/40] via 10.0.3.2, eth1, 00:21:01
I>* 10.0.9.0/24 [115/20] via 10.0.3.2, eth1, 00:21:04
I>* 10.0.10.0/24 [115/80] via 10.0.3.2, eth1, 00:21:01
I>* 10.0.11.0/24 [115/80] via 10.0.3.2, eth1, 00:20:48
I>* 10.0.12.0/24 [115/60] via 10.0.3.2, eth1, 00:21:01
I>* 10.0.13.0/24 [115/20] via 10.0.4.2, eth2, 00:21:02
I>* 10.0.14.0/24 [115/20] via 10.0.4.2, eth2, 00:21:02
I>* 10.0.15.0/24 [115/50] via 10.0.3.2, eth1, 00:20:54
```

Figure 11: Routes discovered by router, r15

Section 4.1.3.4 confirms the how r15 learned all the routes shown in section 4.1.3.3. Same routes were learned in the same manner by r9 according to section 4.1.3.5.

4.1.3.5 Discovered Routes, r9.

```
r9# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

I>* 10.0.0.0/24 [115/50] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.1.0/24 [115/80] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.2.0/24 [115/80] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.3.0/24 [115/50] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.4.0/24 [115/60] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.5.0/24 [115/80] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.6.0/24 [115/80] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.7.0/24 [115/60] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.8.0/24 [115/70] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.9.0/24 [115/40] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.10.0/24 [115/80] via 10.0.20.2, eth2, 00:20:46
I>* 10.0.11.0/24 [115/60] via 10.0.20.2, eth2, 00:20:46
I>* 10.0.12.0/24 [115/90] via 10.0.19.2, eth1, 00:20:48
I>* 10.0.13.0/24 [115/70] via 10.0.18.2, eth0, 00:20:48
  *                       via 10.0.19.2, eth1, 00:20:48
I>* 10.0.14.0/24 [115/60] via 10.0.18.2, eth0, 00:20:48
  *                       via 10.0.19.2, eth1, 00:20:48
I>* 10.0.15.0/24 [115/40] via 10.0.18.2, eth0, 00:20:51
I>* 10.0.16.0/24 [115/50] via 10.0.18.2, eth0, 00:20:46
I>* 10.0.17.0/24 [115/70] via 10.0.20.2, eth2, 00:20:43
I   10.0.18.0/24 [115/60] via 10.0.18.2, eth0 inactive, 00:20:51
C>* 10.0.18.0/24 is directly connected, eth0, 00:21:20
I   10.0.19.0/24 [115/60] via 10.0.19.2, eth1 inactive, 00:20:48
```

Figure 12: Routes discovered by router, r9

4.1.3.6 System IDs View, r9

```
r9# show isis hostname
Level  System ID      Dynamic Hostname
2      0000.0000.0007 r7
2      0000.0000.0013 r13
2      0000.0000.0012 r12
2      0000.0000.0020 r20
2      0000.0000.0004 r4
2      0000.0000.0005 r5
2      0000.0000.0008 r8
2      0000.0000.0010 r10
2      0000.0000.0011 r11
2      0000.0000.0015 r15
2      0000.0000.0016 r16
2      0000.0000.0017 r17
2      0000.0000.0018 r18
2      0000.0000.0019 r19
2      0000.0000.0014 r14
2      0000.0000.0006 r6
2      0000.0000.0002 r2
2      0000.0000.0003 r3
2      0000.0000.0021 r21
2      0000.0000.0001 r1
     * 0000.0000.0009 r9
```

Figure 13: IS-IS network System IDs discovered by r9

4.1.3.7 System IDs View, r15

```
r15# show isis hostname
Level  System ID      Dynamic Hostname
2      0000.0000.0011 r11
2      0000.0000.0010 r10
2      0000.0000.0005 r5
2      0000.0000.0008 r8
2      0000.0000.0017 r17
2      0000.0000.0018 r18
2      0000.0000.0004 r4
2      0000.0000.0012 r12
2      0000.0000.0019 r19
2      0000.0000.0013 r13
2      0000.0000.0016 r16
2      0000.0000.0007 r7
2      0000.0000.0009 r9
2      0000.0000.0020 r20
2      0000.0000.0006 r6
2      0000.0000.0014 r14
2      0000.0000.0002 r2
2      0000.0000.0003 r3
2      0000.0000.0021 r21
2      0000.0000.0001 r1
     * 0000.0000.0015 r15
```

Figure 14: IS-IS network System IDs discovered by r15

4.1.3.8 Interface Metrics, r9

```
r9# show isis interface detail
Area B500:
  Interface: eth0, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 30, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.18.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:51/64
    IPv6 Prefixes:
      2001:18::1/64

  Interface: eth1, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 30, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.19.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:53/64
    IPv6 Prefixes:
      2001:19::1/64

  Interface: eth2, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 30, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.20.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:56/64
    IPv6 Prefixes:
      2001:20::1/64

  Interface: eth3, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 30, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.21.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:58/64
    IPv6 Prefixes:
      2001:21::1/64
```

Figure 15: Details of r9 interfaces showing metrics and circuit levels

4.1.3.9 Interface Metrics, r15

```
r15# show isis interface detail
Area B400:
  Interface: eth0, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 30, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.2.2/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:3e/64
    IPv6 Prefixes:
      2001:2::2/64

  Interface: eth1, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 10, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.3.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:3f/64
    IPv6 Prefixes:
      2001:3::1/64

  Interface: eth2, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 10, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.4.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:41/64
    IPv6 Prefixes:
      2001:4::1/64

  Interface: eth3, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 30, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.5.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:43/64
    IPv6 Prefixes:
      2001:5::1/64
```

Figure 16: Details of r15 interfaces showing metrics and circuit levels

4.1.3.10 Topology View, r9

```
r9# show isis topology
Area B500:
IS-IS paths to level-2 routers that speak IP
Vertex               Type         Metric Next-Hop             Interface Parent
r9
10.0.18.0/24         IP internal  0                                     r9(4)
10.0.19.0/24         IP internal  0                                     r9(4)
10.0.20.0/24         IP internal  0                                     r9(4)
10.0.21.0/24         IP internal  0                                     r9(4)
r13                  TE-IS        30     r13                 eth0      r9(4)
r12                  TE-IS        30     r12                 eth1      r9(4)
r6                   TE-IS        30     r6                  eth2      r9(4)
r7                   TE-IS        30     r7                  eth3      r9(4)
r16                  TE-IS        40     r13                 eth0      r13(4)
                                         r12                 eth1      r12(4)
r14                  TE-IS        40     r13                 eth0      r13(4)
r11                  TE-IS        40     r12                 eth1      r12(4)
10.0.23.0/24         IP TE        40     r13                 eth0      r13(4)
10.0.15.0/24         IP TE        40     r13                 eth0      r13(4)
10.0.22.0/24         IP TE        40     r12                 eth1      r12(4)
10.0.9.0/24          IP TE        40     r12                 eth1      r12(4)
r19                  TE-IS        50     r13                 eth0      r16(4)
                                         r12                 eth1
r8                   TE-IS        50     r12                 eth1      r11(4)
r15                  TE-IS        50     r12                 eth1      r11(4)
10.0.25.0/24         IP TE        50     r13                 eth0      r16(4)
                                         r12                 eth1
10.0.16.0/24         IP TE        50     r13                 eth0      r14(4)
10.0.0.0/24          IP TE        50     r12                 eth1      r11(4)
10.0.3.0/24          IP TE        50     r12                 eth1      r11(4)
r3                   TE-IS        60     r6                  eth2      r6(4)
r18                  TE-IS        60     r13                 eth0      r19(4)
                                         r12                 eth1      r15(4)
r5                   TE-IS        60     r12                 eth1      r8(4)
10.0.18.0/24         IP TE        60     r13                 eth0      r13(4)
10.0.19.0/24         IP TE        60     r12                 eth1      r12(4)
10.0.20.0/24         IP TE        60     r6                  eth2      r6(4)
10.0.26.0/24         IP TE        60     r6                  eth2      r6(4)
                                         r7                  eth3      r7(4)
10.0.11.0/24         IP TE        60     r6                  eth2      r6(4)
10.0.21.0/24         IP TE        60     r7                  eth3      r7(4)
10.0.14.0/24         IP TE        60     r13                 eth0      r19(4)
                                         r12                 eth1
10.0.7.0/24          IP TE        60     r12                 eth1      r8(4)
10.0.4.0/24          IP TE        60     r12                 eth1      r15(4)
r20                  TE-IS        70     r13                 eth0      r16(4)
                                         r12                 eth1
r1                   TE-IS        70     r6                  eth2      r3(4)
r17                  TE-IS        70     r13                 eth0      r18(4)
                                         r12                 eth1
r2                   TE-IS        70     r12                 eth1      r5(4)
10.0.24.0/24         IP TE        70     r13                 eth0      r16(4)
                                         r12                 eth1
10.0.17.0/24         IP TE        70     r6                  eth2      r3(4)
10.0.13.0/24         IP TE        70     r13                 eth0      r18(4)
```

Figure 17: Network topology view from r9

4.1.3.11 Topology View, r15

```
r15# show isis topology
Area B400:
IS-IS paths to level-2 routers that speak IP
Vertex               Type         Metric Next-Hop             Interface Parent
r15
10.0.2.0/24          IP internal  0                                     r15(4)
10.0.3.0/24          IP internal  0                                     r15(4)
10.0.4.0/24          IP internal  0                                     r15(4)
10.0.5.0/24          IP internal  0                                     r15(4)
r11                  TE-IS        10     r11                 eth1      r15(4)
r18                  TE-IS        10     r18                 eth2      r15(4)
r8                   TE-IS        20     r11                 eth1      r11(4)
r12                  TE-IS        20     r11                 eth1      r11(4)
r19                  TE-IS        20     r18                 eth2      r18(4)
r17                  TE-IS        20     r18                 eth2      r18(4)
10.0.0.0/24          IP TE        20     r11                 eth1      r11(4)
10.0.3.0/24          IP TE        20     r11                 eth1      r11(4)
10.0.9.0/24          IP TE        20     r11                 eth1      r11(4)
10.0.4.0/24          IP TE        20     r18                 eth2      r18(4)
10.0.14.0/24         IP TE        20     r18                 eth2      r18(4)
10.0.13.0/24         IP TE        20     r18                 eth2      r18(4)
r10                  TE-IS        30     r10                 eth0      r15(4)
r5                   TE-IS        30     r11                 eth1      r8(4)
r16                  TE-IS        30     r11                 eth1      r12(4)
                                         r18                 eth2      r19(4)
r21                  TE-IS        30     r18                 eth2      r17(4)
10.0.7.0/24          IP TE        30     r11                 eth1      r8(4)
10.0.22.0/24         IP TE        30     r11                 eth1      r12(4)
10.0.25.0/24         IP TE        30     r18                 eth2      r19(4)
10.0.30.0/24         IP TE        30     r18                 eth2      r17(4)
r2                   TE-IS        40     r11                 eth1      r5(4)
r13                  TE-IS        40     r11                 eth1      r16(4)
                                         r18                 eth2
10.0.5.0/24          IP TE        40     r18                 eth2      r17(4)
10.0.8.0/24          IP TE        40     r11                 eth1      r5(4)
10.0.23.0/24         IP TE        40     r11                 eth1      r16(4)
                                         r18                 eth2
10.0.31.0/24         IP TE        40     r18                 eth2      r21(4)
r4                   TE-IS        50     r11                 eth1      r8(4)
r9                   TE-IS        50     r11                 eth1      r12(4)
r20                  TE-IS        50     r18                 eth2      r19(4)
r1                   TE-IS        50     r11                 eth1      r2(4)
r14                  TE-IS        50     r11                 eth1      r13(4)
                                         r18                 eth2
10.0.1.0/24          IP TE        50     r11                 eth1      r8(4)
10.0.6.0/24          IP TE        50     r11                 eth1      r8(4)
10.0.19.0/24         IP TE        50     r11                 eth1      r12(4)
10.0.27.0/24         IP TE        50     r18                 eth2      r19(4)
10.0.29.0/24         IP TE        50     r11                 eth1      r2(4)
10.0.15.0/24         IP TE        50     r11                 eth1      r13(4)
                                         r18                 eth2
```

Figure 18: Network topology view from r15

4.1.3.12 Link-State Database, r15

```
r15# show isis database
Area B400:
IS-IS Level-2 link-state database:
LSP ID                  PduLen  SeqNumber   Chksum  Holdtime  ATT/P/OL
r1.00-00                   147   0x00000005  0xd0f3     581    0/0/0
r2.00-00                   125   0x00000005  0x2faf     590    0/0/0
r3.00-00                   125   0x00000005  0x4795     561    0/0/0
r4.00-00                   125   0x00000006  0xbad4     459    0/0/0
r5.00-00                   191   0x00000006  0x6c32     578    0/0/0
r6.00-00                   191   0x00000006  0x931c     576    0/0/0
r7.00-00                   125   0x00000006  0x5fbe     517    0/0/0
r8.00-00                   191   0x00000006  0x0cec     563    0/0/0
r9.00-00                   191   0x00000006  0xffb6     529    0/0/0
r10.00-00                  126   0x00000006  0x7cfe     558    0/0/0
r11.00-00                  159   0x00000006  0x9f5d     519    0/0/0
r12.00-00                  161   0x00000006  0x3608     526    0/0/0
r13.00-00                  159   0x00000005  0x7eaa     570    0/0/0
r14.00-00                  115   0x00000005  0xe3c5     573    0/0/0
r15.00-00            *     192   0x00000006  0xebb0     531    0/0/0
r16.00-00                  192   0x00000006  0x7955     544    0/0/0
r17.00-00                  159   0x00000005  0xcf5e     559    0/0/0
r18.00-00                  159   0x00000006  0x2e6a     558    0/0/0
r19.00-00                  159   0x00000006  0x6774     534    0/0/0
r20.00-00                  126   0x00000006  0x3f69     523    0/0/0
r21.00-00                  115   0x00000005  0x2906     535    0/0/0
    21 LSPs
```

Figure 19: r15 link-state database

4.1.3.13 Link-State Database, r2

```
Hello, this is FRRouting (version 7.2.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

r2# show isis database
Area H100:
IS-IS Level-1 link-state database:
LSP ID                  PduLen  SeqNumber   Chksum  Holdtime  ATT/P/OL
r1.00-00                   147   0x00000006  0x9512    1102    0/0/0
r2.00-00             *     114   0x00000006  0xaf4a    1035    0/0/0
r3.00-00                   114   0x00000006  0x7c7c    1083    0/0/0
    3 LSPs

IS-IS Level-2 link-state database:
LSP ID                  PduLen  SeqNumber   Chksum  Holdtime  ATT/P/OL
r1.00-00                   147   0x00000006  0xcef4    1105    0/0/0
r2.00-00             *     125   0x00000006  0x2db0    1136    0/0/0
r3.00-00                   125   0x00000006  0x4596    1068    0/0/0
r4.00-00                   125   0x00000007  0xb8d5     990    0/0/0
r5.00-00                   191   0x00000007  0x6a33    1088    0/0/0
r6.00-00                   191   0x00000007  0x911d    1105    0/0/0
r7.00-00                   125   0x00000007  0x5dbf    1043    0/0/0
r8.00-00                   191   0x00000007  0x0aed    1134    0/0/0
r9.00-00                   191   0x00000007  0xfdb7    1074    0/0/0
r10.00-00                  126   0x00000007  0x7aff    1093    0/0/0
r11.00-00                  159   0x00000007  0x9d5e    1045    0/0/0
r12.00-00                  161   0x00000007  0x3409    1055    0/0/0
r13.00-00                  159   0x00000006  0x7cab    1109    0/0/0
r14.00-00                  115   0x00000006  0xe1c6    1126    0/0/0
r15.00-00                  192   0x00000007  0xe9b1    1077    0/0/0
r16.00-00                  192   0x00000007  0x7756    1105    0/0/0
r17.00-00                  159   0x00000006  0xcd5f    1111    0/0/0
r18.00-00                  159   0x00000007  0x2c6b    1092    0/0/0
r19.00-00                  159   0x00000007  0x6575    1067    0/0/0
r20.00-00                  126   0x00000007  0x3d6a    1036    0/0/0
r21.00-00                  115   0x00000006  0x2707    1059    0/0/0
    21 LSPs
```

Figure 20: r2 link-state database showing separate level-1 and level-2 database

It can be observed from 4.1.3.13 that the r2 created two link-state databases as the router was configured as Level 1-2 IS. This further confirms that a level 1-2 router would create a separate level 1 and level 2 link-state database. The r2 interfaces were configured as follows:

```
interface eth1
  ip address 10.0.29.2/24
  ipv6 address 2001:17::2/64
  ip router isis H100
  ipv6 router isis H100
  isis circuit-type level-1-2
  isis network point-to-point
  isis metric level-1 10
```

4.1.4 Nodes Reachability

4.1.4.1 Traceroute n1 to s3

This was implemented with the two-node tool embedded in the CORE network emulator. This allows users to select the source node and destination node to test network connectivity between them. It is a useful tool for troubleshooting network issues and verifying network configurations.

Figure 21 shows s3 is reachable to n1 through *r1-r2-r5-r8-**r11**-r15-r8-r17-r21,* ISs chain. This route was considered in view of the best path available at the time . If a node is assumed to have gone down, a new SPF tree would be recalculated to take into consideration the next best path routes depending on the metric available at the link per time.

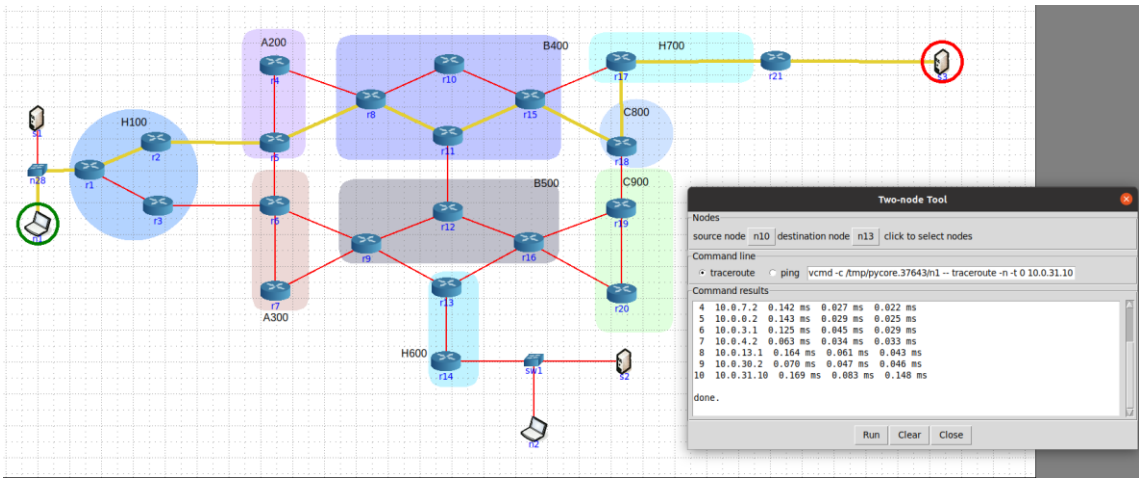


Figure 21: Node reachability check with traceroute; n1 to s3

Figure 22 shows a new route was negotiated to reach S3 in an event ***r11*** is not available and the link attributes is not favoring the SPF algorithm. In this case a new route was negotiated via a new ISs chain, *r1-r2-r5-r8-**r10**-r15-r8-r17-r21,* compared to what was used in Figure 21.

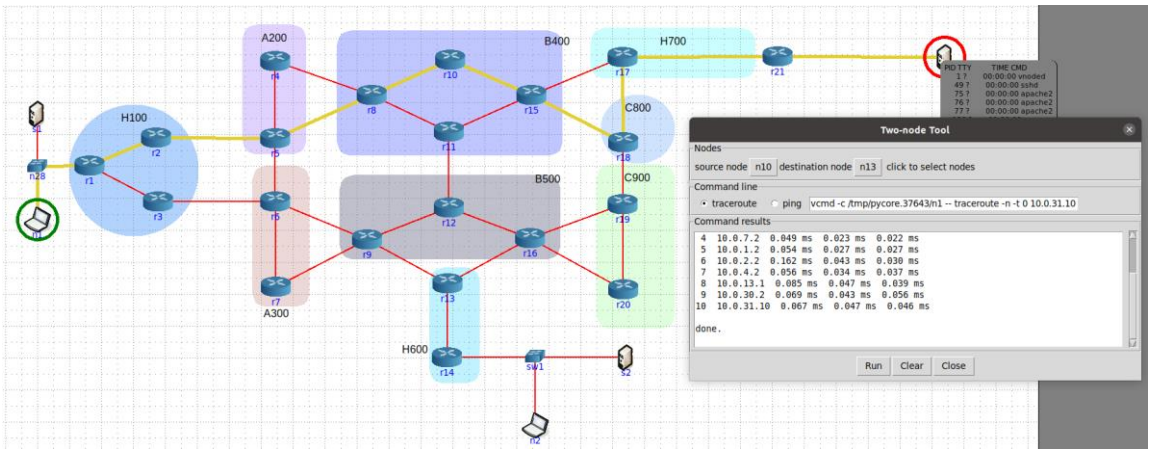


Figure 22: Node reachability check with traceroute; n1 to s3 (r11 was shutdown)

4.1.4.2 Traceroute n2 to s1

A similar demonstration according to section 4.1.4.1 was performed to observe the reachability of n2 from n1. Figure 23 shows n2 is reachable from n1 through the ISs chain, *r14-r13-r16-r12-r11-r8-r5-r2-r1* considering the best path through the SPF algorithm.

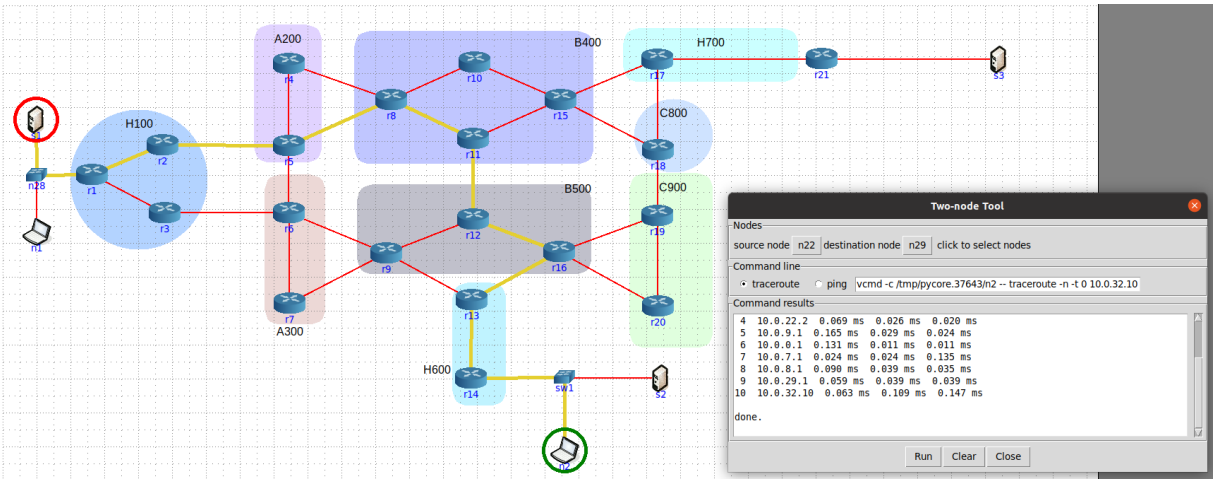


Figure 23: Node reachability check with traceroute; n2 to s1

Further tweak to the links would impact this best route negotiated for *s1* from *n2*. Figure 24 shows the impact of to the SPF tree if for instance *r11* goes down. This causes a new route to be chosen via new set of ISs chain. From Figure 24, it could be seen that a new ISs chain, *r14-r13-r9-r5-r3-r1* was selected to be able to reach *s1*.

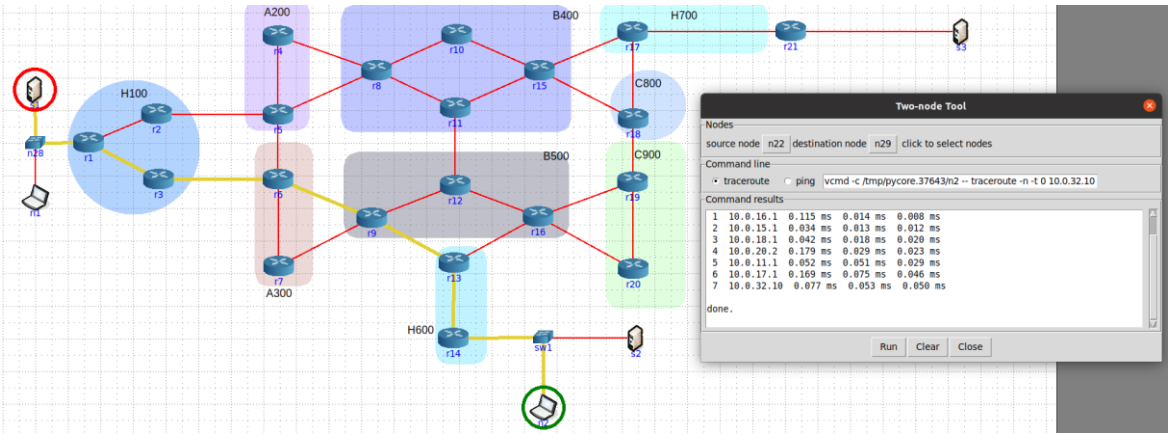


Figure 24: Node reachability check with traceroute; n2 to s1 (r11 was shutdown)

In an additional step to observe further route negotiation from n2 if for instance, *r9* and *r11* goes down, Figure 25 shows the new route when *r9* and *r11* are not available, indicating the SPF algorithm would opt for the next best path reach to reach *s1*. It was preferable to use a much longer chain of ISs to reach s1 in this case.

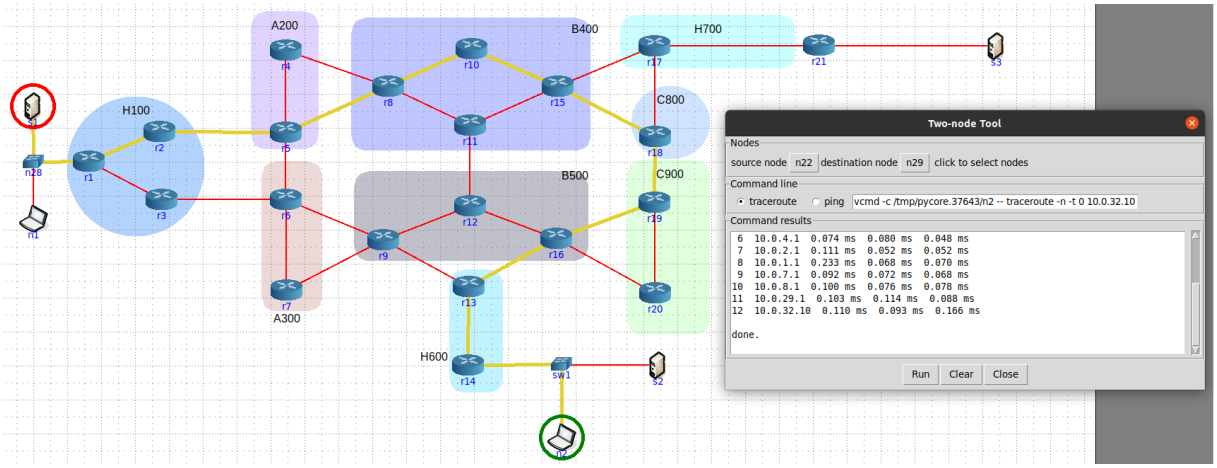


Figure 25: Nodes reachability check with traceroute; n2 to s1 (r9 and r11 were shutdown)

4.1.4.3 Traceroute n1 to s2

In a different observation concerning the bandwidth between the links. The node reachability between *n1* and *s2* was considered. At start all links were assigned an unlimited bandwidth with a default metric value of 10 across all the interfaces. This ensures each of the links is as good and preferable as the other. Figure 26 shows the behavior of the network with constant bandwidth across broad.

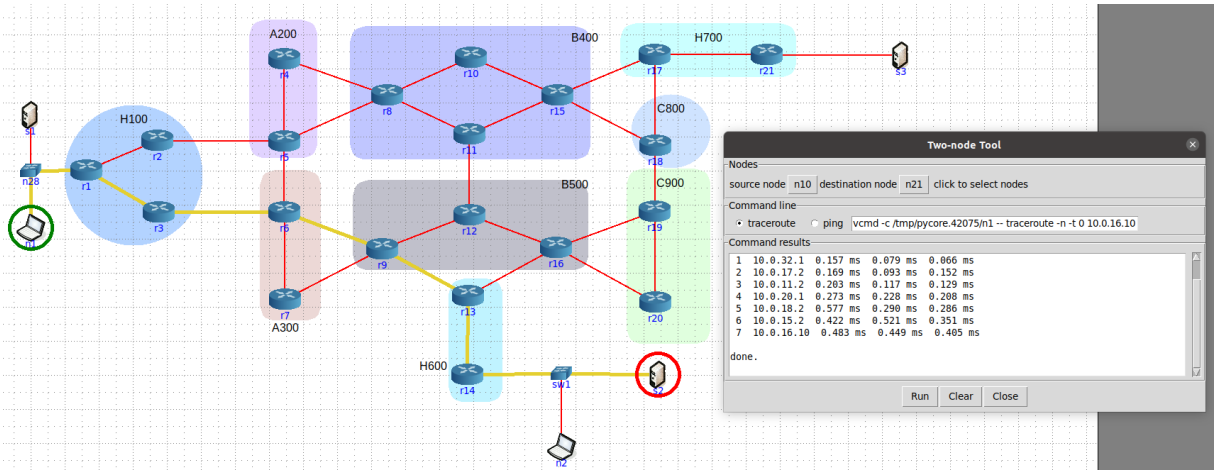


Figure 26: Nodes reachability check with traceroute; n1 to s2 (equal bandwidth)

Further tweaks to the bandwidth during the live network observation, especially by altering links *r5-r9* and *r9-r13* only shows that the default metric value was still used to decide the SPF tree. Since all the links maintained same default metric value of 10, the SPF tree is tasked to select the route with least hops to the destination regardless of the transit delay. A lower bandwidth of 64.00 Kbps (bits per second) only impacted the reachability of *s2* from *n1* in terms of seconds.

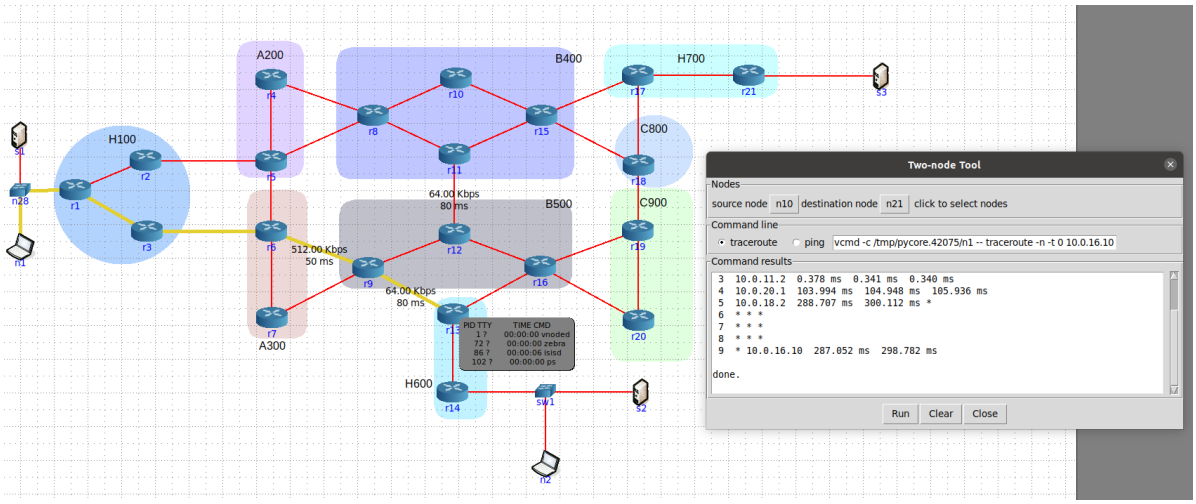


Figure 27: Nodes reachability check with traceroute; n1 to s2 (altered bandwidth)

4.1.5 Packet Capture

Wireshark capture of the traffic activity further demonstrates the deployment of the IS-IS routing protocol within the subject network defined in section 4.1.1.

4.1.5.1 Hello Packet, r14

The Figure 28 shows the capture from r14 indicating the IS-IS Hello Packet, area address and the associated system ID within the network.

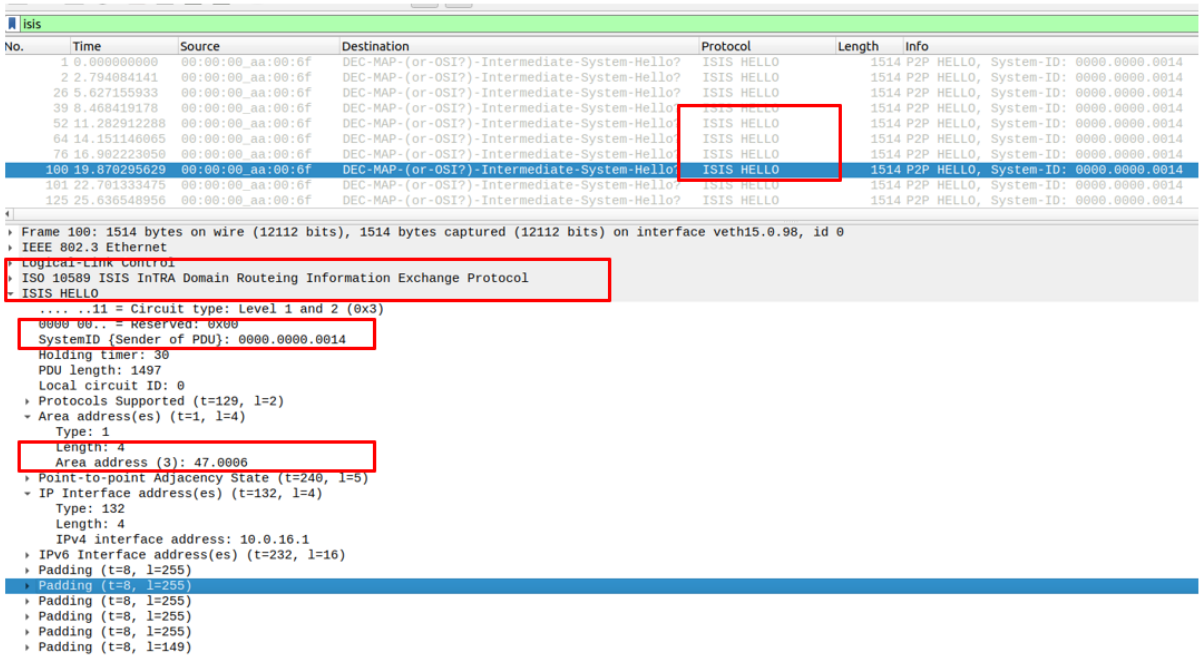


Figure 28: IS-IS Network Hello Packets captured with Wireshark

4.1.5.2 HTTP GET Request n1 to s2

An HTTP server was set up within the network at *s2* as part of the network emulation. The Figure 29 shows the Wireshark capture of the network activity when *n1* downloaded example.txt from the *s2* server.

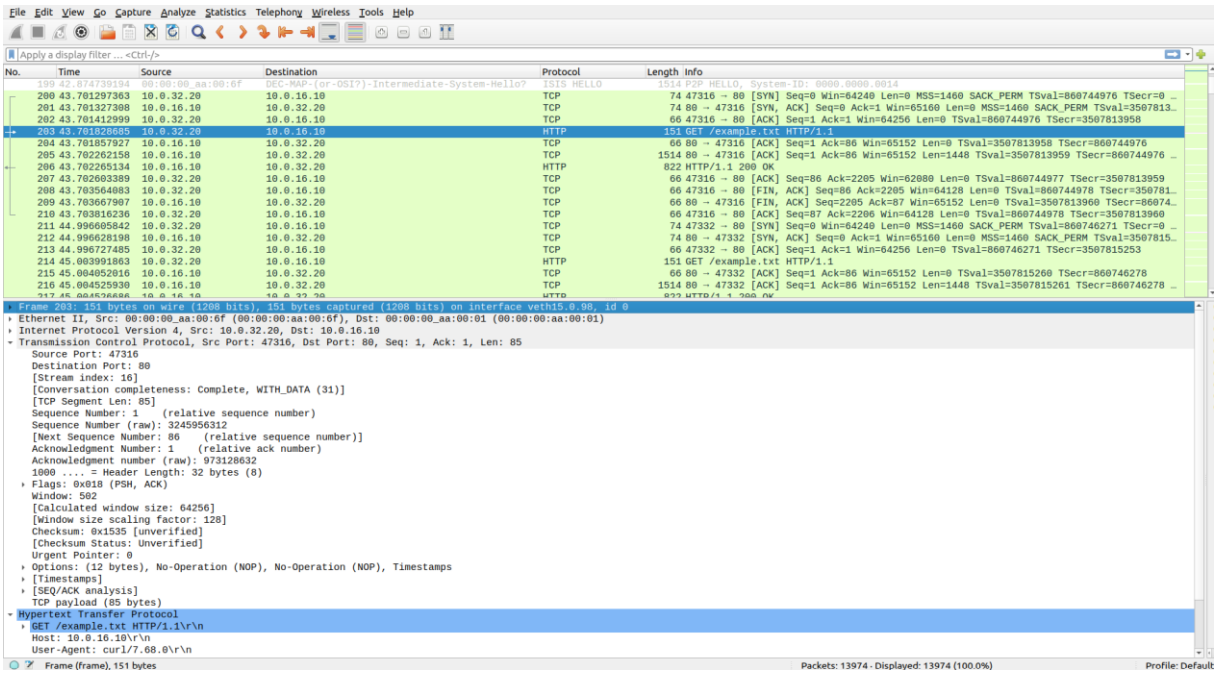


Figure 29: HTTP Traffic captured with Wireshark

The below sample bash script was deployed to establish communication with the HTTP servers, *s1, s2 and s3* from *n1* and *n2* simultaneously.

```
#!/bin/bash

  while true; do
      curl -o /tmp/test-$(date +%s).txt 10.0.31.10/example.txt
      curl -o /tmp/test-$(date +%s).txt 10.0.16.10/example.txt
      sleep 5
  done

```

The bash script was implemented under the *atd* utility node service having made few modifications and saved as a bash script, *startatd.sh* which could be started with the nodes n1 and n2 once the network is active. In general, each node continuously downloads *example.txt* file from the servers and save them with a new name until the process is terminated Each download activity is activated by running the *startatd.sh* script per node via the bash terminal as follows:

```
chmod 777 startatd.sh
./startatd.sh
```

4.2 OpenFabric Network Emulation with CORE

4.2.1 Network Design

A network suitable to emulate the behavior of the spine and leaf architecture was set up to cater for high availability, great flexibility and reduce the performance bottleneck within the network. Figure 30 shows the snapshot of the network implemented for this part of the task.

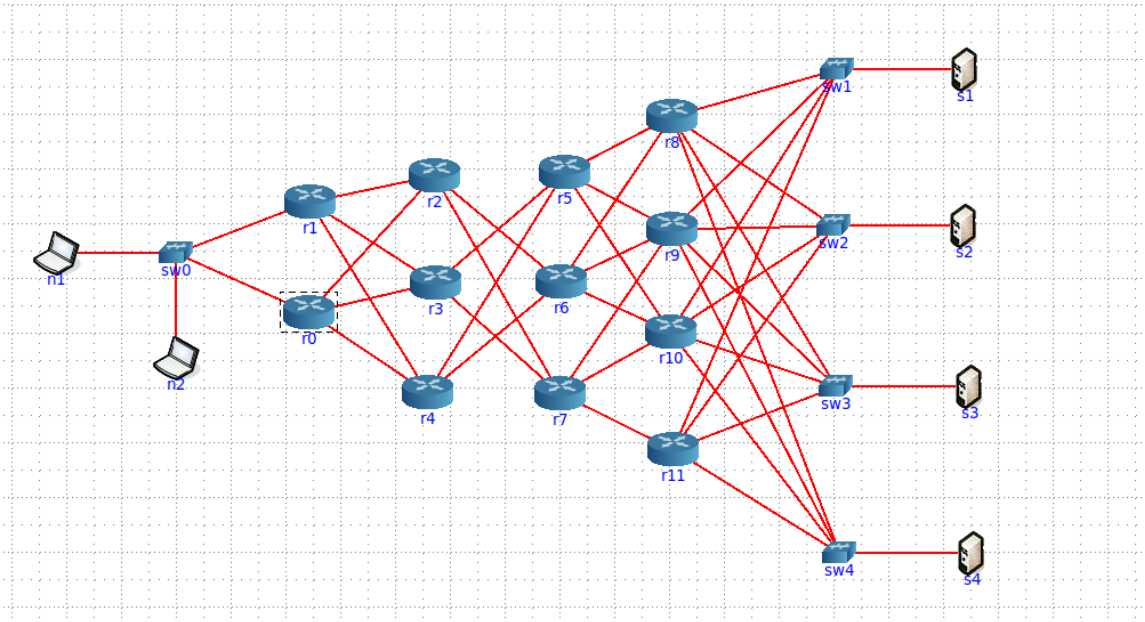


Figure 30: OpenFabric Network Topology

From Figure 30, the following nodes and components were deployed with the help of CORE GUI:

- End Systems
  - Remote PCs: n1, n2
  - HTTP Servers: s1,s2, s3, s4
  - Switches: sw0, sw1, sw2, sw3
- Routers: r1-r11
- Areas: 1
- Links

4.2.2 Configuring FRR OpenFabric

The below sample configuration was deployed per interface at the routers, *r0-r11* to enable participation of all active interfaces in the OpenFabric network. The following snippet shows the interface configuration deployed for interface *eth0* of *r6*.

```
1    interface eth0
2      ip address 10.0.6.1/24
3      ipv6 address 2001:6::1/64
4      ip router openfabric 1
5      ipv6 router openfabric 1
```

Line 2 and 3 respectively assigns ipv4 and ipv6 address to be used within the network. Line 4 and 5 respectively defines active IP address format allowed within the OpenFabric network alongside the router's area label which could be to identify its administrative subdomain.

In addition to the interface configuration each router must be activated to participate in the OpenFabric network by assigning it a unique NET address as follow:

```
1    router openfabric 1
2      net 47.0000.0000.0000.0006.00
```

To allow the FRR OpenFabric daemon to be started within the emulation environment, the fabric daemon must be turned on in the daemon configuration file with the following command:

```
1    fabricd=yes
```

Again, once all configurations are resolved for the active interfaces within the network, network connections could be tested for reachability of all the nodes and systems deployed using FRR's *vtysh* terminal.

4.2.3 Intermediate Systems Configuration

4.2.3.1 Showing active daemons, r6

```
r6# show daemons
 zebra fabricd
r6#
```

Figure 31: FRR OpenFabric daemon, fabricd

4.2.3.2 Route Summary, r6

```
r6# show ip route summary
Route Source         Routes               FIB  (vrf default)
connected            5                    5
openfabric           26                   21
------
Totals               31                   26

r6#
```

Figure 32: Route summary from router, r6

4.2.3.3 Route Summary, r2

```
Hello, this is FRRouting (version 7.2.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

r2# show ip route summary
Route Source         Routes               FIB  (vrf default)
connected            4                    4
openfabric           26                   22
------
Totals               30                   26

r2#
```

Figure 33: Route summary from r2

4.2.3.4 Discovered Routes, r6

```
r6# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

f>* 10.0.0.0/24 [115/20] via 10.0.6.2, eth0 onlink, 00:13:56
  *                      via 10.0.7.2, eth1 onlink, 00:13:56
  *                      via 10.0.11.2, eth2 onlink, 00:13:56
f>* 10.0.1.0/24 [115/20] via 10.0.6.2, eth0 onlink, 00:13:56
  *                      via 10.0.7.2, eth1 onlink, 00:13:56
  *                      via 10.0.11.2, eth2 onlink, 00:13:56
f>* 10.0.2.0/24 [115/20] via 10.0.6.2, eth0 onlink, 00:13:56
  *                      via 10.0.7.2, eth1 onlink, 00:13:56
  *                      via 10.0.11.2, eth2 onlink, 00:13:56
f>* 10.0.3.0/24 [115/20] via 10.0.6.2, eth0 onlink, 00:13:56
  *                      via 10.0.7.2, eth1 onlink, 00:13:56
  *                      via 10.0.11.2, eth2 onlink, 00:13:56
f>* 10.0.4.0/24 [115/20] via 10.0.7.2, eth1 onlink, 00:13:56
f>* 10.0.5.0/24 [115/30] via 10.0.6.2, eth0 onlink, 00:13:53
  *                      via 10.0.7.2, eth1 onlink, 00:13:53
  *                      via 10.0.11.2, eth2 onlink, 00:13:53
```

Figure 34: Routes discovered by r6

4.2.3.5 Discovered Routes, r2

```
r2# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

f>* 10.0.0.0/24 [115/30] via 10.0.15.2, eth0 onlink, 00:25:52
  *                      via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.1.0/24 [115/30] via 10.0.15.2, eth0 onlink, 00:25:52
  *                      via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.2.0/24 [115/30] via 10.0.15.2, eth0 onlink, 00:25:52
  *                      via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.3.0/24 [115/30] via 10.0.15.2, eth0 onlink, 00:25:52
  *                      via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.4.0/24 [115/20] via 10.0.15.2, eth0 onlink, 00:25:57
f>* 10.0.5.0/24 [115/20] via 10.0.15.2, eth0 onlink, 00:25:57
f>* 10.0.6.0/24 [115/20] via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.7.0/24 [115/20] via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.8.0/24 [115/30] via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.9.0/24 [115/30] via 10.0.15.2, eth0 onlink, 00:25:52
  *                      via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.10.0/24 [115/30] via 10.0.15.2, eth0 onlink, 00:25:52
  *                       via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.11.0/24 [115/20] via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.12.0/24 [115/20] via 10.0.15.2, eth0 onlink, 00:25:57
f>* 10.0.13.0/24 [115/20] via 10.0.16.2, eth1 onlink, 00:25:52
f>* 10.0.14.0/24 [115/30] via 10.0.16.2, eth1 onlink, 00:25:36
  *                       via 10.0.19.1, eth2 onlink, 00:25:36
  *                       via 10.0.23.1, eth3 onlink, 00:25:36
```

Figure 35: Routes discovered by r2

4.2.3.6 System IDs View, r6

```
Hello, this is FRRouting (version 7.2.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

r6# show openfabric hostname
Level  System ID      Dynamic Hostname
2      0000.0000.0008 r8
2      0000.0000.0009 r9
2      0000.0000.0017 r0
2      0000.0000.0002 r2
2      0000.0000.0001 r1
2      0000.0000.0003 r3
2      0000.0000.0004 r4
2      0000.0000.0005 r5
2      0000.0000.0007 r7
     * 0000.0000.0006 r6
r6#
```

Figure 36: OpenFabric Network System IDs discovered by r6

4.2.3.7 System IDs View, r2

```
r2# show openfabric hostname
Level  System ID      Dynamic Hostname
2      0000.0000.0004 r4
2      0000.0000.0005 r5
2      0000.0000.0006 r6
2      0000.0000.0007 r7
2      0000.0000.0008 r8
2      0000.0000.0009 r9
2      0000.0000.0010 r10
2      0000.0000.0011 r11
2      0000.0000.0003 r3
2      0000.0000.0001 r1
2      0000.0000.0017 r0
     * 0000.0000.0002 r2
r2#
```

Figure 37: OpenFabric Network System IDs discovered by r2

4.2.3.8 Interface Metrics, r6

```
r6# show openfabric interface detail
Area 1:
  Interface: eth0, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 10, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.6.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:14/64
    IPv6 Prefixes:
      2001:6::1/64

  Interface: eth1, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 10, Active neighbors: 0
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.7.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:16/64
    IPv6 Prefixes:
      2001:7::1/64
```

Figure 38: Details of r6 interfaces showing metrics and circuit level

4.2.3.9 Interface Metrics, r2

```
r2# show openfabric interface detail
Area 1:
  Interface: eth0, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 10, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.15.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:26/64
    IPv6 Prefixes:
      2001:15::1/64

  Interface: eth1, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 10, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.16.1/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:28/64
    IPv6 Prefixes:
      2001:16::1/64

  Interface: eth2, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 10, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.19.2/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:2f/64
    IPv6 Prefixes:
      2001:19::2/64

  Interface: eth3, State: Up, Active, Circuit Id: 0x0
    Type: p2p, Level: L2
    Level-2 Information:
      Metric: 10, Active neighbors: 1
      Hello interval: 3, Holddown count: 10 (pad)
      CNSP interval: 10, PSNP interval: 2
    IP Prefix(es):
      10.0.23.2/24
    IPv6 Link-Locals:
      fe80::200:ff:feaa:3c/64
    IPv6 Prefixes:
      2001:23::2/64

r2#
```

Figure 39: Details of r2 interfaces showing metrics and circuit level

4.2.3.10 Link-State Database, r6

```
r6# show openfabric database
Area 1:
IS-IS Level-2 link-state database:
LSP ID                  PduLen  SeqNumber   Chksum  Holdtime  ATT/P/OL
r1.00-00                   195   0x00000005  0x0d21     303    0/0/0
r2.00-00                   195   0x00000005  0xbc84     357    0/0/0
r3.00-00                   195   0x00000006  0x2506    1141    0/0/0
r4.00-00                   195   0x00000005  0x7cb8     339    0/0/0
r5.00-00                   228   0x00000005  0x2e7d     368    0/0/0
r6.00-00             *     228   0x00000005  0xc9ec     324    0/0/0
r7.00-00                   228   0x00000005  0xe1c4     336    0/0/0
r8.00-00                   261   0x00000074  0x9513    1125    0/0/0
r9.00-00                   294   0x00000006  0xec5d     517    0/0/0
r10.00-00                  284   0x00000074  0x9a46    1177    0/0/0
r11.00-00                  229   0x00000074  0x89f9    1173    0/0/0
r0.00-00                   195   0x00000005  0x8c7e     386    0/0/0
    12 LSPs

r6#
```

Figure 40: r6 link-state database

4.2.3.11 Link-State Database, r2

```
r2# show openfabric database
Area 1:
IS-IS Level-2 link-state database:
LSP ID                  PduLen  SeqNumber   Chksum  Holdtime  ATT/P/OL
r1.00-00                   195   0x00000005  0x0d21     325    0/0/0
r2.00-00             *     195   0x00000005  0xbc84     379    0/0/0
r3.00-00                   195   0x00000005  0x2705     320    0/0/0
r4.00-00                   195   0x00000005  0x7cb8     360    0/0/0
r5.00-00                   228   0x00000005  0x2e7d     389    0/0/0
r6.00-00                   228   0x00000005  0xc9ec     345    0/0/0
r7.00-00                   228   0x00000005  0xe1c4     357    0/0/0
r8.00-00                   261   0x00000073  0x9712    1129    0/0/0
r9.00-00                   294   0x00000006  0xec5d     538    0/0/0
r10.00-00                  295   0x00000073  0x992a    1152    0/0/0
r11.00-00                  229   0x00000073  0x8bf8    1148    0/0/0
r0.00-00                   195   0x00000005  0x8c7e     407    0/0/0
    12 LSPs

r2#
```

Figure 41: r2 link-state database

4.2.3.12 Flooding Information, r6

```
r6# show openfabric flooding
Area 1:
Flooding information for r1.00-00
    Last received on: eth3 (00:05:33 ago)
    RF:
        r10
    DNR:
        r9
        r8

Flooding information for r2.00-00
    Last received on: eth4 (00:04:47 ago)
    RF:
        r10
        r4
    DNR:
        r9
        r8
```

Figure 42: Flooding Table Management from r6

4.2.3.13 Flooding Information, r2

```
r2# show openfabric flooding
Area 1:
Flooding information for r1.00-00
    Last received on: eth2 (00:08:01 ago)
    Received as circuit-scoped LSP, so not flooded.

Flooding information for r2.00-00
    Last received on: (null) (00:07:15 ago)
    RF:
        r0
        r7
        r6
    DNR:
        r1

Flooding information for r3.00-00
    Last received on: eth3 (00:08:06 ago)
    RF:
        r6
    DNR:

Flooding information for r4.00-00
    Last received on: eth1 (00:07:31 ago)
    RF:
        r7
    DNR:

Flooding information for r5.00-00
    Last received on: eth1 (00:07:14 ago)
    RF:
    DNR:

Flooding information for r6.00-00
    Last received on: eth1 (00:07:40 ago)
    Received as circuit-scoped LSP, so not flooded.

Flooding information for r7.00-00
    Last received on: eth0 (00:07:46 ago)
    Received as circuit-scoped LSP, so not flooded.

Flooding information for r8.00-00
    Last received on: eth1 (00:00:23 ago)
    Received as circuit-scoped LSP, so not flooded.

Flooding information for r9.00-00
    Last received on: eth0 (00:04:34 ago)
    Received as circuit-scoped LSP, so not flooded.

Flooding information for r10.00-00
    Last received on: eth1 (00:00:25 ago)
    Received as circuit-scoped LSP, so not flooded.

Flooding information for r11.00-00
    Last received on: eth0 (00:00:20 ago)
    Received as circuit-scoped LSP, so not flooded.
```

Figure 43: Flooding Table Management from r2

According to section 4.2.3.12 and 4.2.3.13 it could be seen how the flooding tables used by the OpenFabric routing protocol to manage traffic and route convergence. The optimized flooding mechanisms reduces the amount of information carried through the network unlike in IS-IS network. It could be seen from both scenarios, how RF and DNR tables are used to manage LSPs distribution throughout the network.

4.2.4 Nodes Reachability

4.2.4.1 Traceroute n1 to s1

Again, this was implemented with the two-node tool. Figure 44 shows *s1* is reachable via the chain route, *r1-r4-r5-r10-sw1*. This was resolved as the best path available at the time with all links maintaining same default metric value of 10 across board.

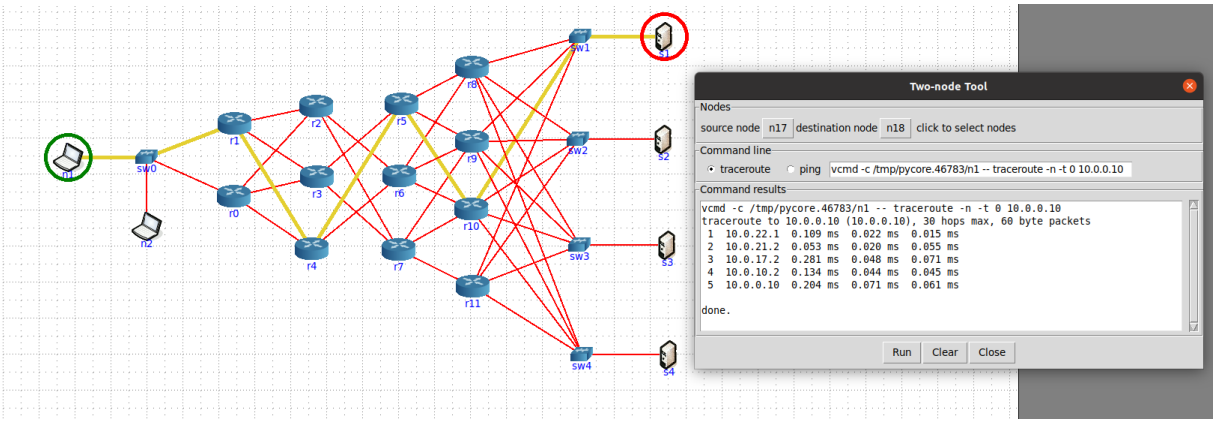


Figure 44: Nodes reachability check with traceroute; n1 to s1

A further tweak to the link bandwidth connecting *r5* and *r10* shows that routing protocol did not consider the delay metric which could have made it possible for a different route to be selected from *r5* instead such as *r5-r8-sw1*, or *r5-r9-sw1* to be able to still reach *s1*. It is not established if this behavior extends to the behavior of the protocol in a production environment like a datacenter.

4.2.4.2 Traceroute n1 to s3

Similar observation from section 4.2.4.1, was considered to check for reachability between *n1* and *s3*. At the start, Figure 45 shows the best route selected by the protocol to reach *s3*.

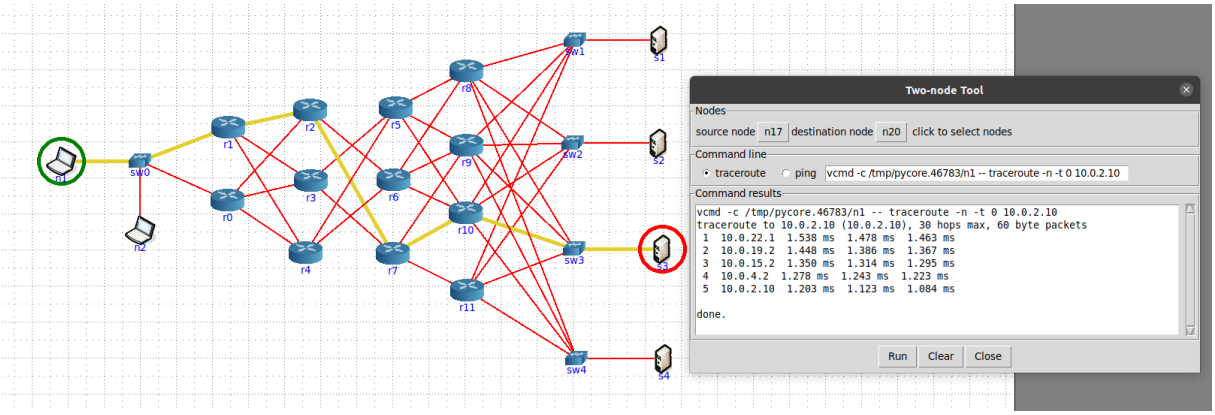


Figure 45: Nodes reachability check with traceroute; n1 to s3

A further tweak to the links metrics for the selected route from Figure 45, *r1-r2-r7-r10-sw3,* was implemented by adjusting the default metric value from 10 to 30 for all the interface contributing to the selected route chain as follows:

```
1    openfabric metric 30
```

The *openfabric metric 30* attributes introduced per interface for the route *r1-r2-r7-r10-sw3,* forced the protocol select a different route to reach *s3* from *n1*. This further confirms how the default metric attribute is regarded within the network. Figure 46 shows the different route negotiated to reach s3 in this case.

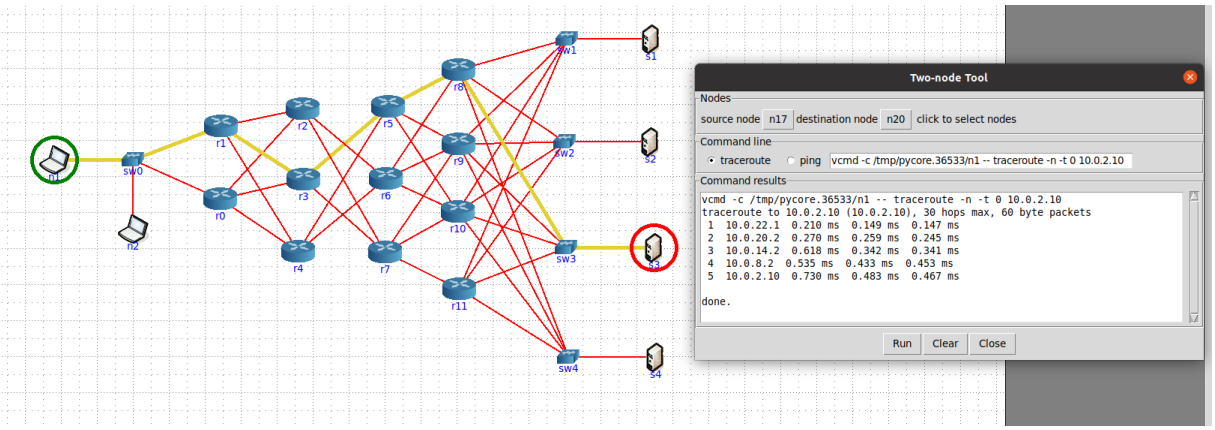


Figure 46: Nodes reachability check with traceroute; n1 to s3 (default metric set to 30)

4.2.5 Packet Capture

4.2.5.1 Hello Packet

The Figure 47 shows the Wireshark capture indicating IS-IS hello packets distributed by *r5* and *r10*. This further confirms that the OpenFabric routing protocol is a revised version of the IS-IS routing protocol, specifically designed to implement high availability and optimized efficiency. ISO 10589 framework identified in the network packet data further underlines this.

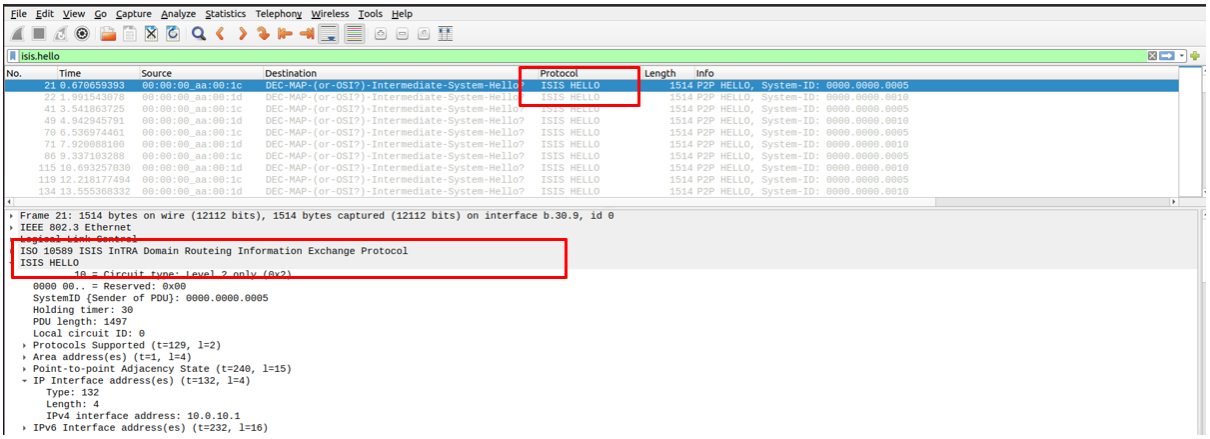


Figure 47: OpenFabric Network Hello Packets

4.2.5.2 HTTP GET Request, n1 to s2

Again, to further utilize the network, an HTTP server was simultaneously deployed for all the servers used in this network, *s1-s4*. The Wireshark capture from Figure 48 shows the network activity during the HTTP GET request.

The below sample bash script was implemented on *n1* and *n2* to allow for simultaneous communication with the servers.

```
#!/bin/bash

  while true; do
      curl -o /tmp/test-$(date +%s).txt 10.0.0.10/example.txt
      curl -o /tmp/test-$(date +%s).txt 10.0.2.10/example.txt
      sleep 5
  done

```

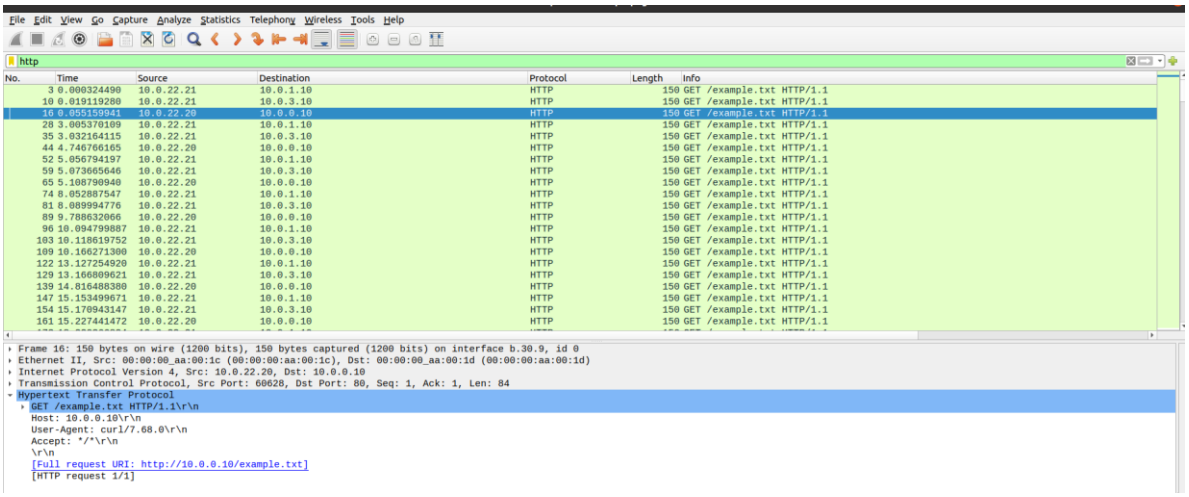


Figure 48: HTTP traffic captured with Wireshark

Again, each node continuously downloads *example.txt* file from the servers and save them with a new name until the process is terminated. The download process could be initiated by running the below commands from *n1* and *n2* via the bash terminal.

```
chmod 777 startatd.sh
./startatd.sh
```

The communication between the servers were emulated as given below:

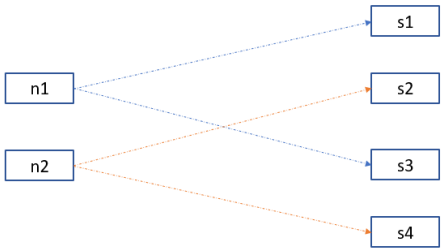


Figure 49: n1 and n2 communication flowchart with s1- s4

The Figure 50 shows the snapshot of the files downloaded from the server during the experiment.

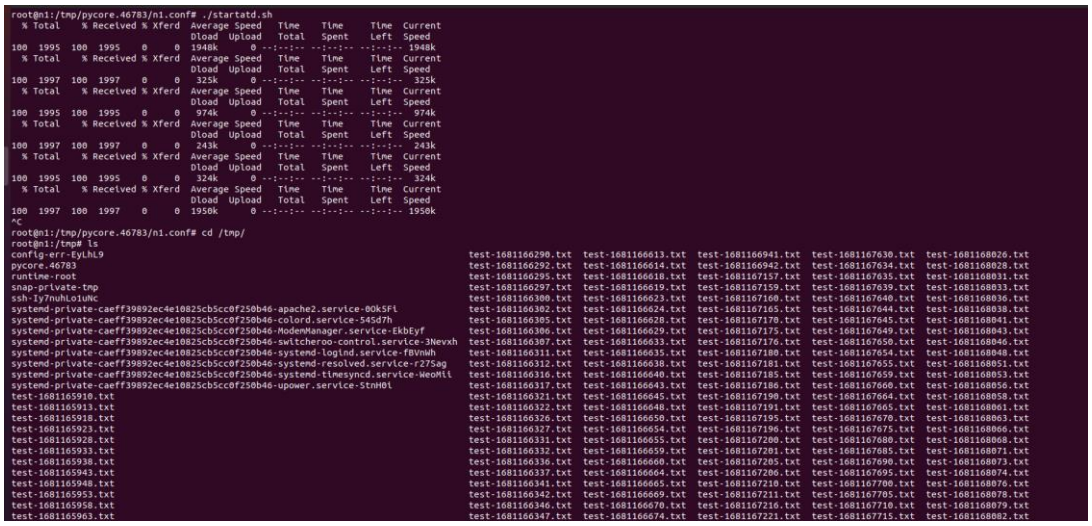


Figure 50: n1 snapshot showing downloaded files

5 Summary and Perspectives

The CORE network emulator allowed me to create and simulate a network environment that closely mirrored the real-world scenario. The user-friendly GUI made it simple to create and configure network topologies and test routing protocols and configurations. The ability to save and restore network topologies also made it easy to quickly iterate on different configurations and test scenarios.

The FRRouting package provided a robust and flexible set of routing protocols and tools that allowed me to easily configure and manage complex network topologies. Additionally, the extensive documentation and online community made it easy to find solutions to any issues I encountered.

Overall, the combination of FRRouting and CORE network emulator made it possible for me to design and test a complex network topology. The project was successfully implemented and tested, demonstrating the practical application of these tools I would highly recommend both tools for anyone working on networking projects or looking to improve their networking skills.

It was also observed that the IS-IS and OpenFabric routing are not entriely different protocols. Both protocols could easily co-exist within the same network without need for specific redistribution. Openfabric uses a subset of the atttibutes already defined for the IS-IS framework except for few adjustments while maintaining a core part of Link-state distribution and complete view of the network as IS-IS.

6 References

1. Arinze, O. (2023). Rinzx/r118349. [online] GitHub. Available at: https://github.com/Rinzx/r118349

2. Cisco Press (2002). Integrated IS-IS Routing Protocol Concepts > IS-IS Routing Domain | Cisco Press. [online] www.ciscopress.com. Available at: https://www.ciscopress.com/articles/article.asp?p=26850.

3. coreemu (n.d.). CORE Documentation. [online] core. Available at: https://coreemu.github.io/core/.

4. FRRouting Project (2017). ISIS — FRR latest documentation. [online] docs.frrouting.org. Available at: http://docs.frrouting.org/en/latest/isisd.html.

5. FRRouting Project (2017). OpenFabric — FRR latest documentation. [online] docs.frrouting.org. Available at: http://docs.frrouting.org/en/latest/fabricd.html.

6. Shen, N., Ginsberg, L. and Sanjay Thyamagundalu (2018). IS-IS routing for spine-leaf topology. [online] Internet Engineering Task Force. Available at: https://datatracker.ietf.org/doc/draft-shen-isis-spine-leaf-ext/07/.

7. White, R. and Zandi, S. (2018). IS-IS support for openfabric. [online] Internet Engineering Task Force. Available at: https://datatracker.ietf.org/doc/draft-white-openfabric/07/.

8. ISO (2002). ISO/IEC 10589:2002 (Information technology — Telecommunications and information exchange between systems — Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)). [online] ISO. Available at: https://www.iso.org/standard/30932.html.

7 Appendix

1. Virtual Machine for Project is available at this link: Project.zip
2. Network Topologies deployed for the project are available at GitHub: Network Topology

Attached CD/DVD content