Enhancing Performance of Conventional Computer Networks Employing Selected SDN Principles

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD)

Electronic and Computer Engineering

College of Engineering, Design and Physical Sciences

Brunel University London

United Kingdom of Great Britain and Northern Ireland

By

Hasanein Hasan

Supervised by Professor John Cosmas

December 2016

Abstract

This research is related to computer networks. In this thesis, three main issues are addressed which affect the performance of any computer network: congestion, efficient resources utilization and link failure. Those issues are related to each other in many situations. Many approaches have been suggested to deal with those issues as well as many solutions were applied. Despite all the improvements of the technology and the proposed solutions, those issues continue to be a burden on the system's performance. This effect is related to the increase of the Quality of Service (QoS) requirements in modern networks.

The basic idea of this research is evolving the intelligence of a conventional computer network when dealing with those issues by adding some features of the Software Defined Networking (SDN). This adoption upgrades the conventional computer network system to be more dynamic and higher self-organizing when dealing with those issues.

This idea is applied on a system represented by a computer network that uses the Open Shortest Path First (OSPF) routing protocol. The first improvement deals with the distribution of Internet Protocol (IP) routed flows. The second improvement deals with tunnel establishment that serves Multi-Protocol Label Switching (MPLS) routed flows and the third improvement deals with bandwidth reservation when applying network restoration represented by Fast Re-route (FRR) mechanism to sooth the effect of link failure in OSPF/MPLS routed network.

This idea is also applied on another system that uses the Enhanced Interior Gateway Routing Protocol (EIGRP) to improve the performance of its routing algorithm.

Adopting the SDN notion is achieved by adding an intelligent controller to the system and creating a dialog of messages between the controller and the conventional routers. This requires upgrading the routers to respond to the new modified system.

Our proposed approaches are presented with simulations of different configurations which produce fine results.

Table of Contents

List of Figures

List of Tables

Abbreviations of General Expressions

ABR	Area Border Routers
AP	Access Point
AS	Autonomous system
ASBR	Autonomous System Boundary Routers
BBR	Bandwidth-Based Routing
BDR	Backup Designated Router
BGP	Border Gateway protocol
BN	Border Node
BPCA	Backup Paths Calculation Algorithm
CBR	Constraint Based Routing
COTS	Connection-Oriented Transport Service
CSMA/CA	Carrier Sense Multiple Access/ Collision Avoidance
CSPF	Constrained Shortest Path First
DCE	Data Circuit-terminating Equipment
DiffServ	Differentiated Services
DNS	Domain Name System
DoD	Department of Defence
DR	Designated Router
DTE	Data Terminal Equipment
ECMP	Equal Cost Multi-Paths
EGP	Exterior Gateway Routing Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
ERO	Explicit Route Object
FEC	Forwarding Equivalence Class
FEP	Fast Emergency Path
FIFO	First Input First Output
FRR	Fast Re-route
HDLC	High Level Datalink Control
IGP	Interior Gateway Routing Protocol
IntServ	Integrated Services
IP	Internet Protocol
IS-IS	Intermediate System to Intermediate System
LAN	Local Area Network
LDP	Label Distribution Protocol
LFIB	Label Forwarding Information Base
LSA	Link State Advertisement
LSP	Label-Switched Path
LSR	Label Switching Router
LSU	Link State Update
MAC	Media Access Control
MAN	Metropolitan Area Network
MIBs	Management Information Base
MIRA	Minimum Interference Routing Algorithm
MP	Merge Point
MPLS	Multi-Protocol Label Switching

NED	Network Description language
NHOP	Next-Hop Backup Tunnels
NNHOP	Next Next-Hop Backup Tunnels
NP	Network Performance
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OVS	Open v Switch
PAN	Personal Area Network
PDV	Packets Delay Variation
PLR	Point of Local Repair
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
PSB	Path State control Block
QoS	Quality of service
RED	Random Early Detection
RIB	Routing Information Base
RIP	Routing Information Protocol
RRO	Record Route Object
RSB	Reservation State control Block
RSVP	Resource Reservation Protocol
RTO	Retransmission Time Out
RTP	Reliable Transport Protocol
SDN	Software-Defined Networks
SF	Safety Factor
SNMP	Simple Network Management Protocol
SPF	Shortest Path First
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TE	Traffic Engineering
TFTP	Trivial File Transfer Protocol
TTL	Time to Live
UDP	User Datagram Protocol
VC	Videoconferencing
VM	Virtual Machine
VOIP	Voice over Internet Protocol
VPN	Virtual Private Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WRED	Weighted Random Early Detection
WSN	Wireless Sensor Network
XML	Extensible Mark-up Language

Acknowledgement

I would like to express the deepest appreciation to ***Professor John Cosmas*** *who has supervised me over the last four years. His guidance and his encouragement were the base for this work.*

Also it has been a privilege to work with my second supervisor ***Dr Tatiana Kalganova*** *who I sincerely grateful to her.*

Furthermore, I would like to thank my family that represented by my three old aunts Rabiaa (Zahra), Mona and Hanaa for their encouragement and prayers.

Finally, I would like to acknowledge the financial, academic and technical support of ***Brunel University*** *with special gratitude to the staff in the* ***Student Centre****, the* ***Post-Graduate Research Office****, the* ***Library****, the* ***Accommodation Office*** *and the* ***Residences Office****.*

Declaration

It is hereby declared that the thesis in focus is the author's own work and is submitted for the first time to the Post-Graduate Research Office. The study was originated, composed and reviewed by the mentioned author and supervisors in the department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences, Brunel University London/United Kingdom. All the information derived from other works has been properly referenced and acknowledged.

Hasanein Hasan

London/ United Kingdom

Submitted 12 December 2016

To

London,

The greatest city in the

world

Chapter 1

Thesis Introduction

Introduction to Chapter 1

This chapter gives a brief background to the reason behind this research. It explains the major challenges that facing some of the technologies used on the contemporary computer networks. It shows the main contributions of the research. It gives general description to the methodology of the research. Moreover, it summarises the later chapters of this thesis.

1.1 Research Motivations

During the last 20 years, the Internet has continuously expanded and there is no evidence that the expansion is slowing [1]. The gigantic growth of the Internet and the intranet networks creates a series challenge to service providers and equipment suppliers in terms of enormous escalation in traffic. This growth involves expanding the existing networks in terms of technology, services and coverage area.

Similar to any working system in the world, the development of the computer networks is escorted with challenges as the performance is escorted with problems. This research represents an attempt to identify some of those challenges and problems and to apply solutions to deal with them by expanding the available capabilities. It is an endeavour to improve the performance of a computer network system by managing the forwarding devices (the routers) through controlling their models within the OSI layers except the physical layer. Performance management is the act of monitoring and maintaining the whole system [2].

Some of the outcomes of improvement that applied on the network system performance are obtained by measuring the Quality of Service (QoS) that sensed by the end-user. QoS depends upon several factors in addition to the performance of the computer network [3] pp 566. The modified system is more efficient and more reliable.

Our research improves the performance of the following technologies that utilized in the conventional computer networks:

- Open Shortest Path First (OSPF) routing protocol.
- Multiprotocol Label Switching (MPLS) forwarding technology.
- Fast Reroute (FRR) link protection mechanism.
- Enhanced Interior Gateway Routing Protocol (EIGRP).

Our research develops all those technologies by adopting the Software Defined Networking (SDN) notion. However, none of the SDN protocols or devices is used. Adopting the SDN principle is represented by adding a controller (as a separate device) to the network in addition to establishing a new messaging system to exchange the necessary information between the controller and the routers. The added controller is compatible with the network as its architecture design is according to the OSI model. The controller has unique features and capabilities. It enhances the performance of the conventional network by making it more robust when dealing with those problems.

1.2 Research Challenges

Some of the main problems that negatively affect the performance of any conventional networking system are:

- The network congestion problem.
- The efficiency of bandwidth exploitation problem.
- The MPLS tunnel restoration time problem.
- The network failure problem.

The degradation of the Network Performance (NP) reduces the Quality of Service (QoS) of the network. Referring to the network congestion problem, congestion is caused either by uneven distribution of traffic among the available paths or lack of hardware infrastructure of the node or the link itself [4]. In addition to that, there is a lack of cooperation between routing and congestion control [5]. Most of the treatment efforts are applied locally at routers, which involve flooding the updates of the routing tables across the network continuously to inform all routers with real time image of flows distribution across the network to provide the best possible flows distribution to avoid the congestion [5-7].

The MPLS tunnel restoration time problem is related to both of the congestion problem and the link failure problem. It is the duty of the head-end router to deal with it [8]. According to [9], the MPLS tunnel restoration time takes from **2-3** seconds at least. This is considered a long time of outage in contemporary computer networks. The proposed algorithms that are used to dynamically establish MPLS tunnels are applied at the ingress of the tunnel (the head-end router) [10-12]. This involves coordination among the routers to achieve better bandwidth utilization.

The network failure reduces the available resources of the network. When a link or node failure event happens in a routed network, there is unavoidable period of disruption to the delivery of traffic. The convergence time of OSPF/IP routed network is less than that of the OSPF/MPLS routed network. However, the network failure still affects the network performance negatively.

1.3 Research Contributions

The contribution of this research is to enhance the performance of the OSPF/MPLS network by adding some features of the Software Defined Networking (SDN) in a modified approach to improve some network's performance metrics like link congestion, packet loss, packet delay variations and total network delay. The first contribution of this research is involving the controller in monitoring flows distribution in real time and intruding to deal with the link congestion issues.

The basic idea behind this work is to create a temporary flow table in the network layer of some routers in addition to the existing OSPF routing table. Those routers use the flow table in forwarding specific flows instead of the OSPF routing table. This diverts selected flows apart from their OSPF main paths to other paths to avoid congestion and to achieve better usage of network resources. The flow table is almost smaller than the routing table and deals with specific flows. The controller creates or removes the flow table according to network real time requirements within specific standards.

The second contribution is involving the controller in creating the primary MPLS tunnel, re-adjust its bandwidth instantaneously and accurately, change its path during very short time or remove the MPLS tunnel (if required to do so). The controller performs all those operations according to FDT and FDA of **chapter 4** which behave to fulfil the real time requirements of the network.

The third contribution of this research is represented by involving the controller to soothe the effect of link failure as much as possible. The contribution goes under two different scenarios:

- **Soothing the Effect of Link Failure in OSPF/IP Routed Network**

  The controller is only involved with congestion testing after link failure recovery in terms of IP routed packets.
- **Soothing the Effect of Link Failure in OSPF/MPLS Routed Network**

  The controller protects the primary MPLS tunnels by establishing dynamic backup tunnels taking the bandwidth sharing issue into consideration.

1.4 Research Methodology

The methodology followed in our research is:

- Viewing an existing technology that is currently used on recent computer networks.
- Illustrating one or more of the problems that degrades the performance of the computer networks and has a relationship with the illustrated technology.
- Suggesting a solution.
- Explaining theoretically how our proposed solution will improve the performance of the computer networks regarding the existing technology as well as the privileges of our solution compared with other solutions.
- Implementing the solution on a computer network system.
- Comparing the results obtained from our solution with those obtained either by experiment or published regarding the existing technology.
- The positive difference in results represents the improvement that our solution added to the system.

1.5 Thesis Organization

In addition to this introductory chapter, our thesis consists of six scientific chapters commencing at **chapter 2** and ending at **chapter 7**. Each chapter is structured independently. Conceptually, the chapters are inter-dependent and the reader should follow the right order in order to better understand the contributions presented in the thesis. The contents of chapters are discussed briefly:

- **Chapter 1: Thesis Introduction**

  This chapter gives a brief background to the reason behind this research and general description to the methodology of the research.

- **Chapter 2: Networking Concepts, Literature Review**

  This chapter presents a brief literature review to computer networks. It illustrates some of the technologies used in the computer networks as well as some of the problems facing those technologies.

- **Chapter 3: OSPF, SDN OpenFlow and Project Basic Design**

  This chapter discusses the basics architecture of our improved network. It shows the added models as well as the preliminary establishments of the project that enable the improvements added on the following chapters.

- **Chapter 4: Congestion Avoidance in OSPF Network by Adopting the SDN Notion**

  This chapter represents the first contribution of our research. It shows the rule of the controller that represented by monitoring flows distribution in real time and dealing with the congestion issue in OSPF/IP routed network.

- **Chapter 5: Improvement of MPLS Technology by Adopting the SDN Notion**

  This chapter represents the second contribution of our research. It shows the rule of the controller that represented by calculating and establishing the primary MPLS tunnels in OSPF/MPLS routed network.

- **Chapter 6: Soothing the Effect of Link Failure by Adopting the SDN Notion**

  This chapter represents the third contribution of our research. It shows the rule of the controller that represented by soothing the effect of link failure in OSPF/IP routed network as well as calculating and establishing the backup MPLS tunnels in OSPF/MPLS routed network.

- **Chapter 7: Improvement of Performance of the EIGRP Algorithm by Adopting the SDN Notion**

  This chapter shows the rule of the controller that represented by monitoring flows distribution in real time and dealing with the congestion issue in EIGRP routed network.

- **Chapter 8: Conclusions and Further Work**

  This chapter discusses the conclusions obtained from our research, the obstacles that may prevent from applying it, the encouragements of applying it on real world and the proposed future work to develop it.

For anyone who reads this thesis and they are not familiar with computer network technologies, we advise reading **chapter 2** as it gives a brief a literature review about computer networks. **Chapters 3, 4, 5** and **6** are related to each other. **Chapter 7** represents different implementation. However, all chapters represent development efforts to the technologies mentioned in **chapter 2**.

Chapter 2

Networking Concepts, Literature Review

Introduction to Chapter 2

This chapter presents a brief literature review to computer networks. It illustrates the architecture of the computer networks where the reader finds some basic definitions to several concepts. It illustrates some of the technologies that are currently used in the computer networks. It also discusses the basics of routing protocols and the routing technologies.

This chapter shows some of the problems facing the computer networks as well as some of their proposed solutions. Many of the concepts and definitions of this chapter are discussed in details or developed later in the following chapters.

2.1 Basic Definition

A computer network consists of computers and some other networking devices that are connected together via communication channels to provide communication and resource-sharing among a range of users [13]. Building modern data communications networks involves using standardised network components, interfaces and protocols based on digital line transmission, packet switching and layered communications protocols [3] pp 67.

Network protocols are a set of rules by which all the networks should abide to provide effective communication among their devices. The Open Systems Interconnection (OSI) model represents the basic architectural model for networks [14] pp 11.

2.1.1 Computers

Personal Computer (PC) and a mainframe computer represent Data Terminal Equipment (DTE) and they are connected to the network by means of by means of Data Circuit-terminating Equipment (DCE) [3] pp 67. The modem is an example of DCE. Fig. 2-1 shows the components and interfaces making up a simple data network.

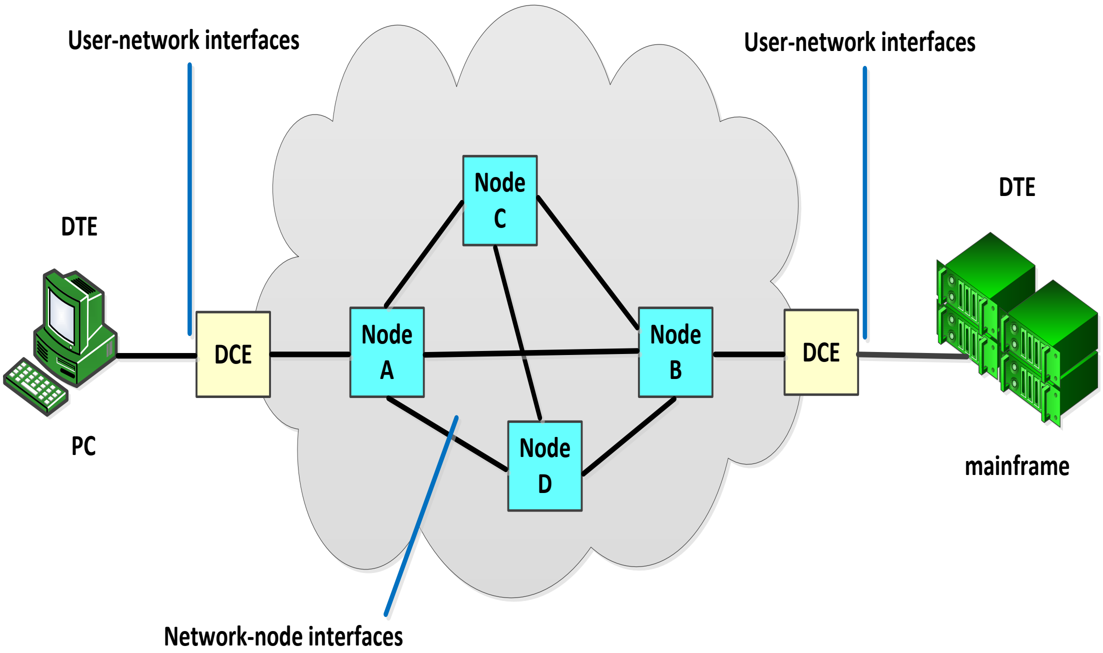


Fig. 2-1 Components and interfaces making up a simple data network [3] pp 68

A node in Fig 2-1 can be a switch, a router, a LAN hub, a multiplexor or some other kind of exchange.

2.1.2 Networking Devices

Networking devices may include routers, network bridges, switches, hubs, etc. The networking devices are responsible of forwarding the packet to the appropriate next node nearer the destination according to the address on each packet (without considering the contents) [3] pp 2 - 3.

The computer network can be a simple network or an arbitrary collection of networks interconnected to provide some sort of host-to-host packet delivery service, which is called *internetwork*. An *internetwork* is made up of lots of smaller networks. Therefore, it is often referred to as a *network of networks* [15] pp 203 – 204.

Fig. 2-2 shows an example *internetwork* where there are multiple single-technology networks like Ethernets, a wireless network and a point-to-point link. Those simple networks are interconnected via routers.

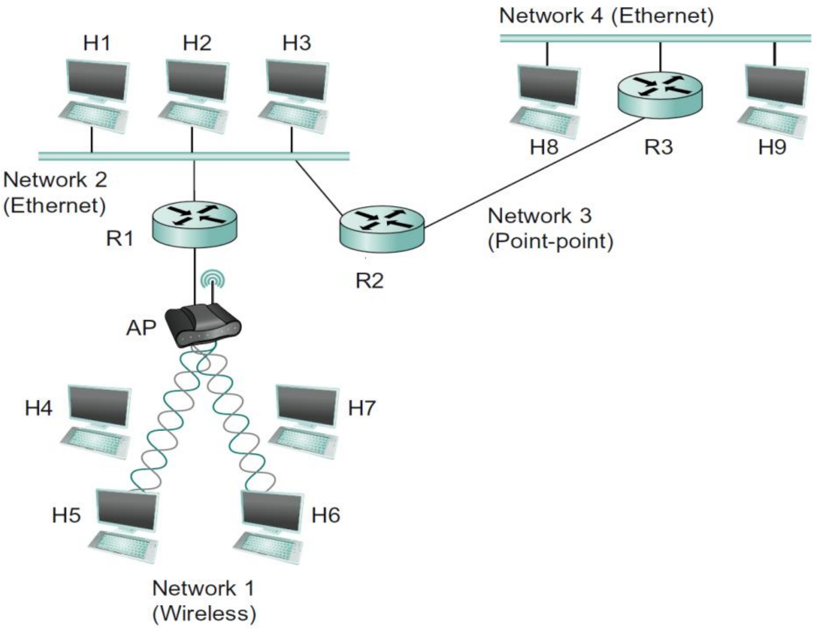


Fig. 2-2 A simple internetwork, Hn =host; Rn =router [15] pp 204

2.2 The OSI Model

The Open Systems Interconnection (OSI) reference model is a conceptual model that is used to create and implement applications that run on a network [14] pp 13. It partitions the various data communications functions into seven independent but interacting layers. The layers interact in a peer-to-peer manner [3] pp 13-14.

The layers of the OSI model are as shown in Fig. 2-3:

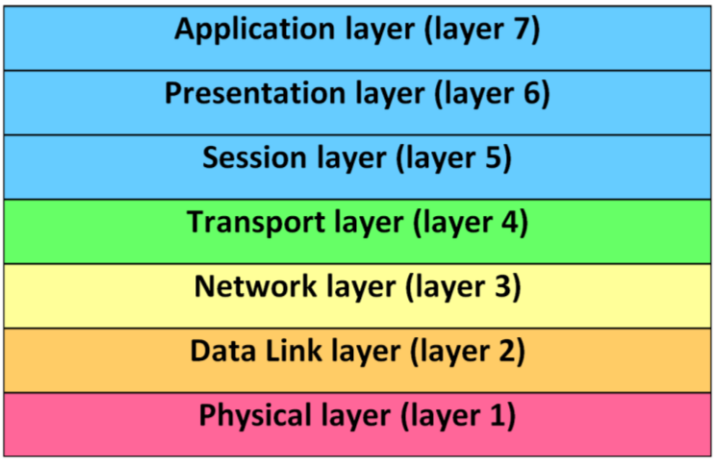

Application layer (layer 7)
Presentation layer (layer 6)
Session layer (layer 5)
Transport layer (layer 4)
Network layer (layer 3)
Data Link layer (layer 2)
Physical layer (layer 1)

Fig. 2-3 Open Systems Interconnection (OSI) model [14] pp13

As none of the three upper layers are concerned about networking or network addresses, only the four bottom layers are used to define how data is transferred over a physical wire or through switches and routers [14] pp 14.

2.2.1 Physical Layer (layer 1)

The Physical layer deals with the *medium* itself by defining the precise electrical, interface and other aspects related to the particular communications *medium* [3] pp 17. It activates, maintains, and deactivates a physical link between end systems [14] pp 30. Physical layer is also responsible of sending and receiving bits which come only in values of ***0*** or ***1*** [14] pp 30.

2.2.2 Data Link Layer (layer 2)

The Data Link layer provides dependable passing of data across a physical network link. It translates messages from the Network layer into bits for the Physical layer to transmit [14] pp 24. The Data Link layer ensures that messages are delivered to the proper device on a Local Area Network (LAN) using hardware addresses [14] pp 24. Various Data link layer specifications define several network and protocol features, including network topology, physical addressing, error notification, frames sequencing, and flow control [16].

2.2.3 Network Layer (layer 3)

The Network layer defines the device address and determines the best way to move data. It is responsible of transporting traffic between devices that are not locally attached. The Network layer uses two types of packets: data packets and route updates packets [14] pp 22. Routers are the devices that work under the Network layer as they provide the routing services within an *internetwork*.

2.2.4 Transport Layer (layer 4)

The Transport layer receives data from the session layer then segments and reassembles it into a data stream for transport across the network [14] pp 16. The transport protocols used on the Internet are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) [16].

2.2.5 Session Layer (layer 5)

The Session layer starts, manages, and tears down communication sessions [16]. It controls the dialog control between devices, or nodes [14] pp 16.

2.2.6 Presentation Layer (layer 6)

The Presentation layer presents data to the Application layer. It is responsible of coding and conversion functions which ensure the information sent from the application layer of one system is readable by the application layer of another system [16].

2.2.7 Application Layer (layer 7)

The application layer supplies communications functions services to be convenient for all possible sorts of data transfer, control signals and responses between *cooperating* computers [3] pp 14. It deals with software applications that perform a communicating component. It is also responsible for setting up the intended communication partner and allocating the availability of sufficient resources for the intended communication exist [14] pp 15.

2.3 The DoD Model

The Department of Defence (DoD) model consists of four layers. It represents a summarized version of the OSI model [14] pp 68-69. The layers of the DoD model and their counterparts of the OSI model is as shown in Fig. 2-4:

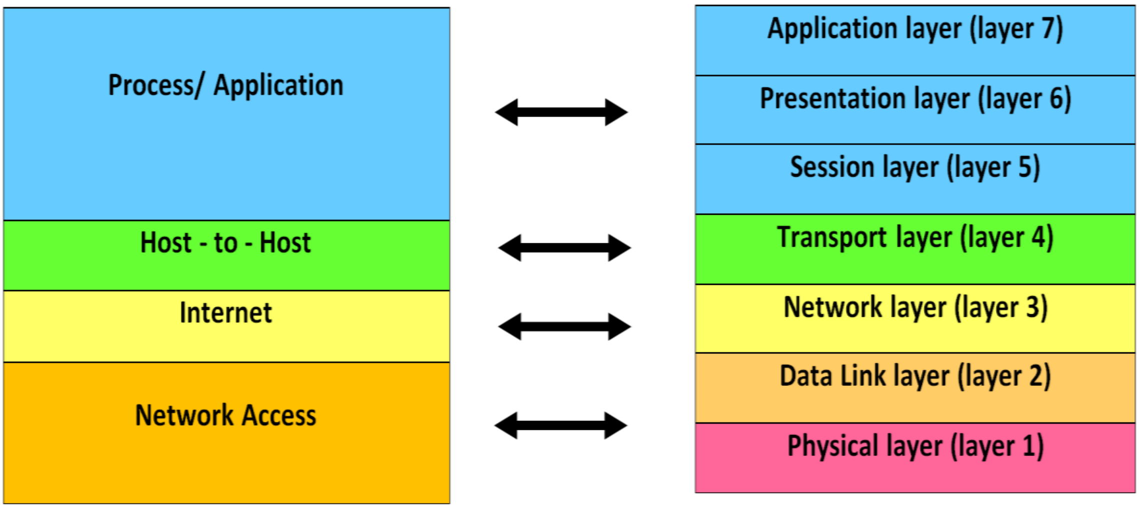


Fig. 2-4 DoD and OSI models [14] pp 69

Both DoD and OSI models represent the same design concept and have similar functions in similar layers [14] pp 69.

2.4 Types of Networks

Sometimes, networks are classified according to their size. The most famous types [17] are:

- Personal Area Network (PANs)
- Local Area Network (LANs)
- Metropolitan Area Network (MANs)
- Wide Area Network (WANs).

2.4.1 Personal Area Networks (PANs)

PAN is deployed inside a small office or residence. It belongs to an individual person and may include one or more computers with several peripheral devices [17].

2.4.2 Local Area Networks (LANs)

The computer network that interconnects computers and devices within a small geographic area is called Local Area Network (LAN). It consists of Layer 1 devices like hubs and repeaters and Layer 2 devices like switches and bridges [3] pp 125.

2.4.3 Metropolitan Area Networks (LANs)

MAN expands across an entire city, college campus or small region. It may connect several LANs together to form a bigger network [17].

2.4.4 Wide Area Networks (WANs)

The computer network that interconnects commuters and devices within a wide geographic area is called Wide Area Network (WAN). It consists of Layer 3 devices (routers) [3] pp 165-166.

The *intranet* is a wide area IP-based network owned by an organization. It uses routers of the same technology and interconnected with the Internet at strictly regulated gateway locations [3] pp 320.

2.5 Internet Protocol Suite

Internet protocol suite is set of open-system (non-proprietary) communications protocols used on the Internet and similar computer networks like LAN and WAN [18]. The protocols of the Internet protocol suite and their corresponding OSI layers are illustrated in Fig. 2-5.

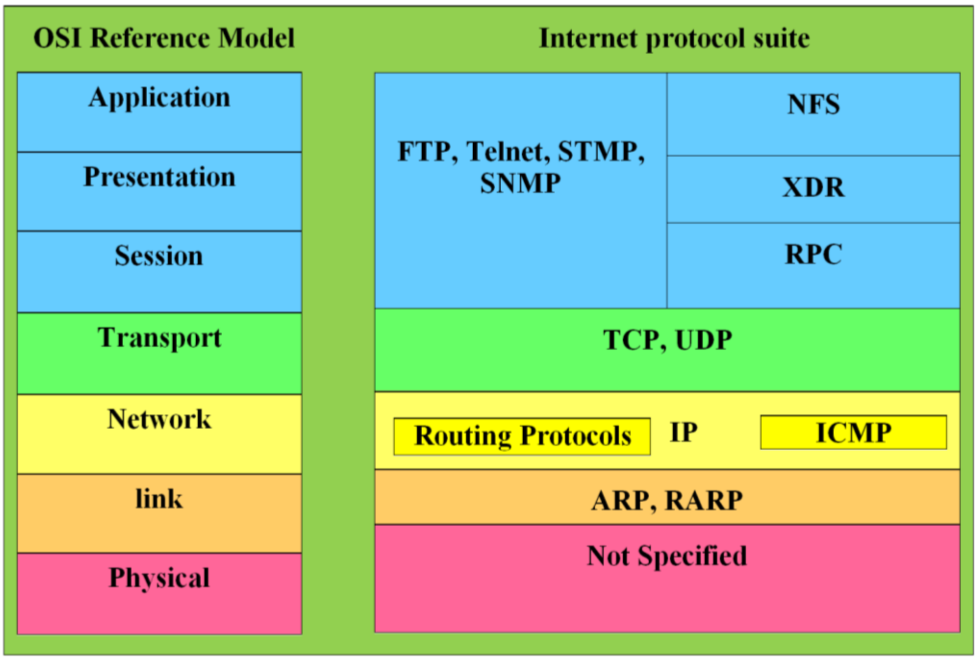


Fig. 2-5 Internet protocols span the complete range of OSI model layers [18]

The Internet protocol suite includes lower-layer protocols (such as TCP and IP). It also specifies popular applications such as electronic mail, file transfer, and terminal emulation.

2.6 Autonomous System (AS)

A network of routers all under the same operational administration is called an autonomous system (AS) [3] pp 224. The single autonomous system identifies its routers as *internal routers* [3] pp 224 or sometimes called *interior routers* [18]. The distribution of routing information between the routers within a single autonomous system involves using Interior Gateway Routing Protocols (IGPs) where routing information are shared among routers and there is no need to hide the routing information or keep it secret. The Border Nodes (BNs) represent the connection points between autonomous systems [3] pp 224. BNs are called Autonomous System Boundary Routers (ASBRs) as well [3] pp 256. Fig. 2-6 shows the BGP network that consists of several autonomous systems while Fig. 2-7 illustrates the ASBR.

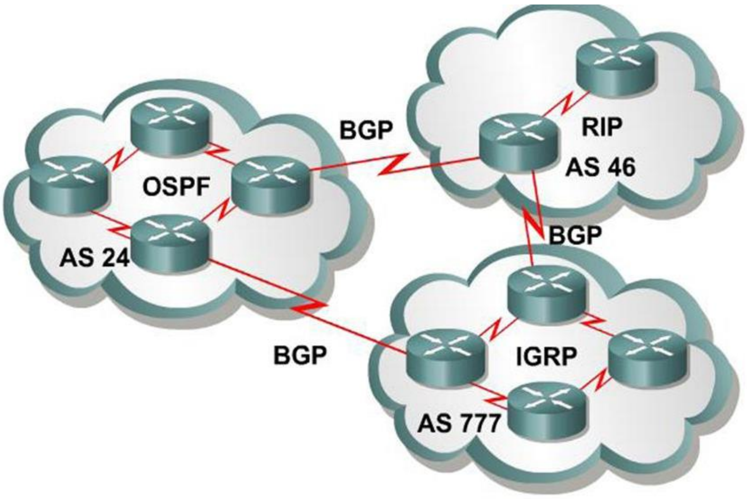


Fig. 2-6 A network that consists of several autonomous systems [19]

The autonomous system may be also called an Administrative Domain [3] pp 224.

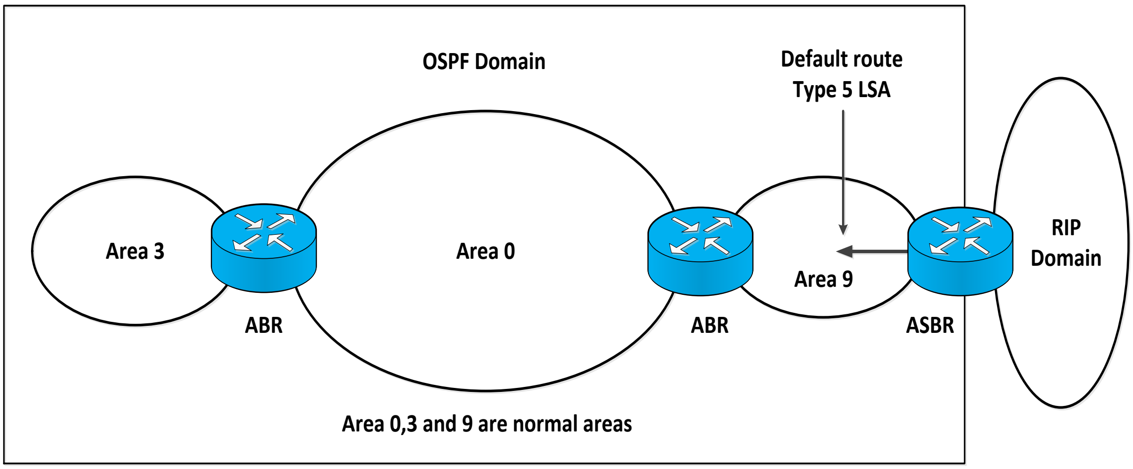


Fig. 2-7 A network where ASBR separates two different domains [20]

2.7 Virtual Private Network (VPN)

A Virtual Private Network (VPN) can be defined as an accessory private intranet network across a public network such as the Internet, creating a secure private connection, essentially through a private tunnel. VPN carries private traffic across the Internet connecting different remote users [21] pp 3. Fig. 2-8 represents fictional virtual private network.

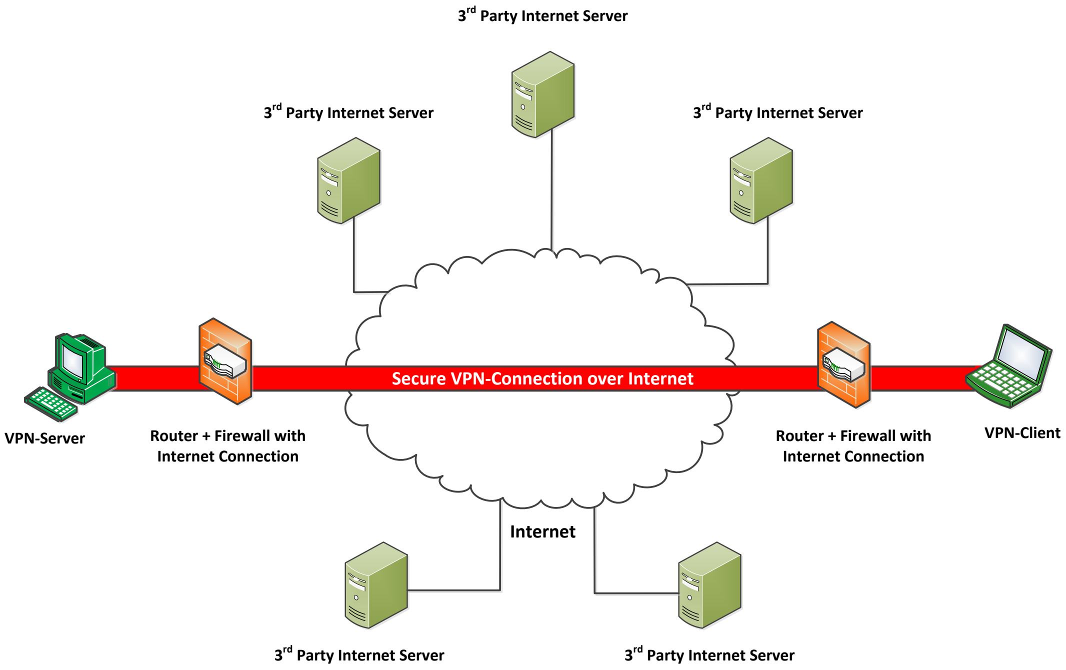

3rd Party Internet Server
3rd Party Internet Server
3rd Party Internet Server
Secure VPN-Connection over Internet
VPN-Server
Router + Firewall with Internet Connection
Router + Firewall with Internet Connection
VPN-Client
Internet
3rd Party Internet Server
3rd Party Internet Server

Fig. 2-8 Virtual Private Network (VPN) [22]

The current VPN technologies can be classified within two categories [18]:

- Trusted VPN technologies — the most promising technologies are the Multi-Protocol Label Switching MPLS-based technologies: MPLS-based L2VPNs and MPLS VPNs using BGP.
- Secure VPN technologies — the most popular technologies are IP Security (IPSec), Layer 2 Tunnelling Protocol (L2TP) or L2TP protected by IPSec, and Point-to-Point Tunnelling Protocol (PPTP).

IPSec uses cryptographic technologies to provide key security services against common security threats on the Internet. It consists of a set of protocols that provides the following security services [18]:

- Authentication ensures that the VPN device contacts the intended entity.
- Confidentiality guarantees the privacy of data by encrypting it.
- Integrity guarantees that the data's content has not been changed during transmission.

2.8 Network Routing

The operation of transferring information across an *internetwork* from a source to a destination defines routing. Routing involves crossing at least one intermediate node along the way. The *internetwork* represents a group of individual networks, linked by intermediate networking devices that behave as a single large network [18].

Routers perform packet forwarding according to their routing tables [3] pp 167. In *static routing*, the routing table is constructed manually by the network operator while in *dynamic routing*; the router is responsible of the creation and maintenance of its routing table. A router collects routing information about the ever-changing topology of the network by means of a *routing protocol*. Thereafter, it calculates the shortest routing *distance* (or route *cost*) to each *reachable* destination and performs dynamic updating of its routing table. Finally, the router always determines the best available route to every reachable destination in routing table calculation [3] pp 216.

In conventional (also called traditional) network architecture, both the control plane and the data plane lie together [23] at the router as shown in Fig. 2-9.

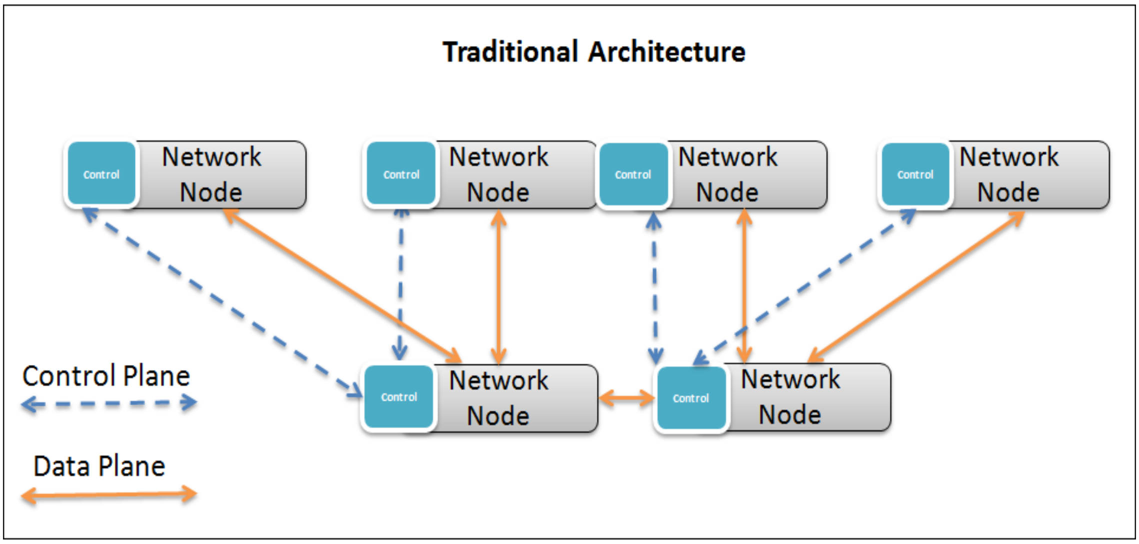


Fig. 2-9 Traditional network architecture [24]

Conventional networks apply network layer (IP) forwarding mechanisms. When a packet traverses the network, each router extracts all the information relevant to forwarding the packet from the network layer header. The extracted information is used in routing table lookup operation to detect the next hop for the packet [25]. Routers along the path repeat this table lookup operation until reaching the destination.

If the network is huge, the table lookup operation becomes complicated and may consume time. The Multi-Protocol Label Switching (MPLS) packet forwarding technology provides the solution to this.

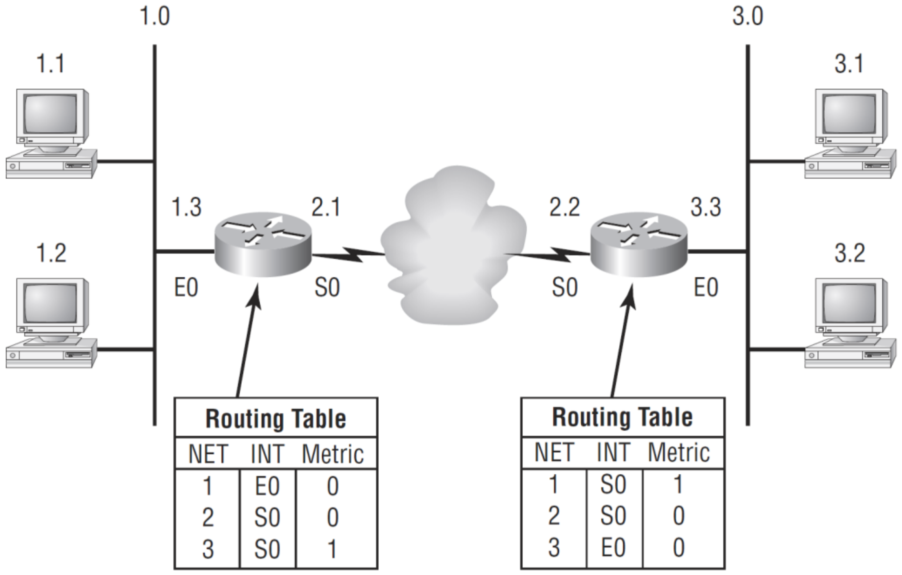
2.8.1 Building the Routing Table

The main considerations while building the routing table [26] are:

- **Administrative distance** which defines the reliability of a routing protocol. Each routing protocol is marked with an administrative distance value .The smaller the administrative distance value, the more reliable the protocol.
- **Metric/ Cost**, each routing protocol uses a different *metric* to calculate the best path to a given destination. The router installs in the routing table the path with the lowest *metric* value.
- **Prefix length.**

2.8.2 Routing Table Structure

Routing table, also called Routing Information Base (RIB), contains the information for routing packets to their *next hop*. It consists of a list of all possible destinations and along with each destination there is the address of next hop (the gateway) and most probably the *cost* or *metric* of the path [3] pp 215. Routing table lies inside the router. Fig. 2-10 shows a routing table that is used in a router of a network.



Routing Table

NET	INT	Metric
1	E0	0
2	S0	0
3	S0	1

Routing Table

NET	INT	Metric
1	S0	1
2	S0	0
3	E0	0

Fig. 2-10 Routing table used in a router [14] pp 22

2.8.3 Convergence Time Definition

The convergence time is the period of time a router spends to build or rebuild its routing table after first being introduced to the network or during its recovery after a topology change (link or node failure) [3] pp 222.

2.9 Routing Protocols

A routing protocol provides the set of rules used by routers when they share routing information to calculate routing tables [14] pp 377. There are two types of routing protocols:

- Interior Gateway Routing Protocols (IGP).
- Border Gateway Protocols (BGP).

2.9.1 Interior Gateway Routing Protocols (IGP)

Interior gateway protocols (IGPs) are used for the sharing of routing information between the routers within a unique autonomous system. Interior Gateway Routing Protocols can be divided into two kinds: Link State Protocols and Distance Vector Protocols [3] pp 224 - 227.

Intermediate System to Intermediate System (IS-IS) and Open Shortest Path First (OSPF) are link state routing protocols while Routing Information Protocol (RIP) and Enhanced Interior Gateway Routing Protocol (EIGRP) are distance vector protocols [3] pp 232.

2.9.2 Border Gateway protocol (BGP)

Border Gateway protocol, also called Exterior Gateway Protocol (EGP) [3] pp 225, is a path-vector protocol that is used to provide for loop-free inter-domain routing and reachability information among autonomous systems (ASs) [3] pp 259 and [27] pp 200. BGP version 4 is the contemporary used version.

To provide successful routing between two different Autonomous Systems (ASs) via BGP, at least one router in each system must be configured to speak BGP.

Each Autonomous System (AS) has a specific number. All reachable destinations are identified according to their AS number, the IP address-ranges associated with the destination AS and the route to the destination AS from the BGP router announcing the path. BGP always selects the single shortest path of fewest intermediate Autonomous Systems (ASs) to the destination. Therefore, it is classified as a Distance Vector routing Protocol [3] pp 259.

2.10 Link State Routing Protocols

Link state routing protocols distribute routing information related to the state of the individual links of the network [3] pp 228. Router builds a complete ‘map’ of the topology of the network from the routing protocol advertisements sent by other routers [3] pp 228.

Router uses its network topology database to calculate the best path route to each individual destination [3] pp 228. Dijkstra’s algorithm, which is also called Shortest Path First (SPF) algorithm, is used in this operation [3] pp 232.

The most important advantage of link state protocols is the amount of network topology detail held in the link state database, as only ‘real’ topology changes need normally be notified by the routing protocol instead of re-broadcasting the whole routing table [3] pp 241.

2.10.1 IS-IS

Intermediate System-Intermediate System (IS-IS) is a link-state Interior Gateway Protocol (IGP). In OSI terminology, a router is referred to as an Intermediate System (IS) [28].

To perform routing within a local area, routers establish Level 1 adjacencies (intra-area routing). Performing routing between Level 1 areas involves routers to establish Level 2 adjacencies (inter-area routing). The default routing behaviour for the routing process will be Level 1-2 if the network administrator does not specify Level 1 or Level 2 routing for the routing process being configured, which is used to connect the inter area routers with the intra area routers [28].

2.10.2 Open Shortest Path First (OSPF)

OSPF is a link state routing protocol that uses the Dijkstra algorithm to build the routing table [14] pp 444. OSPF uses a link *cost* parameter as the basis to calculate the shortest path. A reference bandwidth of 100 Mbps is used for cost calculation [29]. The equation to calculate the *cost* or *metric* value is:

$$Metric = Reference\ bandwidth\ /\ Interface\ bandwidth \qquad (1)$$

i. OSPF features

- Fast convergence [14] pp 445.
- OSPF can divide the routing domain into separate routing areas [3] pp 238.
- OSPF supports load-sharing traffic over Equal Cost Multi-Paths (ECMP) to the same destination (if required) [3] pp 238. The load is split sharply and evenly among the paths [30].

ii. OSPF Traffic Engineering (OSPF-TE)

OSPF-TE is an extension to the standard OSPF protocol. It is commonly associated with MPLS Traffic Engineering which allows using the OSPF routing protocol in MPLS networks [31].

iii. OSPF areas

OSPF routing domain consists of multiple *areas*. The main area is called the *backbone area* (*area 0*) and all other areas are attached to this *backbone* (either directly or by means of a *virtual link*) through connection points called Area Border Routers (ABRs) [3] pp 256. Fig. 2-11 shows the OSPF areas.

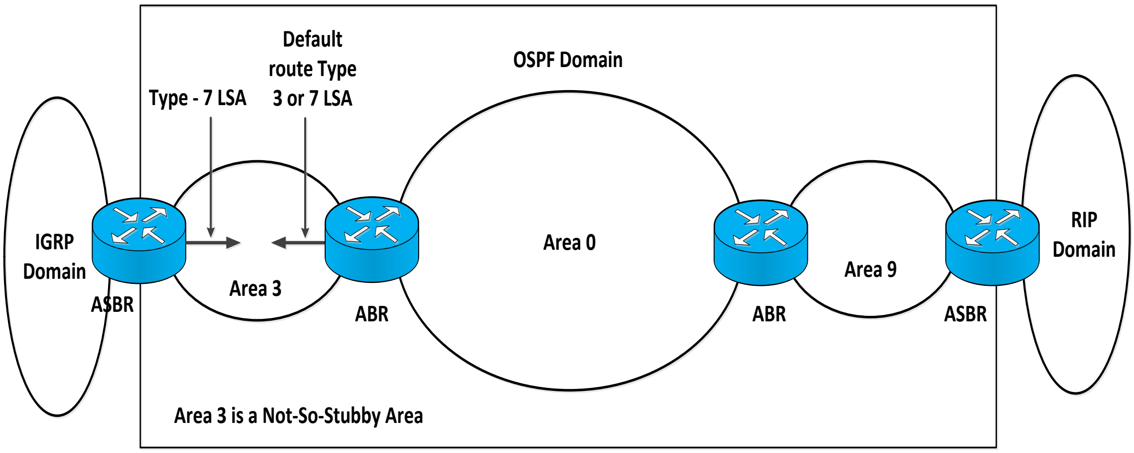


Fig. 2-11 OSPF areas [20]

2.11 Distance Vector Routing Protocols

Distance vector routing protocols work by calculating the direction and the distance from each source router to all possible destinations [3] pp 227.

2.11.1 RIP

Routing Information Protocol (RIP) is the mother of all routing protocols. RIP is a simple distance vector routing protocol [3] pp 230. It operates by determining the shortest path distance as measured in terms of the *hop count* from router to destination. There are two versions of RIP: RIP version 1 (RIP-1) and RIP version 2 (RIP-2) and both are still in use [3] pp 232. RIP is limited by a maximum hop count of 15 in reaching a destination [3] pp 237.

2.11.2 EIGRP

Enhanced IGRP (EIGRP) is a classless, enhanced distance vector protocol, sometimes referred to as EIGRP and is a hybrid routing protocol because it has features of distance-vector and link-state protocols [14] pp 418. Classless means advertisement of subnet information [14] pp 137. EIGRP has the following [18] features:

- Fast convergence.
- Support for variable-length subnet mask.
- Support for partial updates.
- Support for multiple network layer protocols.

The EIGRP-based algorithm depends on both the available bandwidth and the delay. It does not support areas. The EIGRP-based algorithm has been used because it can find several paths to each destination. The route that has the least *metric* value is called the *successor*, and is considered to be the best route to the destination (the *successor* has the highest bandwidth and the lowest delay). The adjacent neighbour routers that have an advertised *metric* less than the *metric* of the current routing table are called *feasible successors* [32]. Theoretically, for every destination there are only one *successor* and up to six *feasible successors* [14] pp 420.

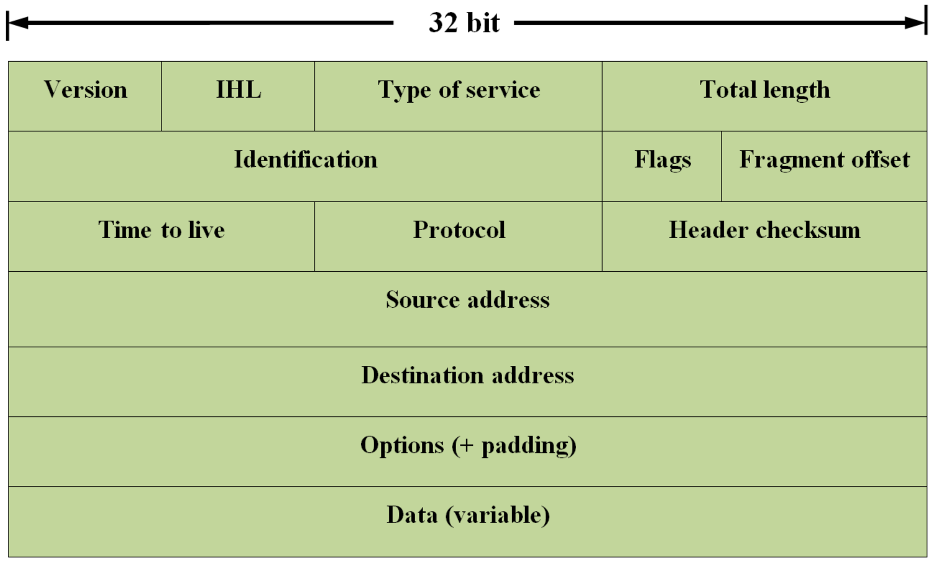
The Enhanced Interior Gateway Routing Protocol (EIGRP) supports unequal cost path load balancing among the *feasible successor* paths depending on the *metric* value of the path. If a path is not a feasible successor, the path is not used in load balancing [33]. Finally, there are three tables that exist in each router: Neighbours table, topology table and routing table [18].

2.12 Routed Protocols

Routed protocols differ from routing protocols. Routed protocols determine the method of packet delivery. They are assigned to an interface. The well-known types of routed protocols are IP and IPv6 [14] pp 328.

The Internet Protocol (IP) is the fundamental network-layer (layer 3) protocol of the internet protocol suite. It is used in end-devices, hosts, which access the Internet and between the routers, of Wide Area Networks (WANs) as well [3] pp 165. It consists of addressing information and holds some control information that enables packets to be routed [18].

An IP packet contains several types of information. Fig. 2-12 shows the frame of the IP packet.



<table>
<tr><td>Version</td><td>IHL</td><td>Type of service</td><td colspan="2">Total length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment offset</td></tr>
<tr><td colspan="2">Time to live</td><td>Protocol</td><td colspan="2">Header checksum</td></tr>
<tr><td colspan="5">Source address</td></tr>
<tr><td colspan="5">Destination address</td></tr>
<tr><td colspan="5">Options (+ padding)</td></tr>
<tr><td colspan="5">Data (variable)</td></tr>
</table>

Fig. 2-12 Fourteen fields comprise an IP packet [18]

2.13 IP Flow Definition

An IP flow is defined as a series of IP packets passing a monitoring point in the network during a specific time interval. All packets belonging to a particular flow have several common properties [34]. Packets belong to a flow share the IP addresses of source and destination [35]. They should satisfy all the defined properties of the flow [34].

In our mathematical model, the values of link loads are assumed to be known with accuracy. However in real networks, the values of traffic data are approximate.

2.14 Time to Live (TTL)

Time to live (TTL), also called hop limit in IP version 6, a field contains an integer binary value corresponding to the duration of time a packet is allowed to *stay alive* inside a network. It also considers the maximum number of hops allowed to be traversed by the packet before it considered *lost* and *undeliverable*. Every time the packet header is processed by a router, the value in the TTL field is checked and decreased by 1 at least. When the TTL value reduces 0, the packet must be dropped [3] pp 181. The TTL field is shown within Fig. 2-12 of IP packet frame.

2.15 Transport Protocols

The transport protocol, layer 4 protocol, is responsible of controlling and managing the end-to-end communication between end devices [3] pp 277. It provides two types of service: the Connection-Oriented Transport Service (COTS) represented in TCP (Transmission Control Protocol) and a Connectionless Transport Service (CLTS) represented in UDP (User Datagram Protocol) [3] pp 278.

2.15.1 User Datagram Protocol (UDP)

UDP is a connectionless transport-layer (Layer 4) protocol. It does not add any reliability, error recovery, or flow-control functions to IP [18]. Removing error-checking makes UDP faster [36]. The UDP segment is as shown in Fig. 2-13.

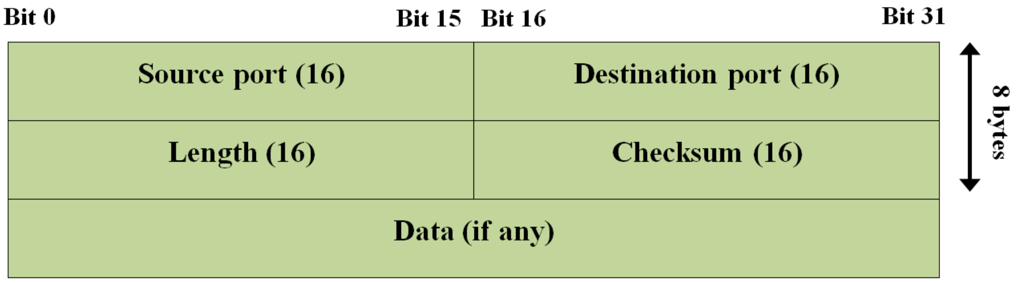


Fig. 2-13 UDP segment [14] pp 78

UDP is used in situations where there is no need for the reliability mechanisms of TCP, such as in cases where a higher-layer protocol might provide error and flow control. The application layer protocols, including Network File System (NFS), Simple Network Management Protocol (SNMP), Domain Name System (DNS), and Trivial File Transfer Protocol (TFTP) use UDP [18].

2.15.2 Transmission Control Protocol (TCP)

TCP (also called TCP/IP) is a connection-oriented transport-layer (Layer 4) protocol. It provides reliable transport of data in an IP environment [3] pp 283. TCP chooses large blocks of information from an application, fractures them into segments and tags each segment with a sequence number [14] pp 75. TCP is slower than UDP. Fig. 2-14 shows the format of TCP segment.

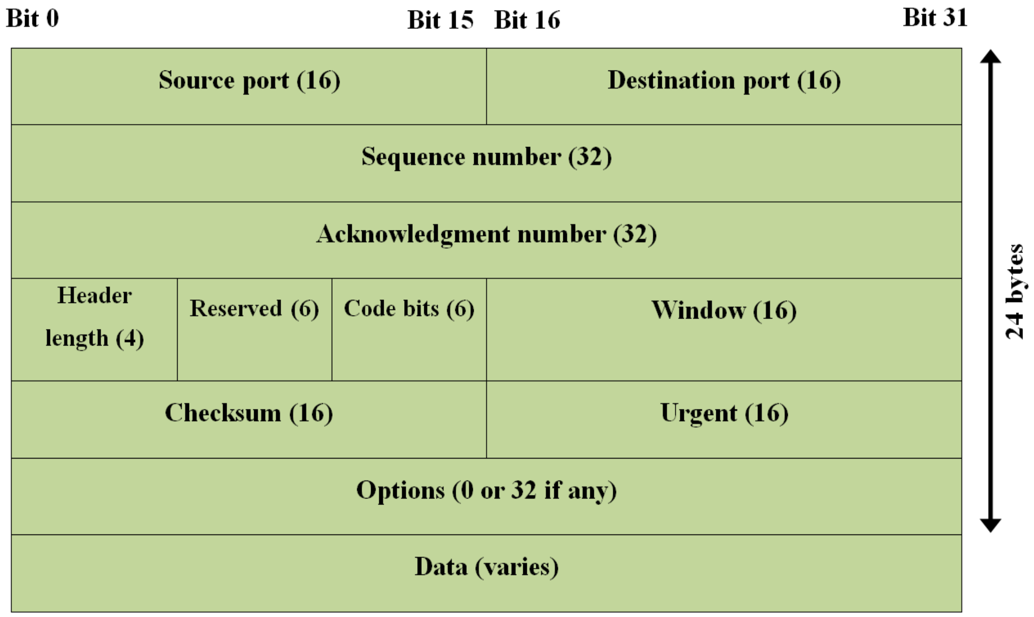


Fig. 2-14 TCP segment format [14] pp75

Connection establishment procedure passes through a three-stage process mechanism performed sequentially. The stages are:

1) *Synchronization:* A connection is initiated between the two hosts.
2) *Data Transfer:* First host (transmitter) starts sending TCP data segments to the second host (receiver).
3) *Acknowledgement:* The second host sends an acknowledgement segment to the first host to confirm the reliable connection.

After the completion of data transfer, the connection is closed unless there is a serious and irresolvable network or connection error, and the connection is *reset* [3] pp 283. The transmitter re-transmits any segments lost during transmission. The *retransmission* provides reliability to the transmission operation. For each TCP segment sent, the transmitter sets a retransmission timer within a period of time called the Retransmission Time Out (RTO). If the timer exceed RTO without having received an *Acknowledgement* for a TCP segment, then the segment is automatically retransmitted. RTO normal value is 3 seconds [3] pp 283.

2.16 MPLS

Multi-Protocol Label Switching (MPLS) is a technology used in packets' forwarding. It uses labels attached to packets to make data forwarding decisions [37] pp 5. MPLS combines the intelligence of routing with the performance of switching. During the entrance of packets to the MPLS domain, labels are attached on the packets, and the label (instead of the IP header) determines the next hop. Labels are taken off at the egress of the MPLS domain [9].

MPLS is considered a fast-forwarding technology [3] pp 305. The MPLS path, which is known as Label Switched Path (LSP), starts at head-end router and ends at tail-end router [9]. The label is a short length natural number that does not include any MAC or IP address [38]. The MPLS tunnel, also called label-switched path (LSP) tunnel, is a configured connection between two routers, in which label switching techniques are used for packet forwarding [39].

2.16.1 Benefits of MPLS

- Enabling scalable support for Virtual Private Networks (VPNs) [3] pp 306.
- Providing traffic engineering capabilities like: Controlling traffic flow in the network, reducing congestion and providing better utilization of network resources [28] pp 6.

2.16.2 MPLS and the OSI Reference Model

MPLS does not fit in the standard OSI layering. Its location is between layer 2 and layer 3 (at layer 2.5) [37] pp 28.

2.16.3 MPLS Domain Network Architecture

The MPLS domain consists of a number of routers capable of supporting MPLS services (LSRs) [3] pp 307. It is organized as follows:

- MPLS nodes are routers capable of supporting MPLS services (also called Label Switching Routers (LSRs))
- The nodes at the edge of a MPLS domain that perform the conversion of other network layer protocols into the MPLS format or provide for gateway functions between different MPLS domains are called are MPLS edge nodes.
- The nodes where MPLS traffic is originated (usually by entering from a non-MPLS routing domain) are considered MPLS ingress nodes and called head-end routers.

- The nodes where MPLS traffic leaves the MPLS domain for delivery via a non-MPLS domain are MPLS egress nodes and called tail-end routers.

Fig. 2-15 below shows the MPLS network architecture and node types.

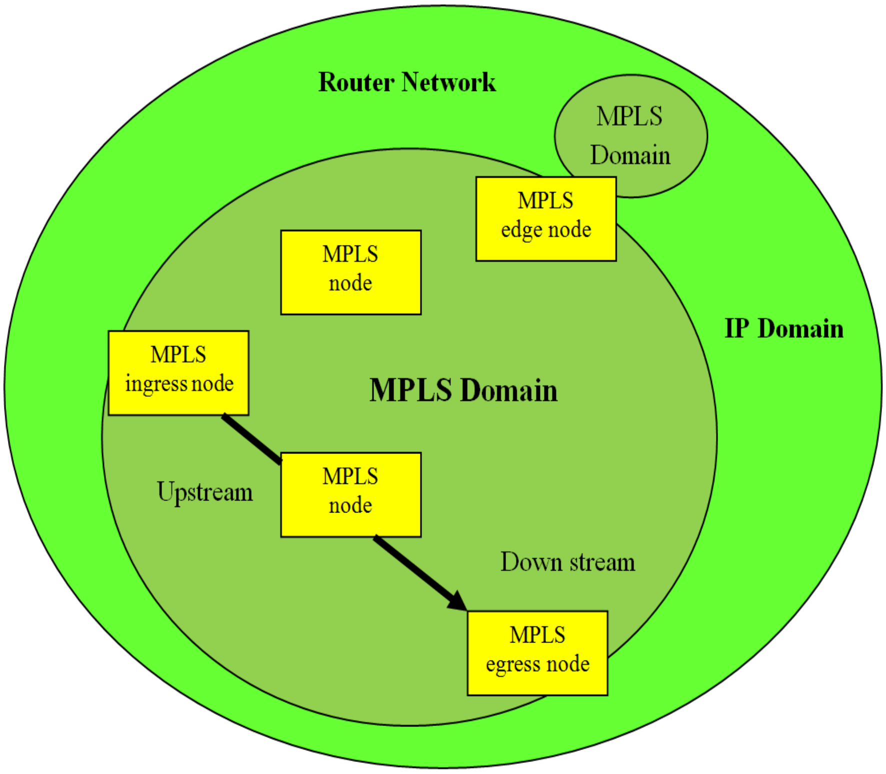


Fig. 2-15 MPLS network architecture and node types [3] pp 308

2.17 Point-to-Point Transmission Line Interfaces

The standard means of connecting routers in a Wide Area Network are by using the point-to-point lines [3] pp 323. The point-to-point line serves as a ‘reserved and private’ connection between two adjacent routers. Each interface is typically used with a layer-2 (Data link) protocol such as PPP (Point-to-Point protocol) or HDLC (High level Datalink Control) and a layer-3 (Network) protocol (IP Internet protocol).

2.18 Point-to-Point Protocol (PPP)

PPP provides a means of carrying datagrams over serial point-to-point links by providing a method for encapsulating IP datagrams [18]. The frame of the PPP is shown in Fig. 2-16.

Field length in bytes	1	1	1	1	Variable	2 or 4
	Flag	Address	Control	Protocol	Data	Frame Check Sequence

Fig. 2-16 PPP frame fields [18]

2.19 SDN

Software-Defined Networks (SDN) can be defined as a network architecture that detaches the control and data planes through changing the place of the control plane (network intelligence and policy making) to an application called a controller [40]. OpenFlow is the practical implementation of SDN. Fig. 2-17 shows the architecture of Software Defined Network.

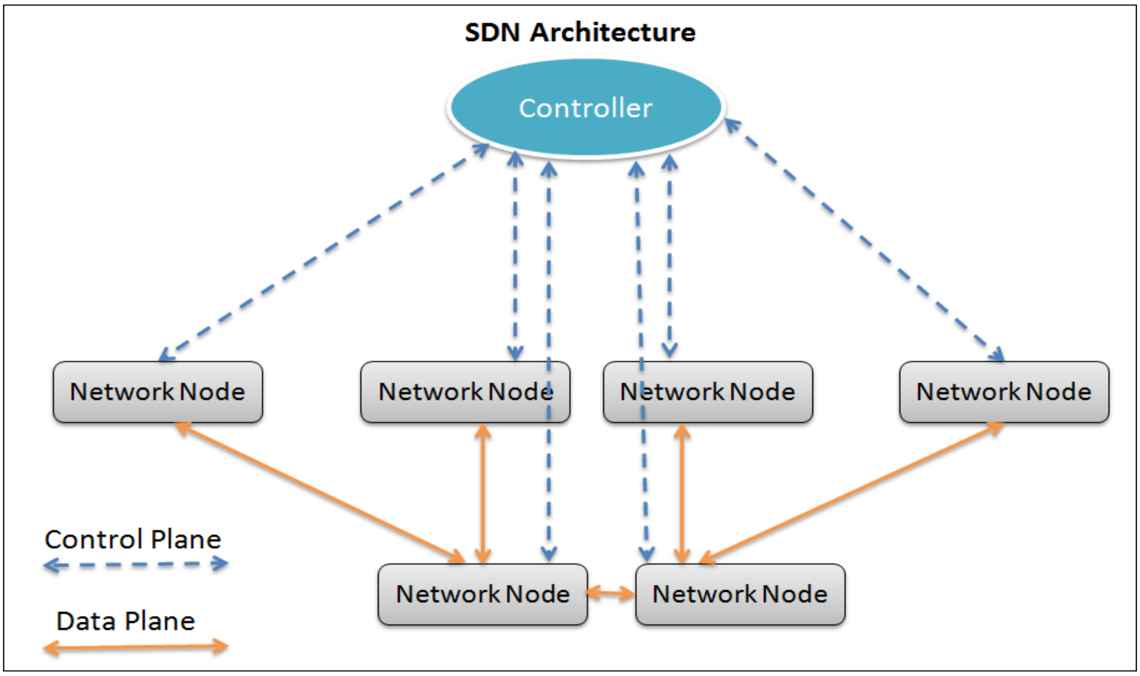


Fig. 2-17 Software Defined Network (SDN) architecture [24]

2.19.1 SDN Architecture

In SDN, the control plane is separated from the data plane. There is a central controller that is responsible of performing all complex tasks, including naming, routing, policy declaration, and security checks. The controller populates the flow table in each switch. Switches use the flow tables to manage and forward flows. SDN can be applied over Ethernet switches (layer 2) and internet routers (layer 3) [35]. The controller exemplifies the control plane while the switch exemplifies the data plane.

SDN provides a flexible architecture that permits fast and easy configuration of network devices [41]. A SDN controller uses OpenFlow protocol running over the Secure Sockets Layer (SSL) to communicate with OpenFlow-compatible switches. The OpenFlow protocol is responsible of describing message exchanges between an OpenFlow controller and an OpenFlow switch. It gives the controller the ability to manipulate the flow entries in the flow tables by adding, updating, or deleting actions [35].

2.19.2 SDN vs. Traditional Networking

SDN detaches the control plane from the data plane which forwards traffic at full speed, provides a well-defined interface between them and represents the control plane in a centralized controller which has the knowledge about application requirements and the ability to exploit the network resources [42].

2.20 Network Performance and Problems

Network performance is measuring the speed of the network. It is also represented by the ability of a network to support transactions that include the transfer of huge amounts of data, and in addition to support a large number of simultaneous transactions [43].

Quality of Service (QoS) can be defined as a set of techniques used to control bandwidth, jitter, delay and loss of packets for flows in a network [44]. QoS indicates the ability of a network to provide better service to selected network traffic over diverse LAN and WAN technologies. It points out the capability of a network to provide better service to selected network traffic over multiple technologies [18].

There are several factors that are at play in the rules for measuring network performance and affecting the QoS:

1. Bandwidth.
2. Network Congestion.
3. Latency.
4. Throughput.
5. Packets Delay Variation.
6. Route Flapping.
7. Error rate.

2.20.1 Bandwidth

The bandwidth of a network is illustrated by the number of bits that can be transmitted over the network during a certain period of time [15] pp 44.

2.20.2 Network Congestion

Congestion is an important factor behind the degradation of network performance. Congestion occurs when a node or link is carrying more data than its capability which means (in link situation) zero or negative residual capacity [45]. This is caused either by uneven distribution of traffic or lack of hardware infrastructure of the node or the link itself [4]. Congestion increases packets queueing delay, packets loss [44] and packets errors. This decreases the throughput and increases latency along with Packets Delay Variation. Network congestion adversely affects the QoS of the network.

Contemporary networks apply congestion avoidance techniques to avoid congestion and collision of packets. Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is a data link layer protocol that is used in Wireless Local Area Network (WLAN) networks to manage how hosts contact the access point [14] pp708.

2.20.3 Latency

Latency represents the total delay of time it takes for data to travel across a network from sender host to final destination host; Latency is measured in fractions of seconds [46] pp 472 - 473. There are several types of delay:

- *Propagation Delay*: The time required for a signal to travel across a transmission medium.
- *Access Delay*: This delay exists in the Wi-Fi wireless network uses a CSMA/CA approach to medium access.
- *Switching Delay*: The time consumed by a device (router or switch) to recognize the next hop and begin transmission.
- *Queuing Delay*: The time that the packet waits at the First Input First Output (FIFO) output queue until packets that arrived earlier are sent. When queuing delays become large, the network is considered congested.
- *Server Delay*: The time required for a server to examine a request, compute and send a response.

2.20.4 Throughput

Throughput is a measure of a system performance [15] pp 45. It is a measure of the rate at which data can be transmitted across the network. Throughput is specified in bits per second (bps) [46] pp 474. The network throughput is defined as the product of the probability of success of each link and the anticipated number of concurrent transmissions [47].

2.20.5 Packets Delay Variation (PDV)

PDV, sometimes referred to as jitter [48], is measured by the variance in delay of arrival packets of a flow. It is mainly applied in networks used for the transmission of real-time voice and video [46] pp 476.

2.20.6 Route Flapping

Route flapping is one of the causes of network instability [3] pp 111. It can occur when routers recalculate their routing tables and advertise the modifications. Those advertisements cause routing table recalculation in other routers and more advertisements. This may lead to a new recalculation of routing table in the first router and so on. Router forwarding decision related to specific destination oscillates between one path and another because of the repeated recalculation [3] pp 221. Congestion is one of the reasons behind route flapping [3] pp 111.

2.21 Network Performance Management

Performance management is an act of monitoring and maintaining the whole system [2].

2.21.1 Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) is an Application Layer (OSI layer 7) protocol carried by the User Datagram Protocol (UDP). It is used to monitor and manage individual items of remote network devices [3] pp 376.

SNMP comprises of three parts—SNMP manager, SNMP agent, and Management Information Base (MIBs) [49].

2.21.2 Traffic Management

Traffic demand is measured as a long-term trend. Statistical records of traffic demand can be gathered from network routers, switches or other nodes. Unpredicted traffic demand or unexpected network failures can occur at any time. This may cause degradation to network performance. Dealing with such situations involves monitoring network performance in real-time. Monitoring traffic activity and network performance is a critical operation [3] pp 595. Network traffic management is part of bandwidth management.

2.22 Load Balance

Load balancing provides a router the ability to forward packets over multiple paths to a destination [50]. In normal situation and when a router discovers multiple routes to a specific network via multiple routing processes and protocols, the router sets up the route with the lowest administrative distance in the routing table. In the situation of selecting a route from among many learned routes via the same routing process with the same administrative distance, the router chooses the path with the lowest *cost* (or *metric*) to the destination. [50, 51]. Load balance increases both the capacity and the redundancy of the network [3] pp 590.

2.22.1 Per−Destination and Per−Packet Load Balancing

In Per−Destination load balancing, the router distributes the packets based on the destination address. It sends packets to the same destination over the same path. Per−Destination maintains the sequence of the packets [50].

Per−packet load balancing means sending packets to the same destination over different paths. This ensures distribution of equal load across all links. However this does not preserve packet order with probability that some packets arrive out of order [50].

2.22.2 Equal and Unequal Path Load Balancing

During the routing table construction operation, if different paths to a destination network announce the same routing *metric* value, then it is possible to split the traffic equally among them. OSPF supports equal load balance [3] pp 590. If different paths to a destination network announce variable routing *metric* values, some routing protocols like EIGRP has the ability to unequally split the traffic to the destination among those different paths within conditions [33].

2.23 Integrated Services and Differentiated Services

The Integrated Services (IntServ), also called QoS signalling, authorizes an end station (or network node) to communicate with its neighbours to demand specific treatment for a given traffic type. This demand is propagated through every hop in the packet's path to the destination [44].

The Resource Reservation Protocol (RSVP) is a protocol for use in an IntServ environment [44].

The Differentiated Services (DiffServ) is a set of end-to-end Quality of Service (QoS) capabilities. DiffServ architecture provides different services to different types of traffic in a scalable way. The packets of different types are marked differently. This provides the possibility to treat them differently based on that marking at each hop throughout the network, without having to perform additional complex classification and marking [44].

2.24 Network Failure

Network failure refers to partial or complete failure of one component or more of a network. The reason of this phenomenon is either design malfunction, or natural or human-caused disasters. The most important types of network hardware failure are link failure and node failure.

Link Failure is defined as complete loss of communication across the link [3] pp 601. Link failure can cause packets loss which degrades the QoS. Both link failure and node failure effects can be reduced or almost eliminated by applying network restoration.

2.25 Network Restoration

Network restoration is possible to be achieved by providing more plant in the network than the normal traffic load requires. During times of failure this ‘spare’ or restoration plant is used to replace the failed equipment [3] pp 597.

Fast Reroute (FRR) is a mechanism used to protect MPLS traffic engineering (TE) label-switched path (LSPs) from link and node failures. It provides fast traffic recovery by reserving backup tunnels that bypass only a single link or next-hop nodes of the label-switched path (LSP). FRR locally repairs the LSPs at the point of failure and it belongs to MPLS Traffic Engineering (TE) [52].

2.25.1 FRR Link Protection

FRR link protection is represented by the backup tunnels that protect LSPs if a link along their path fails by bypassing the failed link and rerouting the LSP’s traffic to the next hop. Therefore they are called next-hop (NHOP) backup tunnels [52].

2.25.2 FRR Node Protection

FRR node protection is represented by the backup tunnels that bypass next-hop nodes along LSP paths and terminate at the node following the next-hop node of the LSP paths. These are referred to as next next-hop (NNHOP) backup tunnels [52].

2.26 Conclusions of Chapter 2

This chapter showed the basics of the computer networks. Those basics represented by the architecture of the networking system, the protocols used and their relationship with the architecture, the forwarding technologies and the performance of the network system along with the problems facing it. Any development that is added to the system on the later chapters will undergo those basics. The reason behind adding any development is to deal with some of the problems mentioned in this chapter.

Chapter 3

OSPF, SDN OpenFlow and Project Basic Design

Introduction to Chapter 3

This chapter discusses the basics of the OSPF routing protocol, the OSPF routing table construction, the Software Defined Networking (SDN) OpenFlow protocol features and its development efforts. Then it describes the direction of the research based on the general trajectory of the technology evolution described in **chapter 2** including the basic design of our project represented by the physical components and their structure according to the DoD model. It also explains similarities and the differences between our project and the SDN OpenFlow system. Finally, it describes the preliminary establishments of the project that enable the improvements added on the following chapters. This chapter represents the basic technical structure of the project's network design.

3.1 OSPF Routing Protocol

OSPF is one of the Interior Gateway Routing Protocols (IGP). It is used to route the Internet Protocol (IP) packets. Furthermore, OSPF is a link state routing protocol that uses the Dijkstra algorithm to build a routing table by constructing a shortest path tree and then populating the routing table with the resulting best paths [14] pp 444.

OSPF is a fast-converging protocol. It can divide routing domain into separate routing areas. OSPF also supports load sharing traffic over Equal Cost Multi-Paths (ECMP) to the same destination (if required). OSPF uses a link cost parameter as the basis to calculate the shortest path. A reference bandwidth of 100 Mbps is used for cost calculation [29]. The OSPF constantly uses the same path to forward packets between source and destination and only switches to another path during the event of link or node failure [5, 53].

3.2 OSPF Routing Table

Before delving into procedure of building the routing table, it is better to briefly mention the OSPF standard messages used by the OSPF protocol and their duties. Some expressions need to be identified as well.

3.2.1 OSPF Standard Messages and Some Basic Definitions

Below, some of the standard OSPF messages (also called OSPF protocol packets [53]):

- *Hello* messages: Discover/maintain neighbours.
- *Database description* messages: Summarize database contents.
- *Link state request* messages: Database download.
- *Link state update* messages: Database update.
- *Link state acknowledge* messages: Flooding acknowledgment.

Designated router (DR) is the router that is responsible of creating and advertising the network Link State Advertisement (network-LSA). Back-up Designated Router (BDR) is the router that ensures continued operation of OSPF in the case of DR failure [3] pp 242.

3.2.2 Brief OSPF Routing Table Construction Procedure

The router is in the non-active (*DOWN*) state when first introduced to the network. The router begins to identify its directly connected routers through a point-to-point link or single network by means of the *hello procedure.* It informs its neighbours about its details (IP address, routing area, …) through sending the first *Hello* message.

The second step is selecting of neighbouring routers and creating an *adjacency relationship,* which is followed by commence of *data synchronization* process. After completing *data synchronization* process and satisfying all Link State Requests, the databases are considered synchronized and the routers are marked fully adjacent. The link state database symbolizes the aggregation of the full set of link state advertisements (LSAs), which describe the entire topology of the network.

After the *data synchronization* process, a router has an entire copy of the link state database and it is now able to calculate its routing table to each reachable destination by using the Shortest Path First (SPF) algorithm (also known as the Dijkstra algorithm) [3] pp 243 - 252.

3.2.3 OSPF Flooding Process

In the OSPF system, each router is responsible of maintaining an up-to-date copy of the link state database. This involves knowing the full topology of the routing *domain* or routing *area* at all times. A router is required to update any changes in domain or area topology to all other routers in the domain (or *area*) by using of *Link state updates* (LSUs) containing *Link state advertisements* (*LSAs*) [3] pp 252.

3.2.4 Routing Table Lookup Operation (Forwarding of IP Packets)

When receiving an IP data packet, an OSPF router finds the routing table entry that matches the destination of the packet. The routing table entry provides the outgoing interface to use in forwarding the packet. Several routing table entries may match the destination address of the packet. In this situation, the "best match" is the routing table entry that provides the most specific match [53].

In the case of no matching routing table entry, the packet’s IP destination is considered unreachable. The router discards the IP packet and returns an Internet Control Message Protocol (ICMP) destination unreachable message to the packet’s source [53].

3.3 The SDN OpenFlow System

The basic OpenFlow system consists of:

- Controller.
- OpenFlow switch.
- Secure channel.

The OpenFlow system is represented in Fig. 3-1.

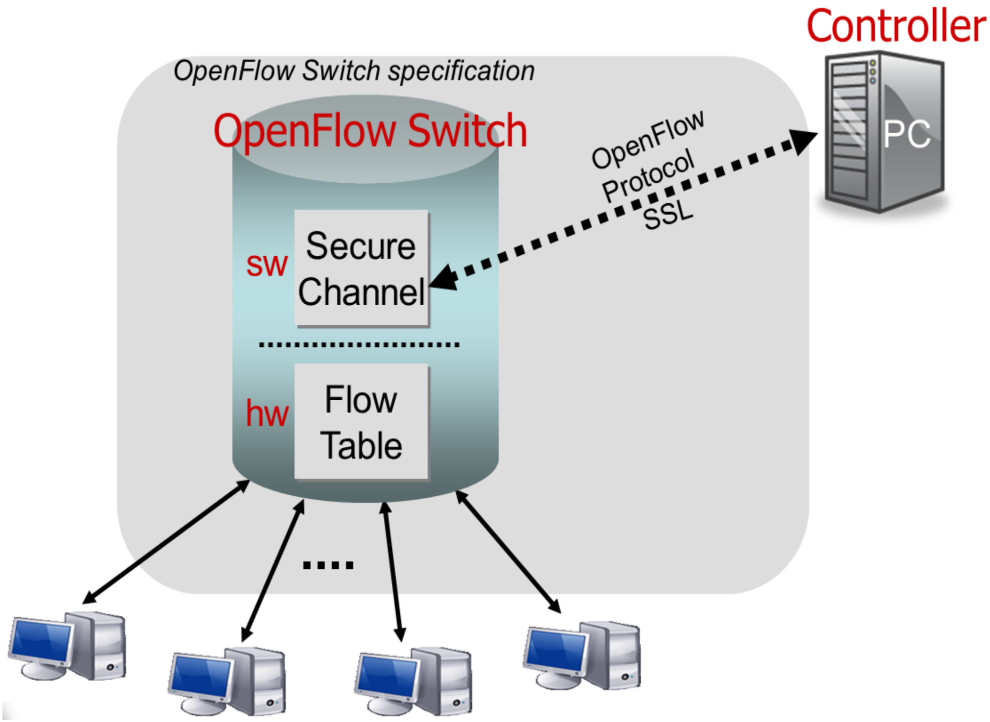


Fig. 3-1 OpenFlow system [54]

A Software Defined Networking (SDN) OpenFlow example provides a centralized decision of packets' forwarding. The controller is connected via a secure channel to all switches. The controller's responsibility is to inform the switches how to forward the traffic [41]. This central controller can see the network and all its circulating flows thoroughly, giving global and optimal management of network. The OpenFlow switches are simple and dumb since the forward decisions are defined by the controller [54, 55]. For every new flow (a flow with header that does not match any value in the existing flow table), the switch is required to ask the controller on how to distribute the flow and so the controller has to make a decision (an improved flow table) [41]. This means that the switches will fully depend on the controller, which provides central flows management and load balance where the congestion can be avoided. Therefore the OpenFlow has been proposed to be used in load balancing of data centres [56].

An important consideration when leveraging OpenFlow is the method used to program switch forwarding rules. Rules may be programmed reactively, where packets that do not match an already-installed rule are sent to the controller, which then installs an appropriate rule. They may also be programmed proactively when a new host is discovered or a virtual network is defined. This shows a trade-off between the added latency of contacting the controller regarding the first packet of each flow and necessity to maintain rules installed for all flows versus only currently active flows [42].

3.3.1 The OpenFlow Switch

The OpenFlow switch mainly [54] consists of:

- OpenFlow protocol.
- Flow table.

i. Flow table

The flow table represents the basic building block of the logical switch architecture. For any packet that arrives at a switch, it passes via one or more flow tables. The switch matches incoming packets of a particular flow with its flow tables then specifies the functions that need to be performed on the flow packets [35]. The flow table consists of one or multiple entries.

ii. Flow table entry of the OpenFlow switch

The flow table consists of entries [54] and each entry consists of three fields:

1) *Packet header*: defines the flow.
2) *The action*: defines the method the packets should be processed.
3) *Statistics*: records statistics like the number of packets and bytes for each flow, and the time since the last packet matched the flow.

iii. Switch behavior

The OpenFlow switch performs one of the flowing actions [54] in dealing with flow's packet:

1) Forward to specific port which represents the routing decision.
2) Send the packets via the secure channel to the controller.
3) Drop the packets of the flow.

iv. OpenFlow protocol

OpenFlow is a communication protocol that provides an open and standard way for a control plane that is represented by the controller to communicate with a forwarding plane which is represented by the OpenFlow switch or router [54].

3.3.2 OpenFlow Controller

The controller is responsible for all routing decisions. It is responsible of adding, updating and removing flow entries of the flow table [41, 54]. It is connected via the secure channel to the switch which is utilized to manage the flow entries in the flow table of the OpenFlow switch [41].

3.4 SDN OpenFlow Development Efforts

In SDN OpenFlow, several methods were proposed to develop performance [55] and load balance [57- 59]. In [55], Fernandez et al. propose the use of a proactive controller that pre-populates flow tables to the forwarding devices, which solves the processing time problem.

In the case of the LOad-Balancing over UnStructured networks (LOBUS) algorithm [57], Handigo et al. considers a load-balancer, called Plug-n-Serve. Plug-n-Serve has the ability to balance the load over arbitrary unstructured networks, and tries to reduce the average response time. In [58], Long et al. suggest the application of the LoAd-BalancEd Routing with Openflow (LABERIO) algorithm, which deals with congestion and selects a path according to the available bandwidth; a *NOX* controller is used in LABERIO. In [59], Koerner et al. slice the network resources and uses a different controller for every slice by creating a Flow Visor application to deal with different services. In [60], Yu et al. make use of the controller to deal with the link failure incidences in OSPF networks.

The SDN-Based Equal Cost Multi Path (ECMP) distribution algorithm in data centers is proposed by Zang et al [61]. The OpenFlow protocol is used to optimize the ECMP algorithm to provide dynamic adjustment to the flow forwarding operation depending on the bandwidth utilization of core links. If the bandwidth utilization of a core switch exceeds a specific threshold, the controller will forward some flows to other links which have lower bandwidth utilization [61]. The SDN-based ECMP distribution algorithm is restricted to deal with network of FatTree topology where the costs of the paths are equal and the change in flow path is applied on the core switches only.

The latency in communication between the routers and the controller has been discussed in [62] where Phemius et al experience its effect on TCP and UDP packets by using a network *floodlight* controller. Two solutions are suggested and to be applied together: double the contact with the controller and increase the buffer size of the switch where packets are held. Even with those solutions, the latency in taking decisions still exists.

3.5 Research Objectives

This chapter shows the basic structure of our project. It illustrates the used devices and their genuine, developed and added models. It shows the locations of the models in comparison with the standard DoD model. It also describes preliminary preparation of the network.

This chapter represents an introduction to the later chapters.

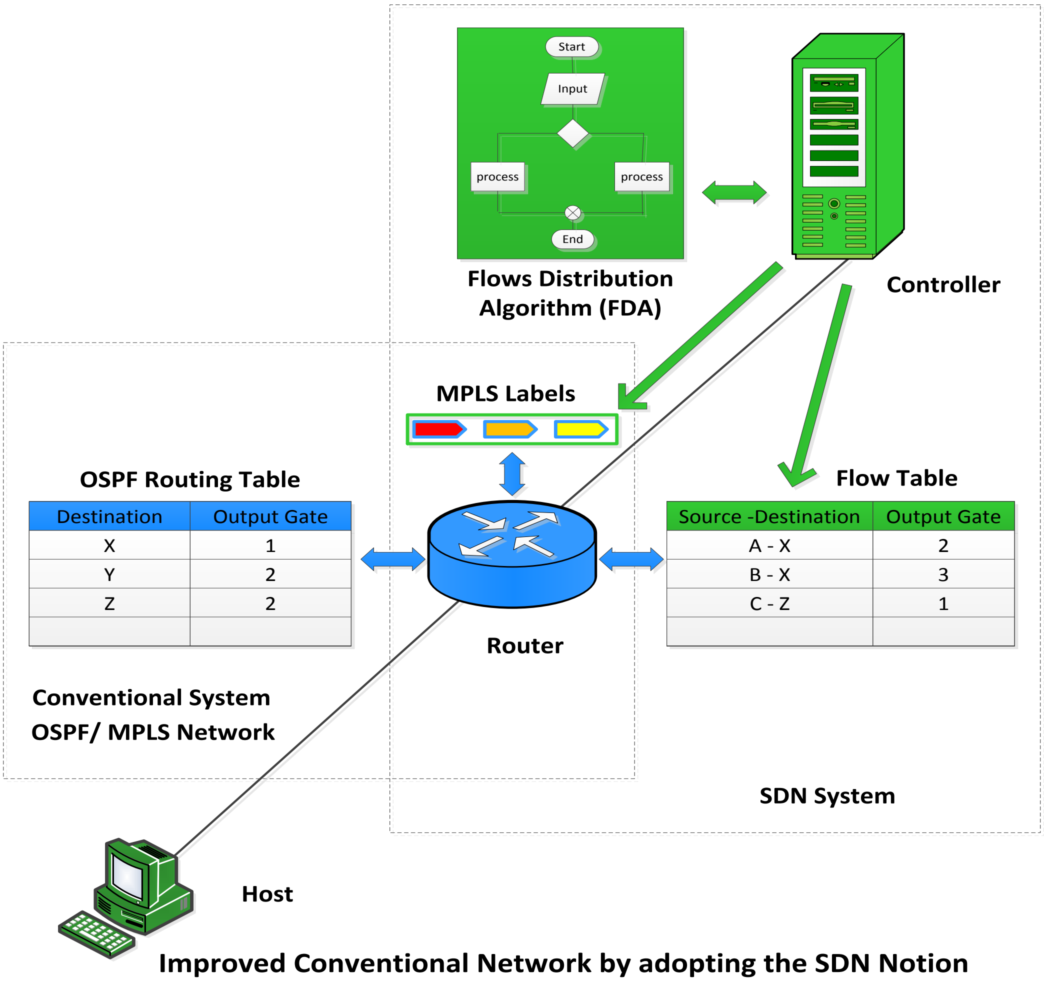
3.6 Software and Models

All the designs, improvements and tests are implemented by using OMNeT++ simulator (version 4.3) through its INET Framework project. OMNeT++ is an object-oriented modular discrete event network simulation framework. It uses C++ and Network Description Language (NED) to instantiate and to operate computer networks. The OMNeT++ simulator provides infrastructure and tools for writing simulations. It also enables reusing the code of simple modules that compose the compound model via sub-classing and redefining virtual member functions [63]. This feature enables the developers to develop models by adding extra code to perform specific duties without affecting the original standard code.

Packages in INET Framework project are designed and organized strictly to instantiate the OSI layers. INET Framework contains IPv4, IPv6, TCP, SCTP, UDP protocol implementations in addition to several application models and technologies like OSPF and MPLS [64].

3.7 The Basic Structure of the Network

Our network is a conventional OSPF/ IP-MPLS routed flows computer network that extends to utilize a methodology similar in many aspects to that used in SDN networks. It combines the practical side of the conventional networks with the intelligence of the SDN. The basic design is as shown in Fig. 3-2.


Start
Input
process
process
End
Flows Distribution Algorithm (FDA)
Controller
MPLS Labels
OSPF Routing Table
Flow Table
Router
Conventional System OSPF/ MPLS Network
SDN System
Host
Improved Conventional Network by adopting the SDN Notion

OSPF Routing Table

Destination	Output Gate
X	1
Y	2
Z	2

Flow Table

Source -Destination	Output Gate
A - X	2
B - X	3
C - Z	1

Fig. 3-2 Network basic design

The network hardware components are:

- Controller.
- Routers.
- Hosts.
- Channels (links).

Our network represents a *flat routing system*; the routers are peers of all others [18] as well as the hosts. The used controller has different capabilities. The routers are extended with extra models in addition to their genuine models, which are also developed to be computable with the new requirements.

3.7.1 Network Experimental Topology and Basic Features

The basic topology of our project is shown in Fig. 3-3. It is a general and simple topology and does not represent any specific existing network. The plain topology makes the explanation of operations easier to be followed. However, our development extends to more complicated topologies, which show fine results as well.

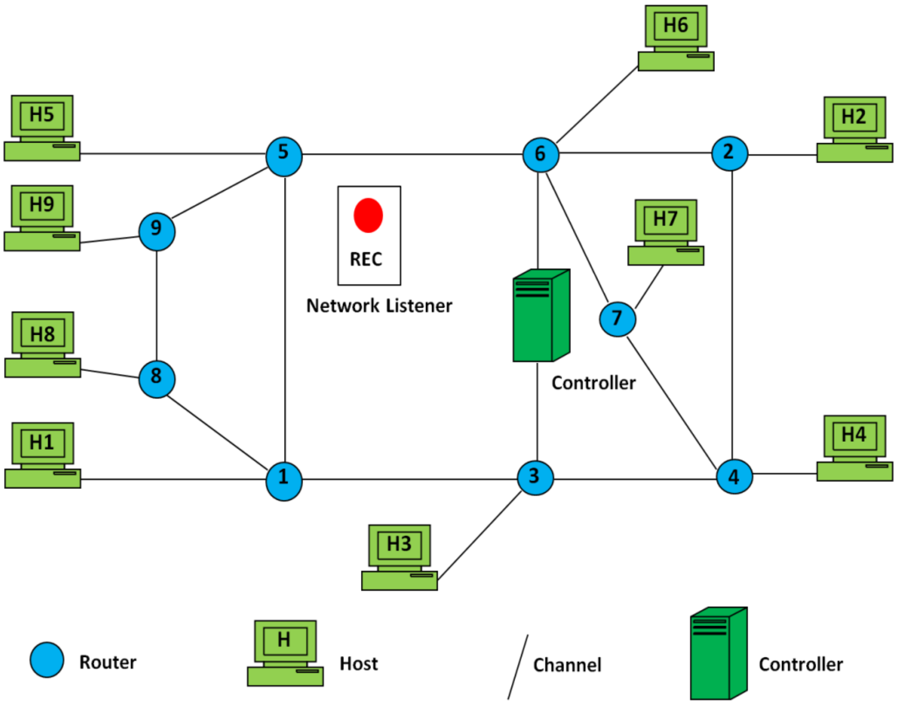


Fig. 3-3 Project network experimental topology (9 routers network)

- Wide Area Network (WAN).
- The hierarchical design of all devices implements the DoD model structure.
- Per−Destination load balancing concept.
- One autonomous system.
- Wired system (routers are connected with each other by serial cables).
- Uses OSPF routing protocol to forward IP messages and packets (dynamic routing).
- Implemented under single OSPF area.
- Provided with MPLS forwarding mechanism.
- Uses UDP in control messages because it is faster.
- Consists of single controller, multiple routers and hosts. Each router is connected to one host.

3.7.2 Connecting the Controller to the Network

The controller is a separate device that can be placed anywhere inside the network. It cannot be directly connected to all routers of the network. It is directly connected to some of the network's routers. The channel that connects the controller with its directly connected routers has the features of the Ethernet cable.

For the routers that are not directly connected to the controller, they contact the controller through the paths identified according the OSPF routing tables across the other routers. The controller contacts the routers that are not directly connected to it according to the same method but via the opposite direction. The information, updating and control messages that exchanged between the controller and the routers use the OSPF routing tables to find their way from controller to the routers and vice versa.

In the basic network topology of our project, Fig. 3-3, the controller is directly connected to two routers (router 3 and router 6) among nine total routers that exist on the network. Router 8 (for example) contacts the controller by sending messages that passes through routers 1 and 3 subsequently until reaching the controller. The controller contacts the router 8 by sending messages that passes through routers 3 and 1 subsequently until reaching router 8.

3.7.3 Network Devices Connections

The routers contact the controller as they contact any host. The controller has a simple routing table. It contacts routers according to its routing table. There is no direct connection between the controller and any of the network hosts.

3.7.4 Similarity and Difference between our Project and SDN OpenFlow

Regardless the perfectness of the network design, unpredicted traffic demand or unexpected network failures may affect the performance of the network negatively. Our project represents a conventional OSPF network that utilizes a methodology similar to that used in the SDN (existence of a controller and the messaging system between it and the routers). The reason behind using this methodology is to improve the Quality of Service (QoS) by enhancing the network's performance. Our network uses neither SDN devices nor SDN protocols as the aim of this project is to develop the conventional OSPF/IP -MPLS routed networks rather than the SDN networks.

Implementing the SDN methodology in a conventional network involves upgrading the hardware of the existing devices (the routers) to be compatible with the new design as well as adding a controller to the network. It also involves upgrading a messaging system as a dialog between the controller and the routers.

Unlike the traditional conventional network which combines both the control plane and the data plane at the forwarding devices and unlike the SDN which completely separates the control plane from the data plane, our project splits the control plane into two parts. The first part lies inside the controller while the other part lies inside the routers. The better the cooperation between those parts results in higher level of performance. Our improved network keeps the main features of the conventional network as well as imports some features from the SDN. Therefore, it can be considered as a hybrid system that lies between the conventional networking system and SDN as it utilizes the best features of both of them.

Unlike the forwarding dummy devices of the SDN OpenFlow system, the routers of our project are more intelligent and have more capabilities. They have the ability to build their routing tables without any help from the controller. During topology changes event, our improved routers can update the routing tables locally without any help from the controller as well. The controller intrudes to improve the performance of the conventional network when dealing with four main subjects:

- Congestion resulting from un-even flows distribution in an OSPF network.
- Time consumed to establish primary MPLS tunnels in an OSPF/MPLS routed network.
- Link Failure in an OSPF network.
- Bandwidth reservation of backup MPLS tunnels.

Our improved conventional network adopts some of the ideas mentioned at **section 3.4** which related to OpenFlow developments efforts. Similar to [55], the controller of our project is a pro-active device but when dealing with congestion problem. Similar to the approaches proposed in [57-58], the controller of our project has its algorithm and it uses this algorithm when dealing with flows distribution. The controller of our project also deals with link failure which makes it similar to the approach proposed in [60].

3.8 Network Architecture

The architecture of all network's devices matches the architecture of the DoD model. This section gives brief description on some of the most important models. However, more description on them and the other models will be given when necessary on the later chapters.

3.8.1 Controller Hardware Design

The controller combines both router and host features beside its unique features. Similar to routers, the controller has several interfaces that connect it to several routers. Therefore, it has different addresses. Routers can contact the controller through contacting any of its addresses. The controller has also a routing table, which it uses to contact routers. Similar to hosts, the controller neither sends any OSPF hello messages nor plays any role in *Data* packet forwarding operation. Thus, it can be said that the controller represents a group of hosts provided with routing table and an algorithm responsible of flows distribution called Flows Distribution Algorithm (FDA). The controller's architecture (hardware structure) is illustrated in Fig. 3-4.

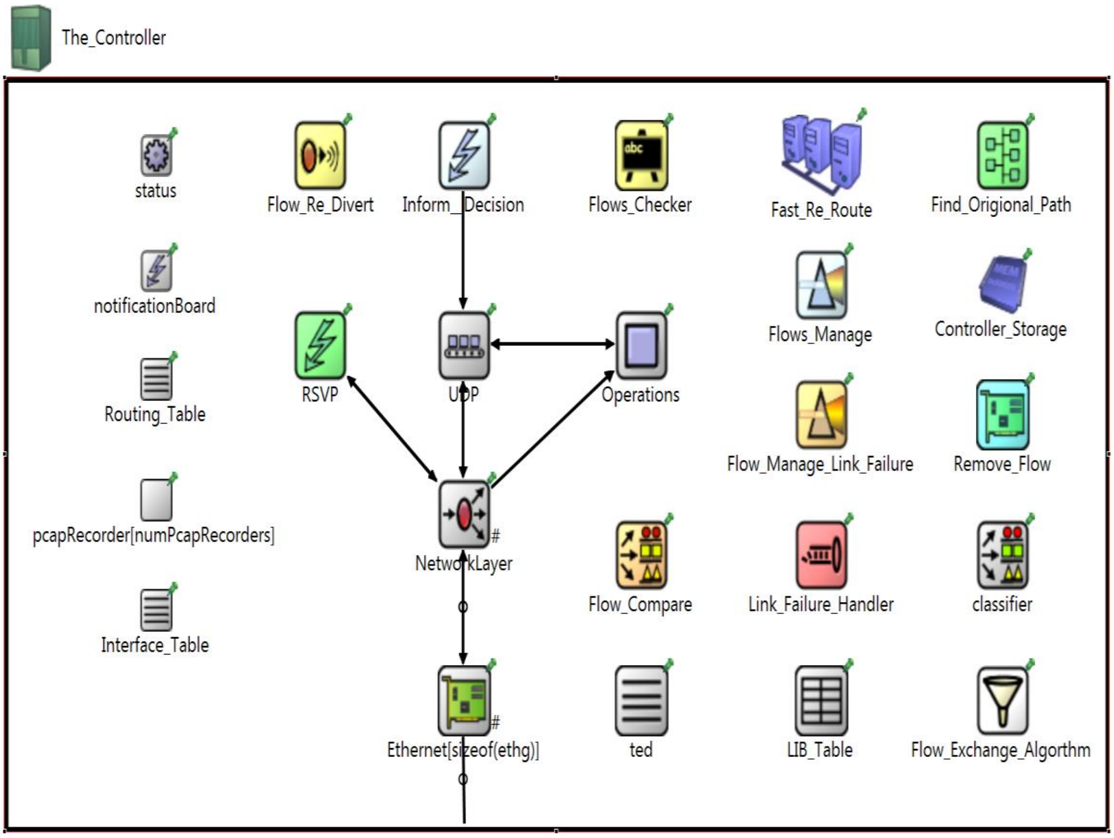


Fig. 3-4 Controller's architecture (hardware structure)

Similar to any network device, the controller construction represents the 4 layers DoD model. The basic models of the controller and their functions are:

i. Process/ Application Layer Models

- ***Operations* model:** Manages all the activities of the controller and controls all other models. It is also responsible for data transfer among the other models.
- ***Controller Flows Manage* model:** Operates to deal with congestion problem and flows distribution. It controls: *Flow Compare* model, *Flows Checker* model, *Flow Exchange Algorithm* model, *Remove Flow* model, *Find Original Path* model and *Flow Re-Divert* model.
- ***Inform Decision* model**: The messages that are responsible for the establishment or removal of the flow table are created in this model.
- ***Controller Storage* model**: This container represents the controller's database. It contains the network information like topology, devices addresses, links bandwidth, routing tables of the routers.
- ***Link Failure Handler* model**: This model is responsible for updating the database information during the event of link failure.
- ***Fast Re-Route* model**: This model is responsible for calculating the backup MPLS tunnels.

ii. Host to Host Layer (Transport Layer) Models

- ***RSVP* model**: The messages that responsible for the establishment and removal of the MPLS tunnels are created in this model. This is transport layer model.
- ***UDP* model**: Any packet that has been established at the upper layer is identified as a UPD frame at this model.

iii. Internet Layer

- ***Network Layer* model**: The normal network layer in the OSI model which consists of several sub-models like *IP* model, *ICMP* model *ARP* model and *Error Handling* model.
- ***Routing Table* container model**: In this container lies the routing table of the controller.

iv. Network Access Layer

- ***Ethernet* model**: The normal data link layer.
- ***Interface Table* container model**: In this container lies the interface table of the controller.

3.8.2 Router Hardware Design

The design of the routers in our network represents the basic Inet-OMNeT++ design which instantiates the real routers. It is developed with modifications in code and several added models to be compatible with the new design requirements. The developed routers change their behaviour according to the *Control* messages received from the controller as they can contact the controller. The extra code modifies the performance of the routers without changing any of the standard written code. Fig. 3-5 shows the architecture design of the improved router represented by its hardware structure.

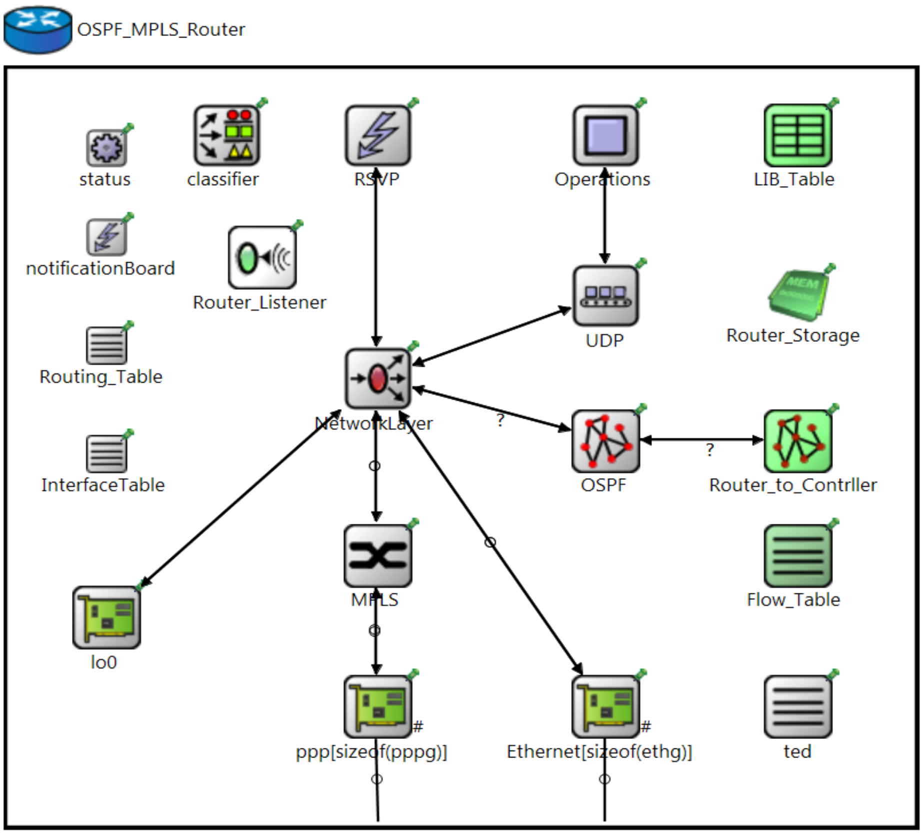


Fig. 3-5 Router's architecture (hardware structure)

To enable the controller from controlling the router, some kind of application layer has been added to it and it is represented by the *Operations* model. The most important models of the router are as illustrated below while the others will be mentioned later when used.

- ***Operations*** **model:** Manages all the activities of the router regarding the relationship with the controller. It represents some kind of process/application layer (of the DoD model) that is added to the router. It has control over some other models as well.

- ***Router Storage* model**: This container contains part of database of the router like the controller addresses along with their metrics and gateways. It is is an additional model added to the standard router.
- ***Router Listener* model**: The activities of the router are recorded at this model like the number of the forwarded and the dropped *Data* packets.
- ***Flow Table* container model**: Flow table model is an additional model added to the router. The flow table is temporary and has higher priority than the OSPF routing table as it substitutes the routing table in forwarding specific flows.
- ***Developed OSPF Routing* model**: It is a development to the standard *OSPF Routing* model which is the model that is responsible of constructing the OSPF routing table. It orders and receives the *OSPF standard* messages.
- ***Developed IP* model:** The *IP* model represents genuine part of the *Network layer* model. It has been modified to serve the added requirements of the network and its name is changed to *Developed IP* model.
- ***Developed RSVP* model:** This model is responsible of creating the MPLS tunnel. It represents an improvement to the standard *RSVP* model.
- ***Developed MPLS* model:** This model is responsible of forwarding the *MPLS Data* packets.
- ***Ethernet model and PPP model:*** Those models represent the normal data link layer.

3.8.3 Host Hardware Design

The host used represents the standard host of the Inet-OMNeT project. The models of the host is not subject to any code modifications except adding some extra features used to record specific data. Each host sends UDP *Data* packets to some other hosts. Fig. 3-6 shows the architecture design of the host represented by its hardware structure. The private models of the host are the *Host Listener* model and the *UDP App* model.

- ***Host Listener* model**: This model is the only added model to the Inet-OMNeT standard designed host. It records Quality of Service (QoS) outcomes of the flows like flow latency and packets delay variation (PDV).
- ***UDP App* model**: This model is responsible of generating the *Data* packets that represent the flows. It is genuine model within the standard host of the Inet-OMNeT project.

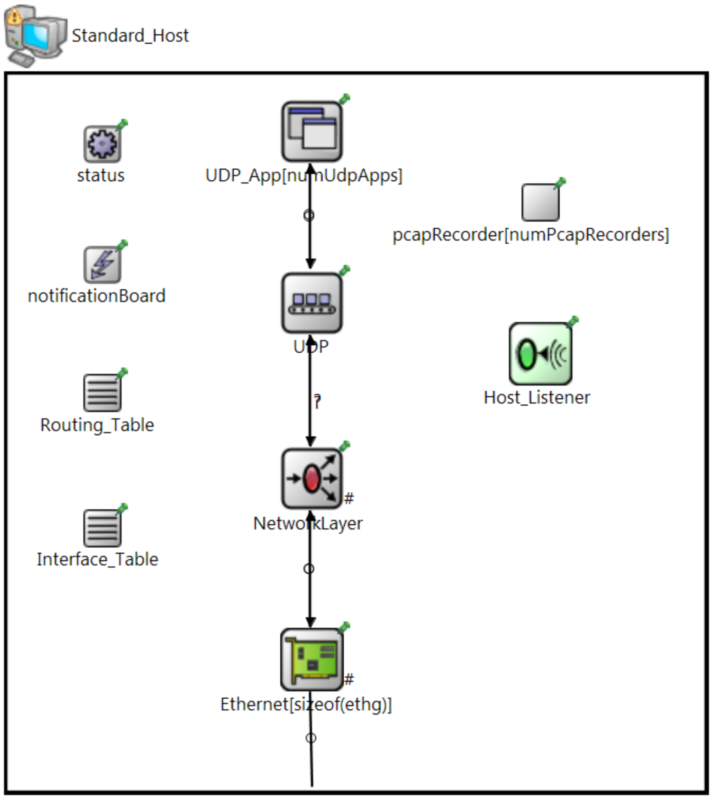


Fig. 3-6 Host's architecture (hardware structure)

3.8.4 Network Listener Model

This model only records the network performance statistics like throughput. It collects some of its data like the total number of packets obtained from the *Router listener* models and *Host Listener* models. The *Network listener* model has no effect on the network operation. It was shown in Fig. 3-3 within the network.

3.8.5 Network Channels (Links)

The data rate channel is the standard channel that is used, which takes into consideration both bandwidth and delay. The type of channels that connect the routers with each other adopts the features of general data rate channels, while the channels that connect routers with hosts or with the controller, adopt the features of the Ethernet cable.

3.9 Messages and Packets of the Network

Before delving into the network operation, it is better to mention the most popular messages and packets that travel across the network.

3.9.1 Popular Messages

- *Network data* message: issued by each router and directed to the controller. The *Network data* message carries the network static information represented by the topology table of the router.
- *Routing table* message: issued by each router and contains the routing table of the router issuing it. It is directed to the controller.
- *Real time flow status updating* message: issued by the head-end router and directed to the controller. The *Flow status updating* messages carry the flow current information like flow source, flow destination and flow data rate.
- *Real time link failure* messages: the routers directly connected to the failed link issue this message and send it to the controller to inform the link failure event. It contains the address of the failed link.
- *Control* messages: issued by the controller and directed to specific routers to order those routers to perform specific jobs. There are several kinds of *Control* messages which to be explained when due.
- *OSPF standard* messages.
- *RSVP standard and developed* messages: used to establish the primary MPLS tunnels. Some of the *RSVP developed* messages are kinds of *Control* messages.
- *Backup tunnels* messages: used to establish the backup MPLS tunnels. They are also part of the *Control* messages.
- *Data* packets: issued only by hosts and directed to other hosts. The *Data* packets represent the flows. They are defined as User Datagram Protocol (UDP) packets.

All the above messages and packets traverse the routers of the network to reach to their destinations. Beside those messages, there are some other messages used for specific reasons. The next sections and chapters explain all messages and their contents in details. Messages and packets use UDP transport protocol as it is faster than TCP.

3.9.2 Messages Generation over the DoD Model

Messages that exchanged between routers and controller are issued by orders of the *Operation* model of the sender. They may be issued at the application/process layer or any other layer depending on the receiving layer at the receiver as every layer at the sender contacts its counterpart at the receiver.

3.10 Simulation Commences and Preliminary Setup Operation

Development of network performance involves a preliminary setup operation. Each device has a specific duty. Both controller and routers integrate with each other. The OMNeT++ simulator of version 4.3 supports the OSPF version 2 in its Inet project. Similar to any other OSPF network, the routers at the beginning of the simulation generate *OSPF hello* messages and send them to their neighbours.

3.10.1 Building the Topology Table of the Router

From the first *Hello* message received at the *Developed OSPF Routing* model, the router extracts the address of the sender (neighbour) router and the gateway to this neighbour. It also gets the total bandwidth of the link from the interface table model. The router uses this information to build its neighbours' table.

3.10.2 Building the OSPF Routing Table of the Router

Similar to any other OSPF network, routers of our network commence and continue exchanging *OSPF Link state update* messages and updating their routing tables until reaching the state that all existing networks are identified in the routing tables via their best paths. When reaching to this level, there are no more updates added to the constructed routing tables. The time consumed until reaching to this state depends on the degree of complexity of the network's topology as well as the number of the existing routers.

3.10.3 Broadcasting the Controller's Addresses

Upon receiving the first *Hello* message, the controller dismantles this message in its *Operation* model in a similar way the *OSPF Routing* model of a router dismantles the received *Hello* message. It extracts the address of the directly connected router, saves the address in its database (*Storage* model) and deletes the *Hello* message. The controller then issues a *Controller addresses* message which contains all its addresses and sends it to the router that is directly connected to it. The directly connected router extracts the controller addresses, saves them into its storage database and keeps them on hold until finishing the construction of its routing table. The structure of the *Controller addresses* message is as shown in Fig. 3-7.

Controller Interfaces Number (N)	**Controller Address 1**	**Controller Address 2**	**……**	**Controller Address N**

Fig. 3-7 Controller addresses message structure

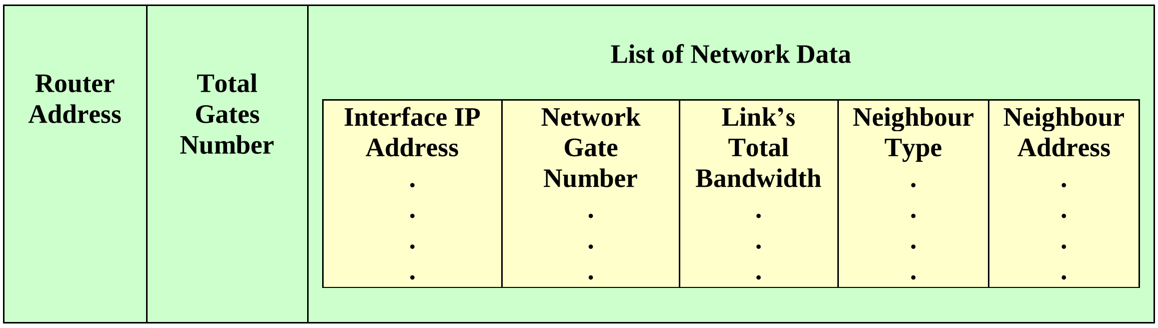
After finishing the construction of their routing tables, the directly connected routers to the controller announce the addresses of the controller by broadcasting *Controller addresses* message to other routers which repeat the broadcasting and saving operation until all the routers of the network become familiar with the addresses of the controller. The network performs this operation only once.

3.11 The Router Preliminary Steps

A router is the device that forwards the *Data* packets to their destinations. Normally, it forwards packets according to its routing table. Every router is connected to one host and some other routers. Some routers are directly connected to the controller. The routers that are not directly connected to the controller use other routers' forwarding capabilities depending on their OSPF routing table to contact the controller.

After building its routing table and receiving the *Controller addresses* message, the router adopts the controller address that matches the existing address of least *metric* value on its routing table as to be explained on **section 3.14** later. It sends its static data (neighbours' table) and dynamic data (routing table) to the controller.

Each router dynamically collects the information regarding its premises which dealt with as static data. It sends this static data that is represented by neighbours' table to the controller via the *Network data* message. The structure of the *Network data* message is as shown in Fig. 3-8.



Router Address	**Total Gates Number**	**List of Network Data**				
		Interface IP Address	**Network Gate Number**	**Link's Total Bandwidth**	**Neighbour Type**	**Neighbour Address**
		.	.	.	.	.
		.	.	.	.	.
		.	.	.	.	.
		.				.

Fig. 3-8 Network data message structure

Router Address**:** Represents the main address of the router.

Total Gates Number**:** Represents the total number of interfaces of the router.

List of Network Data**:** *Interface IP Address*, *Network Gate Number*, *Link Bandwidth*, *Neighbour type* and *Neighbour Address*.

The *List of Network Data* fields represent the information about all the directly connected devices. *Neighbour type* field is either a router or a host. (The router considers the controller as a host).

In addition to the topology table that sent to the controller, each router sends its dynamic data represented by its routing table to the controller via the *Routing table* message. The structure of the *Routing table* message is as shown in Fig. 3-9.

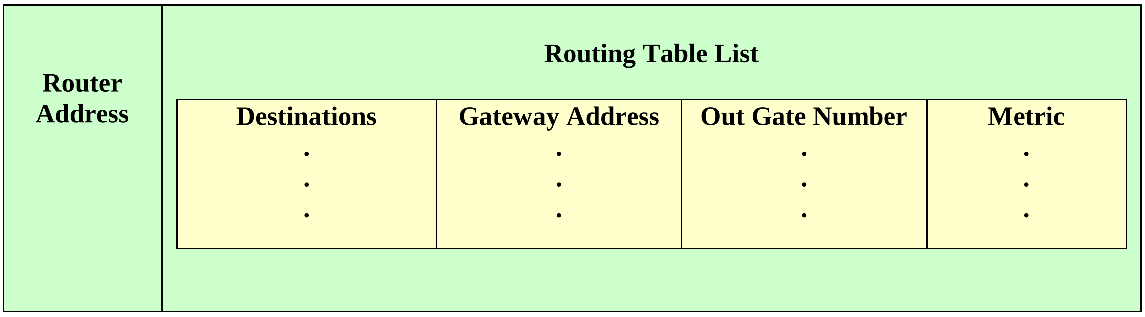


Fig. 3-9 Routing table message structure

***Router Address*:** Represents the main address of the router.

***Routing Table List*:** This list represents the routing table inside the router. It contains: *Destination, Gateway Address, Out Gate Number* and *metric* to that destination.

The controller receives both the *Network data* messages and the *Routing table* messages from all routers. It extracts all the information it needs from them and store it at its database.

From the received *Network data* messages, the controller constructs the topology of the network and from the received *Routing table* messages, the controller knows the behavior of the each router in terms of flows forwarding.

All routers send their static and dynamic data to the controller before hosts commence exchanging the *Data* packets which represent the flows.

Informing the controller both the network topology and the routing tables of all routers makes it familiar of network behavior when dealing with any possible flow. Being familiar with network resources and behavior enables the controller from developing the network performance under the available scope.

3.12 The Controller Preliminary Operation and Features

The controller is the device that has the responsibility of monitoring the traffic distribution across the network. It only intrudes to develop the network's performance when necessary.

The controller is a real time manager device that manages the network successfully, which should have the following features:

- Knowledge.
- Real time monitoring.
- Analysis tools and intelligence.
- Ability to affect the network devices.

The controller can be connected only to some routers. For the routers that are not directly connected to the controller, the information, updating and control messages use the OSPF routing table to find their way from routers to the controller and vice versa. In the basic network topology of our project, Fig. 3-3, the controller is directly connected to two routers among nine total routers that exist on the network.

3.12.1 Knowledge

The more the knowledge the controller has the better it can perform. The controller collects all the information related to the network, such as topology, channels bandwidth and OSPF routing tables of all routers from the *Network data* messages and the *Routing table* messages sent by the routers. According to this information, the controller re-constructs the network topology inside its data base. Thus, it knows exactly the behavior of the routers (the forwarding decision related to *Data* packets) when they deal with the applied flows.

It can be said that the mathematical and comparison operations inside the controller represent a replica of the network and the decision is the most complicated operation of the controller. The knowledge of the controller is represented by:

- Topology construction.
- Constructing the routing table of the controller.
- Calculating all possible paths between all hosts.

i. Topology construction

According the information obtained from of the received *Network data* messages, the controller reads out the values of the bandwidth of links (channels) and their locations inside the network (their directly connected nodes). The controller inserts the links location and values into a map container to use them later. Every link is added only once. Fig. 3-10 represents the structure of network topology container inside the controller's database representing a fictional network.

First Router	**Interface Address (IPv4)**	**Link Bandwidth**	**Interface Address (IPv4)**	**Second Router**
Router A	*X1*	*U Mbps*	*X2*	*Router B*
Router A	*Y1*	*V Mbps*	*Y2*	*Router C*
Router G	*Z1*	*W Mbps*	*Z2*	*Router F*
. .	. .	. .	. .	. .

Fig. 3-10 Network topology container

The network topology container table provides the controller the knowledge that it needs when dealing with flows distribution.

For example, part of the topology table that represents our network project of Fig. 3-3 is shown in Table III-I.

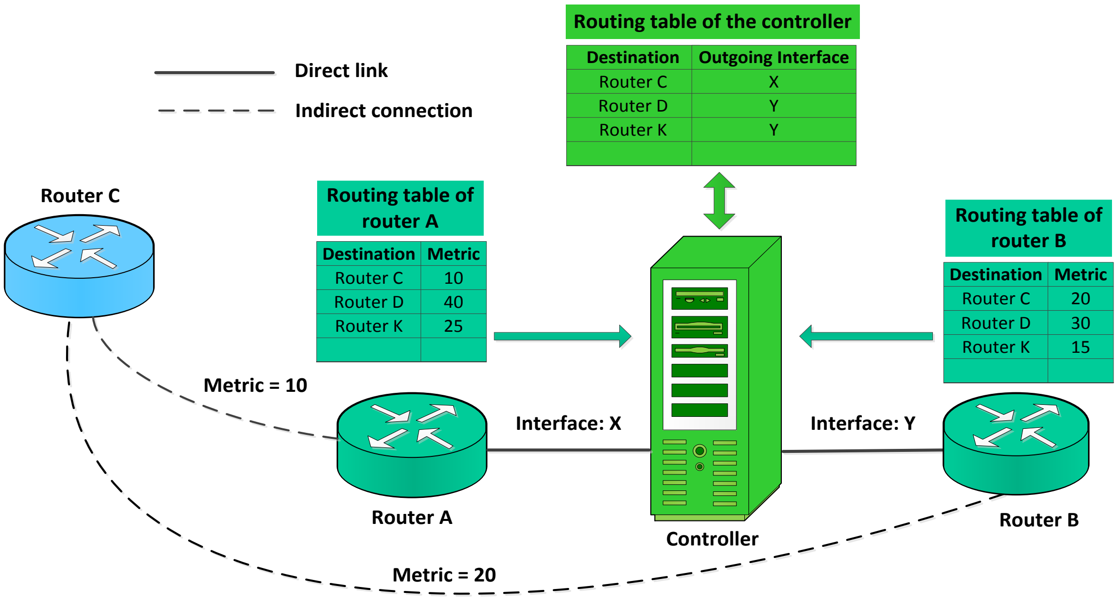
TABLE III-I
TOPOLOGY TABLE OF NETWORK OF FIG. 3-3

First Router	**Link Bandwidth**	**Second Router**
Router 1	*30 Mbps*	*Router 3*
Router1	*30 Mbps*	*Router 5*
Router 1	*20 Mbps*	*Router 8*
Router 2	*40 Mbps*	*Router 4*
Router 2	*40 Mbps*	*Router 6*
. . .	. . .	. . .

ii. Constructing the routing table of the controller

The controller imports its routing table from its directly connected neighbour routers. After finishing the construction operation of their routing tables, the routers that are directly connected to the controller send the constructed OSPF routing tables to the controller. The controller extracts the addresses of the destinations and their corresponding *metric* values from the received routing tables. It compares the *metric* values of each address obtained from the different received routing tables and finally it adopts the address of the lowest *metric* value and its corresponding interface by adding them to its routing table. It also stores the other higher *metric* values of destinations and their interfaces in a subsidiary container to be used in case of link failure.

It can be said that the controller behaves as a router when contacting other routers. It sends its *Control* message through the nearest interface to the router that it wants to contact. Routers deal with the controller as if it is a group of hosts and send their messages to the nearest host. Fig. 3-11 shows how the controller chooses the outgoing interfaces to contact the destinations depending on the lowest *metric* values in a fictional network where Router A and router B are directly connected to the controller while, router C lies far away from the controller. The controller adopts the path via router A to contact router C as it has the less *metric* value.



Routing table of the controller

Destination	Outgoing Interface
Router C	X
Router D	Y
Router K	Y

Routing table of router A

Destination	Metric
Router C	10
Router D	40
Router K	25

Routing table of router B

Destination	Metric
Router C	20
Router D	30
Router K	15

Fig. 3-11 Controller’s routing table construction

For example, in our basic project network of Fig. 3-3, the controller uses the outgoing interface to router 3 in contacting routers 1, 4 and 8 while it uses the outgoing interface to router 6 in contacting routers 2, 5, 7 and 9.

iii. Calculating all possible paths connecting all existing hosts

In the fictional network of Fig. 3-12, there is ***P*** number of paths between *source-destination* pairs of hosts including one OSPF path and ***P-1*** non OSPF paths. The non OSPF paths may share one or several links with the OSPF path as well as among each other. All paths begin at the head-end router node and terminate at the tail-end router node. The head-end router represents the router that is directly connected to the *Source host* while the tail-end router represents the router that is directly connected to the *Destination host.* At least, any path consists of two routers (nodes) and one link. It can consist of several routers and links. The number of the routers of the path is higher than the number of its links by one. Fig. 3-12 shows general depiction of the proposed paths between *source-destination* pairs including the OSPF path.

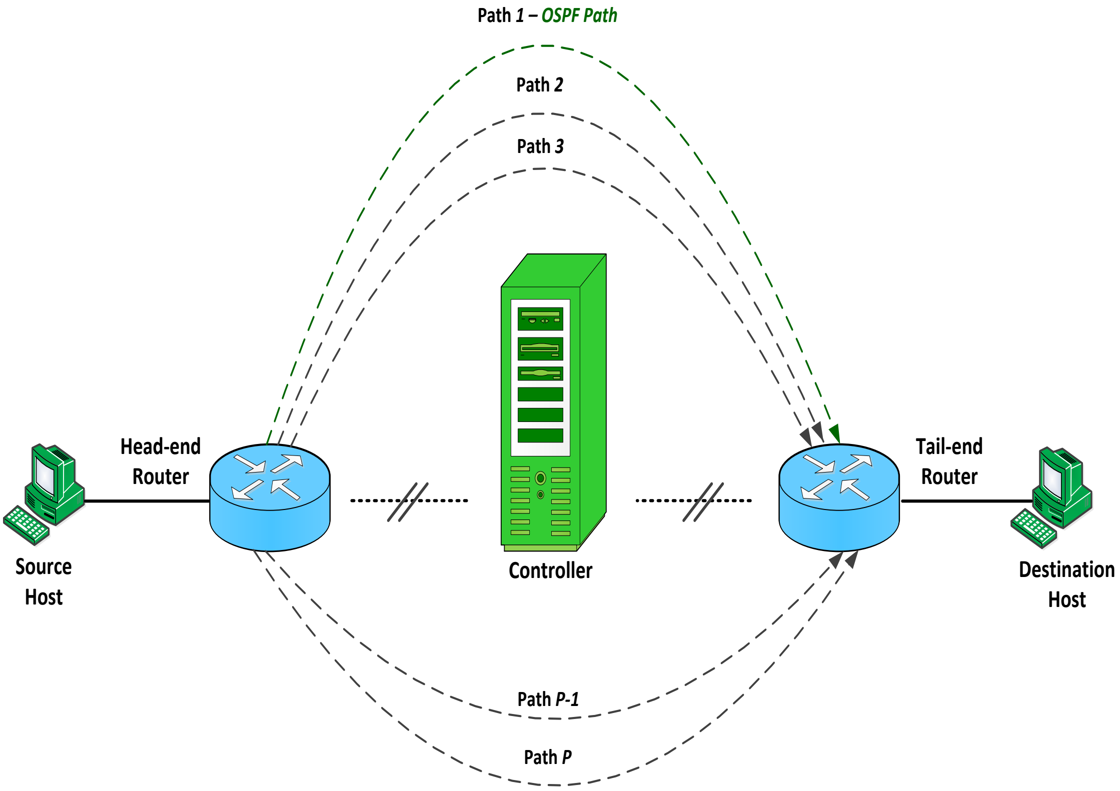


Fig. 3-12 Paths between source-destination pair

The controller integrates its knowledge by calculating all the available paths from all sources to all destinations. It first calculates the OSPF paths then it calculates all the other paths.

- **Calculating the OSPF paths**

From the received *Network data* messages, the controller extracts the addresses of the routers and the addresses of the hosts. It puts those hosts in a list named the *list of hosts*. It puts each host with its directly connected router in a specific map container named *router-host* map. From the received *Routing table* messages, the controller extracts the routing table of each router and stores it into a container. The routing table of each router should contain the addresses of all hosts that exist in the network and gateway to each host. The controller applies the procedure of extracting the OSPF paths at its *Find original path* model. The procedure is illustrated in Fig. 3-13.

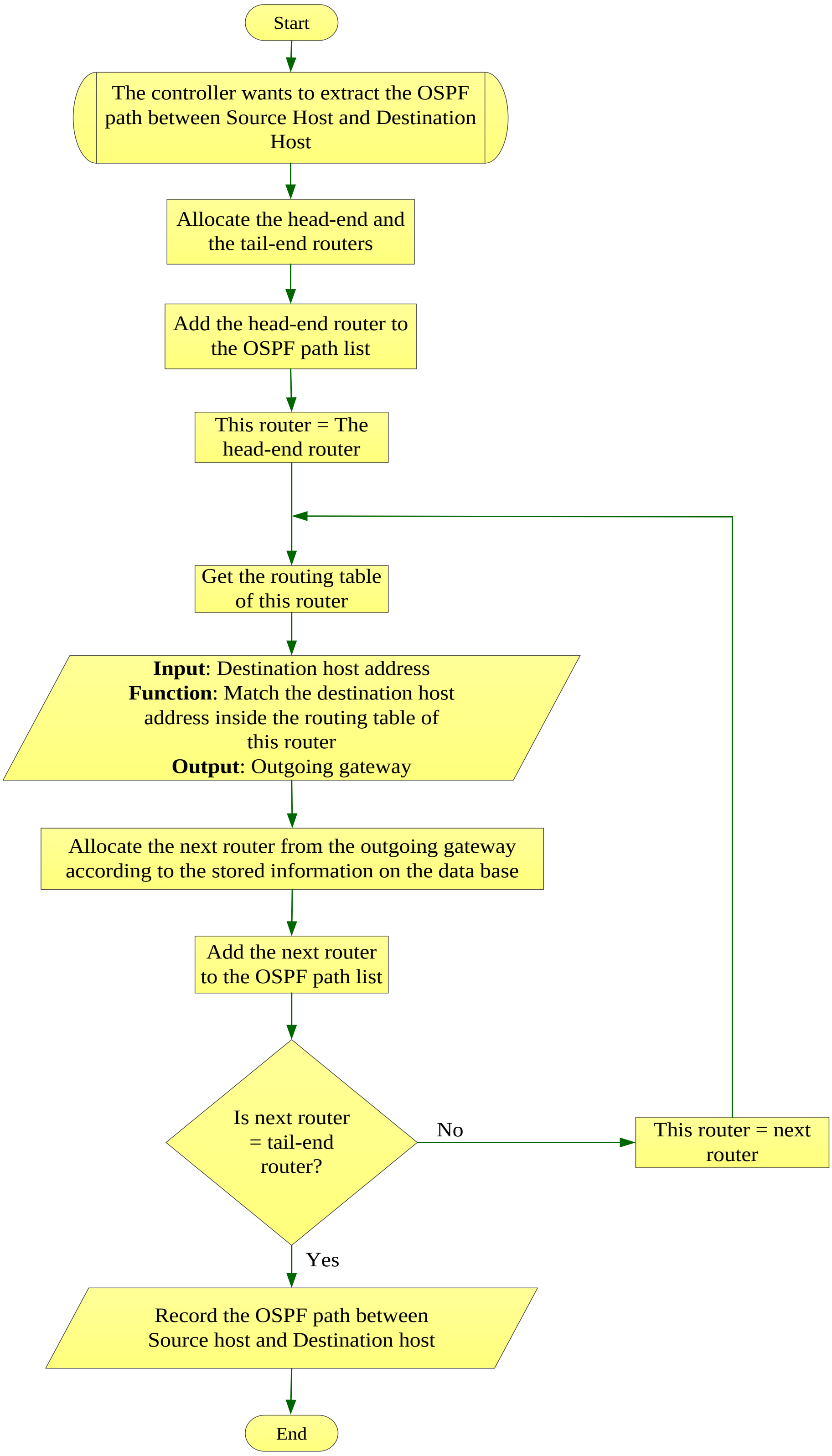

Start
The controller wants to extract the OSPF path between Source Host and Destination Host
Allocate the head-end and the tail-end routers
Add the head-end router to the OSPF path list
This router = The head-end router
Get the routing table of this router
Input: Destination host address
Function: Match the destination host address inside the routing table of this router
Output: Outgoing gateway
Allocate the next router from the outgoing gateway according to the stored information on the data base
Add the next router to the OSPF path list
Is next router = tail-end router?
No
This router = next router
Yes
Record the OSPF path between Source host and Destination host
End

Fig. 3-13 Procedure of extracting the OSPF paths inside the controller

The controller finds the OSPF paths among all existing hosts. After calculating all the OSPF paths between the hosts, the addresses of all routers and the average bandwidth of the links that the path passes through are recorded. The total bandwidth of the path is the bandwidth of the narrowest link over it. From each host source to each host destination, there is only one OSPF path. The OSPF paths inside the controller's database represent a replica of their counterparts on the network.

For example, Table III-II represents some of the OSPF paths of network shown previously in Fig. 3-3. Those paths were extracted from the routing tables and calculated by the controller then stored in its data base.

TABLE III-II
Some OSPF Paths of Network of Fig. 3-3

Source	**Destination**	**Routers along the OSPF Path**
Host 1	Host 2	1 – 3 – 4 – 2
Host 1	Host 5	1 – 5
Host 1	Host 6	1 – 5 – 6
Host 9	Host 7	9 – 5 – 6 – 7
Host 5	Host 2	5 – 6 – 2
. . .	. . .	. . .

The OSPF path from host 1 to host 2 passes through routers 1, 3, 4 and 2.

- **Calculating the non OSPF paths**

In addition to the OSPF paths, the controller calculates all possible paths from all source hosts to all destination hosts, records their addresses and bandwidth and stores them. For any *source- destination* pair, some of the non OSPF paths share several links with the OSPF path and others are completely different but the head-end router and tail end router are always common. After calculating all possible non OSPF paths, they are sorted from the shortest to the longest in terms of hops' number and stored at the controller's database. The procedure of calculating the non OSPF paths is illustrated in Fig. 3-14 below.

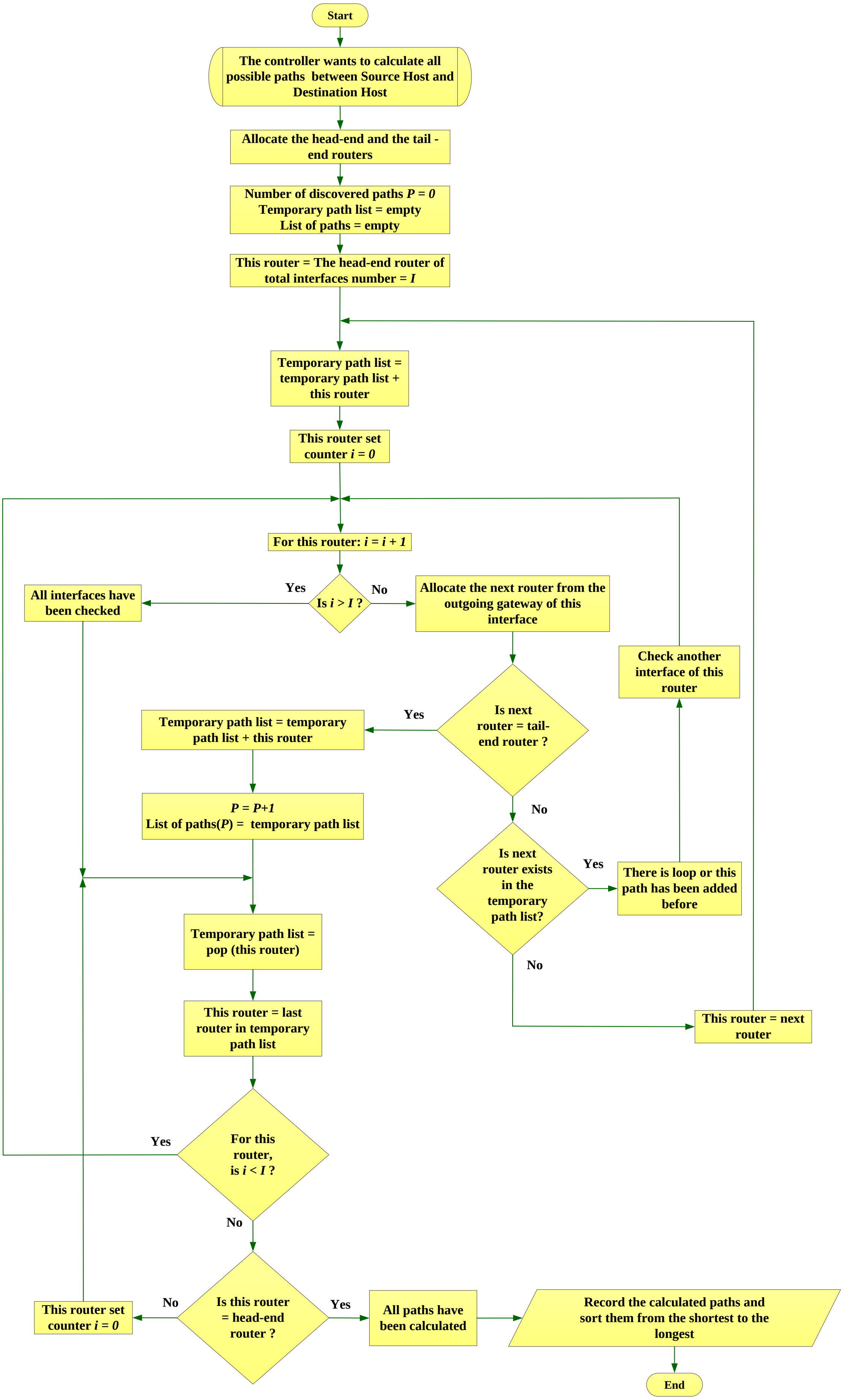

Start
The controller wants to calculate all possible paths between Source Host and Destination Host
Allocate the head-end and the tail - end routers
Number of discovered paths *P = 0*
Temporary path list = empty
List of paths = empty
This router = The head-end router of total interfaces number = *I*
Temporary path list = temporary path list + this router
This router set counter *i = 0*
For this router: *i = i + 1*
Is *i > I* ?
Yes
No
All interfaces have been checked
Allocate the next router from the outgoing gateway of this interface
Check another interface of this router
Is next router = tail-end router ?
Yes
Temporary path list = temporary path list + this router
P = P+1
List of paths(*P*) = temporary path list
No
Is next router exists in the temporary path list?
Yes
There is loop or this path has been added before
No
This router = next router
Temporary path list = pop (this router)
This router = last router in temporary path list
For this router, is *i < I* ?
Yes
No
Is this router = head-end router ?
No
This router set counter *i = 0*
Yes
All paths have been calculated
Record the calculated paths and sort them from the shortest to the longest
End

Fig. 3-14 Procedure of calculating all paths from a source host to a destination host

For example, Table III-III represents the non OSPF paths that connect some of the hosts of the network shown previously in Fig. 3-3. The controller applies the procedure shown in Fig. 3-14 along with the information of the network topology table that is obtained from its database to calculate the non OSPF paths which are also stored at its database.

TABLE III-III
SOME NON OSPF PATHS OF NETWORK OF FIG. 3-3

Source	**Destination**	**Routers along the Non OSPF Paths**
Host 1	Host 2	1 – 5 – 6 – 2
		1 – 3 – 4 – 7 – 6 – 2
		1 – 8 – 9 – 5 – 6 – 2
		1 – 8 – 9 – 5 – 6 – 7 – 4 – 2
Host 9	Host 7	9 – 8 – 1 – 5 – 6 – 7
		9 – 8 – 1 – 3 – 4 – 7
		9 – 5 – 1 – 3 – 4 – 7
		9 – 5 – 6 – 2 – 4 – 7
		9 – 8 – 1 – 3 – 4 – 2 – 6 – 7
		9 – 5 – 1 – 3 – 4 – 2 – 6 – 7
. . .	. . .	. . .

3.12.2 Real Time Monitoring

The controller is a dynamic, proactive and real time device. To achieve those features in the controller, there should be a mechanism of informing the controller about the important events taking place in the network during the simulation time. The most important events are the notable changes in the data rates of the circulating flows and the link failure. Two types of *Real time monitoring* messages inform the controller about those events: The *Flow status updating* message and the *Link failure updating* message.

i. Flow status updating message

The *Flow status updating* message informs the controller about the current status of the flow. The head-end router, which is the router where the flow enters the networking domain, creates and sends this message to the controller. The head-end router informs the controller when there is a new flow that commences or if the data rate of an existing flow changes significantly. The *Flow status updating* message contains information of *head-end router address, flow source host address, flow destination host address, flow current data rate and flow forwarding type* whether IP routed or MPLS routed. The structure of the *Flow status updating* message is shown in Fig. 3-15 below.

Flow Head-end Router Address	**Flow Source Host Address**	**Flow Destination Host Address**	**Flow Data rate**	**Flow Forwarding Type (either IP or MPLS)**	**Code**

Fig. 3-15 Flow status updating message structure

From received *Flow status updating* message and its previously stored data, the controller imagines the flows' distribution across the network.

ii. Link failure updating message

The link failure is one of the most destructive events that affect the network. It causes change in the network topology and reduction in the network resources. Both routers that are connected to the failed link are responsible of informing the controller about the failure event. The *Link failure updating* message contains information of address of *router of failed link*, *failed link address*, *failed link gate number* and the *address of the next router* of the failed link. The structure of the *Link failure updating* message is shown in Fig. 3-16.

Router Address	**Failed Link Address**	**Failed Link Gate Number**	**Next Router Address**	**Code**

Fig. 3-16 Link failure updating message structure

3.12.3 Analysis Tools and Intelligence

The purpose of the controller is to solve the network problems and to supervise the performance. Therefore, it should be an intelligent device. It is provided with Flows Distribution Test (FDT) procedure and Flows Distribution Algorithm (FDA). It adopts traffic management and restoration strategies in order to make a backbone network survivable. Both FDT and FDA are embedded inside the controller's models.

The main duty of the FDT is to check the distribution of the flows over the network while the duty of the FDA is to re-distribute the flows without congestion. Flow will be rejected from distribution over a path when one of the links along the path from source to destination does not satisfy the requested bandwidth. The summation of flows' data rates passing through a link should not exceed the capacity of the link. FDT and FDA work under fixed topology of IP and MPLS routed flows. FDT and FDA work under link failure of IP and MPLS routed flows as well.

In the situation of link failure in a network of MPLS routed flows, FDT and FDA should be preceded by applying a backup restoration procedure.

The next chapter, **chapter 4**, explains the operation of FDT and FDA in details. **Chapter 6** explains FDT and FDA along with backup restoration procedure.

3.12.4 Ability to Affect the Routers of the Network

Without affecting the operation of network's routers, the existence of the controller is in vain. The ability of the controller to affect the routers is represented by the *Control* messages it orders. Every *Control* message deals with a specific situation. To enable the controller to affect the routers, the conveying of *Real time monitoring* and the *Control* messages should be guaranteed as much as possible.

The built-in C++ code that constructs some of the router's models has been modified via sub-classing and redefining virtual member functions (without changing in its primary designed performance) to accept the controller's instructions represented by the *Control* messages. The controller is designed according to the DoD model. It consists of models that represent the DoD layers. Each model within a layer in the controller contacts its counterpart in the router and vice versa. Both the controller and the routers integrate each other.

3.13 The Secure Channel between Controller and Routers

As mentioned before, the controller as a manager of the network must obtain a real time image of the network and flows distribution. If there is any topology change or any remarkable change of a flow data rate, the controller must be notified as fast as possible. Achieving an ideal secure channel between routers and the controller is not possible. However, a channel with high transportation reliability can verify the conditions. Providing the following conditions improves the transportation reliability of the *Real time monitoring* and *Control* messages: Increasing message sending priority, using *Acknowledge* messages and using the nearest path or different paths to a destination.

3.13.1 Increasing Message Sending Priority

The *Real time monitoring* messages and the *Control* messages (from routers to the controller and vice versa) traverse the network with a sending priority higher than other messages and packets. Higher sending priority means that during forwarding operation, the router puts the messages at the front of the outgoing queue. Therefore, there is neither queueing delay nor possibility of message drop even if the queue is filled.

3.13.2 Using Acknowledge Messages

For every *Real time monitoring* or *Control* message sent or forwarded, there is a re-transmission timer at the sender and an *Acknowledge* message sent from the receiver. Re-transmission timer reaches to zero during *Retransmission Time Out (RTO)*, and upon receiving the *Acknowledge* message at the sender, the re-sending timer is cancelled. This protects against the link failure message drop. Every *Real time monitoring* and *Control* message is provided with a randomly generated *code*. The *Acknowledge* message uses the *code* to switch off the timer. If the sender does not receive any *Acknowledge* message within specific time (*RTO*), it will re-send a copy of the sent message.

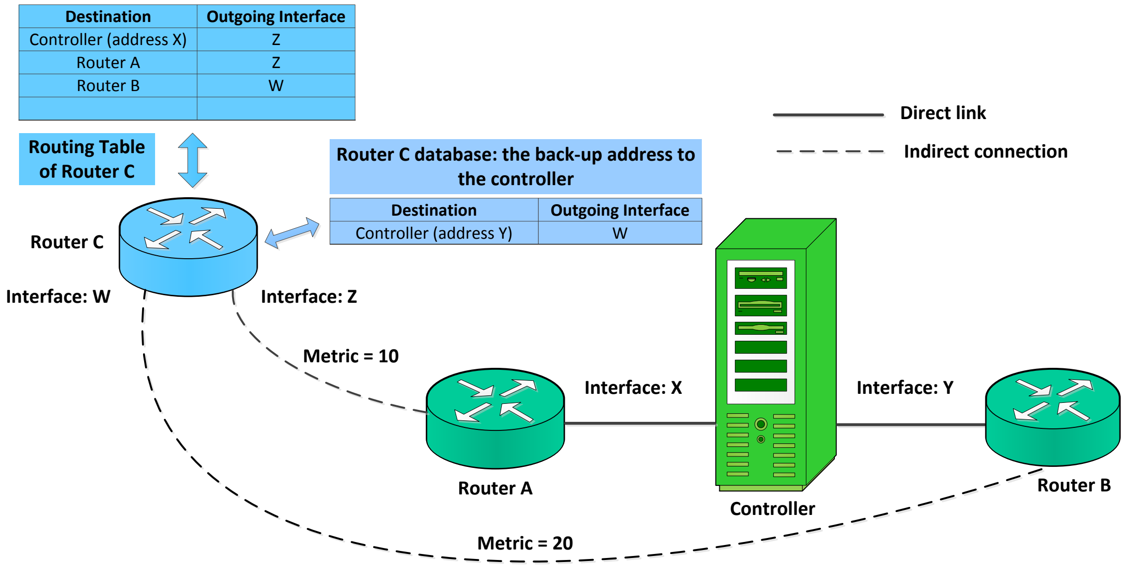
3.13.3 Using the Nearest Path or Different Paths to Reach a Destination

Connecting the controller directly to all routers is not feasible. Connecting the controller to several different routers looks more feasible as it increases the channel reliability in terms of link failure or transmission time. Routers use their OSPF routing table to send or forward the *Real time monitoring* and *Control* messages to the controller and vice versa. Using different paths to contact the controller involves each router to know all the possible addresses of the controller and to adopt the available path to the nearest one at the time.

3.14 Controller Addresses inside Router

At the beginning, the controller broadcasts its addresses to its directly connected routers which later re-broadcast them again across the network. After completing the setup of the routing tables inside all routers, each router compares the different addresses of the controller with its routing table and adopts the address of the lowest *metric* value as the main address of the controller. It also stores the other addresses as alternative (backup) addresses in its database. This adopted address represents the shortest distance to the controller. If the link of this address fails, the router uses the controller address of the next higher *metric* value.

In the fictional network of Fig. 3-17, the controller has two addresses *X* and *Y*. It is directly connected to router A and router B. Router C lays faraway form the controller. Router C can reach both the addresses of the controller but with different *metric* values. It reaches address *X* with metric value = 10 via its outgoing interface *Z* (of the less *metric* value) and reaches address *Y* with metric value = 20 via its outgoing interface *W*. Therefore, it adopts the address *X* as the main address to the controller via outgoing interface *Z* to router A and saves the address *Y* as a secondary address to the controller, which is to be used if there is a link failure along the path to *X*.



Destination	Outgoing Interface
Controller (address X)	Z
Router A	Z
Router B	W

Routing Table of Router C

Router C database: the back-up address to the controller

Destination	Outgoing Interface
Controller (address Y)	W

Fig. 3-17 Adopting the best path to the controller

For example, in our basic project network of Fig. 3-3, router 1 contacts the controller via router 3. In the cases that the link between routers 1 and 3 goes down or the link between the controller and router 3 goes down, router 1 contacts the alternative (back-up) address of the controller across routers 5 and 6.

3.15 Other Features

Some other features of our project are:

3.15.1 Addressing of Devices

Our network uses IP version 4 addressing when applying the OMNeT++ simulation; each device has its specific IP address, also each sub-network. Global IP addresses were used. However in thesis text and to simplify its explanation, routers are identified with natural numbers while hosts are represented with capital (**H**) letters followed by natural number. The controller is identified by its name.

3.15.2 Other Topologies

The network of Fig. 3-3 is not the only topology used in our project. Our project also uses other networks with more complex topologies and more number of nodes. We used the network of Fig. 3-3 because it is simple and easy to follow. However, the other topologies will be shown when they are used later.

3.16 Conclusions of Chapter 3

This chapter represented the preliminary stage of the contributions that to be added later. It illustrated the OSPF routing protocol, the SDN OpenFlow protocol and our project network model. It showed the similarities and the differences in design between our designed network project and the SDN OpenFlow. It showed the genuine and added models of the devices and their locations regarding the DoD model. It gave a brief description to the messaging system between the routers and the controller. Finally, it explained how the controller and the routers behave at the beginning of the simulation. Those additions provided the base to the contributions that to be added later to improve the network performance.

Chapter 4

Congestion Avoidance in OSPF Network by Adopting the SDN Notion

Introduction to Chapter 4

This chapter discusses the development of OSPF routing protocol performance when dealing with IP routed flows distribution suffering from congestion problems. It shows the relationship between the OSPF routing protocol and the congestion problem. It illustrates some of the approaches proposed by other researchers to deal with the congestion problem in OSPF networks. It addresses our proposed solution and methodology represented by Flows Distribution Test (FDT) and Flows Distribution Algorithm (FDA). It applies flows of high and different data rates of variable patterns upon both our basic and its counterpart improved networks. The positive improvement of the results obtained from the simulation show the validity that our solution added to the system. This chapter represents the first contribution of our research.

4.1 Network Congestion Problem

Congestion is an important factor behind the degradation of the Network Performance (NP). The effect of congestion increases queueing delay, which increases Packet Delay Variation (PDV), Latency (end to end delay), packet drop probability and error rate. This reduces the Quality of Service (QoS) of any network. QoS tools can help in mitigating most congestion problems [18]. Dropping of packets occurs when there is not enough buffer space for the packets to be en-queued into the outgoing queue of router [45].

4.1.1 Congestion and Routing

Unfortunately, the existing routing algorithms only count shortest paths between sources and destinations. Furthermore, there is a lack of cooperation between routing and congestion control [5].

4.1.2 Congestion in OSPF Networks

OSPF always uses the shortest path to forward IP packets regardless of the utilization ratio of the current shortest path. When the router is installed, the costs of a router's interfaces are set at fixed values. If the current shortest path is congested, OSPF cannot avoid forwarding traffic through it. This increases the congestion as the QoS of the network degrades greatly [6]. The shortest path a router selects for packet forwarding maybe not the best path. Hence, OSPF cannot be used to guarantee QoS as it does not have the ability to adjust the resource of the whole network. This considers the OSPF suffering from QoS related shortcomings. [65].

4.1.3 Congestion Control and Congestion Avoidance

The principle of congestion flow control is allowing a restricted number of packets to pass into the network [3] pp 602. Congestion avoidance is a type of queue management. Congestion-avoidance techniques monitor flows crossing the network to identify and avoid congestion at common network bottlenecks [18].

The Random Early Detection (RED) algorithm works by monitoring traffic load at points in the network and stochastically discarding packets if the congestion begins to increase. Weighted Random Early Detection (WRED) is the primary Cisco IOS congestion avoidance tool. WRED has the capabilities of the RED algorithm and IP precedence. It can selectively discard lower-priority traffic when the interface begins to get congested [18].

4.2 Research Problems

From the above discussion, two main problems are identified:

- The link congestion problem.
- The inefficient bandwidth usage problem resulted from the un-even distribution of the flows among the existing paths.

4.3 Research Objectives

The aim of this chapter is to enhance the performance of the Open Shortest Routing Protocol (OSPF) network by adding some features of the Software Defined Networking (SDN) in a modified approach to improve some network's performance metrics like link congestion, packet loss, packet delay variations and total network delay. The improvement of those metrics leads to an improvement in the total throughput of the network. This approach has been achieved by extending an OSPF (layer 3) network to adopt a methodology similar in many aspects to that used in SDN (layer 2) networks (the existence of the controller and the messaging system between it and the routers).

The new OSPF network contains a real time dynamic supervisory controller, which is capable of detecting the location of a congestion that may take place before or at the brink of its occurrence and dealing with selected flows on selected routers across the network in a way that prevents the congestion by applying a smart heuristic Flows Distribution Algorithm (FDA). The privileges of this method are:

- Semi-central management provides better control on flows distribution.
- Better usage of network resources especially the bandwidth.

Extending the OSPF network to adopt this methodology involves modifying the hardware of the OSPF routers (by adding extra models) as well as designing a controller that is compatible with them. This modification was illustrated in **chapter 3** before.

4.4 OSPF and QoS Development Efforts

Many studies have been conducted to develop the performance of the OSPF protocol in terms of flows distribution and bandwidth usage.

4.4.1 OSPF Development Efforts

In [5], Al-Shabibi et al. suggest a congestion aware multi-path routing protocol in a multi-route network, where there is no central controller to manage the traffic and the amount of traffic assigned to each path is determined by each router in response to the congestion signals received from other routers along the path. Every router sends its updated information to the neighbouring routers when there is any significant change detected.

In [66], Oki et al. propose the Smart Open Shortest Path First (S-OSPF) load balance algorithm based on a model where the traffic is split only at source edge nodes and distributed to the neighbour nodes with optimum ratios. From the neighbour nodes, the traffic is routed according to the OSPF protocol. The optimal traffic distribution is obtained through Linear Programming (LP) by using decision variables like network congestion ratio and traffic portion from parameters of traffic demand and link capacity. The traffic is not split over multiple paths.

In [67], Antic et al. analyze Load Balancing using Shortest Path Routing Protocol (LB-SPR) for arbitrary traffic patterns. The optimization takes in consideration the weights assigned to the network nodes according to their assessed load demand. Traffic is routed over two phases. First phase starts from source node to an intermediate node. Second phase starts from the intermediate node to the destination node. This solution is successful if the load demand per node is estimated. However, it does not deal with the erratic conditions.

The load sensitive routing algorithm is proposed by Anirudha Sahoo in [68] which is based on Dijkstra’s shortest path algorithm. The load sensitive routing is invoked when load on the outgoing link of the router reaches a certain threshold. Load sensitive routing tries to find an alternate next hop for the packets that may be transiting through the congested link. The method that is used for finding the next hop is based on its OSPF properties. Each node runs Dijkstra’s algorithm to build two kinds of routing tables. The first one is from itself to all other nodes in the network (*active* routing table) while the second are from its neighbours to other destinations (*passive* routing tables). When LSR flag is *TRUE* the next hop node would be the one calculated by the LSR algorithm, otherwise it would be the one found by OSPF. The strength of our algorithm compared to load sensitive routing is that our algorithm recognizes the congestion before or at the brink of its occurrence as it has global monitoring over the network.

The Cost Adaptive OSPF (CA-OSPF) has been proposed in [6]. Haijun et al. suggest that the interface's cost is dynamically adjusted according to the utilization ratio of the interface's bandwidth. CA-OSPF sets two thresholds for the interface: upper limit and lower limit. When the interface bandwidth utilization ratio exceeds the upper limit (over used state), the router will increase the interface's cost and when the interface bandwidth utilization ratio falls below the lower limit (under used state), the router will reduce the interface's cost. For both cases, the router generates Link state Advertisement messages (OSPF-LSA) to inform other routers within its area to update their routing tables.

A Wavenet-Based Dual-Path Congestion Control Routing Mechanism (WBDPCCRM) [69] is introduced by extending the conventional open shortest path first (OSPF) routing protocol from a single-path routing protocol to a dual-path routing protocol. This mechanism applies the wavelet neural network to anticipate the state of any link. A least congested shortest path is added to the original OSPF path between any *source-destination* node-pair to control congestion. The load is distributed over both paths. The least congested shortest path is an adaptive path that considers the congestion state of each network link, while it determines forwarding paths that delivers packets from source to destination through it. The second path calculation is performed by using WBDPCCRM mechanism and during the system operation with an adaptive manner.

Another development method to the OSPF is by using Local-Unicast Routing Control Agent (L-URCA) [7]. L-URCA uses local information to dynamically update the OSPF link costs to re-route traffic away from congested or highly utilized links. Routers reroute traffic according to the updated link costs and over the shortest path. The L-URCA heuristics applies robust optimization, which incorporates uncertainty parameters to trace the lack of information related to the load and capacities of links in the network graph. The higher the uncertainty parameters for a path, the higher the difference in the actual path and link delay from expected delay based on solution of the nominal problem. Delay mainly represents the queuing delay.

In [65], Tiwari et al, develop the OSPF routing protocol by proposing the Local Load Sensitive Routing (L-LSR) Protocol. The L-LSR is a routing protocol that uses alternate paths to provide QoS along OSPF paths. When a node experiences congestion on any outgoing link, it informs its neighbours about the congestion by sending congestion notification to all of them except the neighbour connected to it over the congested link. The neighbouring nodes cooperate with this node to forward packets through alternate paths. The alternate paths are chosen in such a way that avoids loops.

All previously mentioned development efforts are applied at routers, which involve flooding the updates of the routing tables across the network continuously to provide all routers with full image of the network. This consumes significant portion of bandwidth as well as processing time until reaching to the stable state. Our improvement is applied at the controller and some involved routers. This provides semi-central management and less bandwidth consumption by the *Flow status updating* messages and the limited number of the *Control* messages. It does not involve changing the interface's cost which affects the distribution of all flows. The more often are weights between nodes changed, the more the network becomes unstable [70]. The previously mentioned development efforts also deal with congestion after its occurrence while our improvement identifies the congestion and deals with it as fast as possible.

4.4.2 General Congestion Control Development Efforts

The improved Weighted Random Early Detection (WRED) algorithm is suggested in [71] where Peculea et al propose a framework that allows designing and testing different algorithms' congestion avoidance in a physical test network. The main idea behind this work is a method for the dynamic adjustment of the queue's length function of their average queue size. All packets that cross the network are captured and analyzed. According to the collected information, the packets are classified into their corresponding class of traffic. The improved WRED consists of two components. First component, defines the degree of burst that will be permitted in the gateway queue for computing the average queue size. The second component, determines how frequently the gateway marks packets, given the present level of congestion.

In [4], Liu et al. present a Congestion Location Detection (CLD) algorithm that allows an end host to detect whether congestion occurs at the local access link or at more remote links. The CLD is based on queueing delay patterns. If many flows see synchronized congestion, then, the local link is the congested link where most flows are experiencing high delays at a similar level; otherwise, CLD considers the congestion to be remote. Compared to CLD which is re-active algorithm, our approach proposes a pro-active algorithm that can exactly allocate the location and the amount of congestion by using the capabilities of the controller.

4.4.3 SDN and OSPF Together

Using the OSPF under the SDN or combining both technologies has been studied in several approaches. The possibility of building and operating networks in transition from existing infrastructure, where both legacy and SDN devices can exist together is discussed in [72]. This combination involves an interoperability and integration between control plane of SDN networks and IP control plane of legacy networks. Chemalamarri et al aim to solve problem of bridging legacy control plane with the SDN control plane at Layer 3 by using OSPF routing protocol in communication between legacy plane and SDN controller. The proposed architecture for Hybrid Software Defined Networks (SYMPHONY) sets communication between the SDN domain and the legacy network via a legacy route server connected to the SDN controller. The legacy route server acts as central repository that stores topology information of the legacy network. The name – SYMPHONY is obtained from the hybrid SDN controller that orchestrates legacy and SDN control domains.

In [73], Caria et al propose a method of hybrid SDN/OSPF operation. This method uses SDN nodes to partition the initial OSPF domain into sub-domains, as a result of that achieving the traffic engineering capabilities comparable to full SDN operation. The SDN nodes start the update process in the individual sub-domains by flooding routing updates that are separately tuned per sub-domain. The routing inside each sub-domain is only based on OSPF so that it remains stable and unchanged at all times. The inter-sub-domain paths can be optimized by marking the paths in each traversed sub-domain.

In [74], Nakahodo et al. propose Hybrid Software Defined Networking (H-SDN) where the SDN system is implemented with conventional network. SDN is set only over the edge nodes to implement Smart OSPF (S-OSPF). Two Virtual Machines (VMs) have been constructed on hybrid edge routers. VM1 represents the ordinary OSPF routing and VM2 represents the OpenFlow switch. OpenFlow Switch on VM2 uses Open vSwitch (OVS) adjustments of the amount of data by using flow’s lifetime period. For packets received from other intermediate nodes, Hybrid-Edge router must obey to OSPF table. VM2 is operating Open vSwitch. VM1 connected VM2 with a virtual Link. It is used to address the transfer of data from OVS without interference of OSPF table. To address OSPF routing issue, VM2 must know OSPF routing table.

Our approach extends the conventional OSPF to utilize a methodology similar to that of the SDN in dealing with the congestion problem. Is also represents some kind of collaboration between the conventional networking and the SDN. However, none of the SDN devices or SDN protocols is used in it as mentioned before.

4.4.4 Other QoS Improvement Efforts

The average residual bandwidth in the path as a localized QoS routing metric is proposed in the Bandwidth-Based Routing (BBR) scheme [75].The quality of the path is measured by calculating the average residual bandwidth for each candidate path. The path with the highest average residual bandwidth is used to route the incoming flow. BBR is applied at the source node, which uses a setup message to travel along the selected path with each connection request. Each intermediate node tests the outgoing link's residual bandwidth to verify the ability of the link to satisfy the requested bandwidth. If there is sufficient bandwidth on the outgoing link, the requested bandwidth is reserved for that connection and the message is sent to the next hop. If any link along the path does not support the requested bandwidth, a failure message occurs.

The privilege of our approach regarding the BBR is that the controller takes in consideration both the length of the path and the residual bandwidth along it. It prefers the OSFP path and diverts to another path only when necessary to avoid congestion. As the controller knows the flows' distribution in real time across the network, there is no need to use any setup messages or failure messages to examine the path.

In [76], Chin et al propose the Largest Widest Shortest Path among Limited Choices (LWSP-LC) which performance is depended on the size of limited choices. The LWSP-LC develops a simple mathematical model to derive the value of the upper bound of the size of limited choices for engineering design. It searches a path only from limited choices instead of all possible choices. The LWSP-LC selects a path with lower hop count and higher route bandwidth among all available connected paths. If more than one path has less hop count and higher route bandwidth, the LWSP-LC choses a path with the highest total available bandwidth.

The privilege of our approach regarding the LWSP-LC is that the controller takes in consideration all the available paths along with applied traffic pattern when choosing the best path to distribute a flow when dealing with congestion. Some of the paths that excluded from being chosen by the LWSP-LC may serve the network in more efficient way.

4.5 The Basic Operation of the Network

The network behaves as a normal OSPF network in terms of building of OSPF routing tables and forwarding traffic. Under high applied traffic, the network activates the rule of the controller in re-distributing some flows to deal with the congestion problem (if the latter exists).

4.5.1 Status of Network Topology

During the simulation, the network topology remains fixed. Neither link failure nor node failure occurs during the simulation time period. Thus, there is no change to the basic OSPF routing tables of the routers.

4.5.2 Traffic over the Network

The traffic is generated by letting the source host choose the destination host from amongst all hosts except itself with changeable data rate through the simulation time. *Data* packets employ UDP transport layer protocol. Our project supposes that the applied flows' data rates are unpredictable. Therefore, the controller should be ready to deal with any situation.

The hosts generate the *Data* packets at constant generation timing of one packet at every 0.0001 second per destination. The length of the packets is changeable which represents the changeable flow's data rate.

4.5.3 Measurement of Performance Improvement

The network design is applied into two identical networks, the first network is provided with controller that enhances the performance of the network by using its FDA and the second network is not provided with any controller (represents a traditional OSPF network).

Same load is applied over both networks over several different cases. Each case represents a specific load pattern that causes a problem in the traditional OSPF network. The controller solves the problem. The improvement of the obtained results represents the difference in performance between the network supplied with controller and its counterpart of without controller.

4.6 The Modified OSPF Router

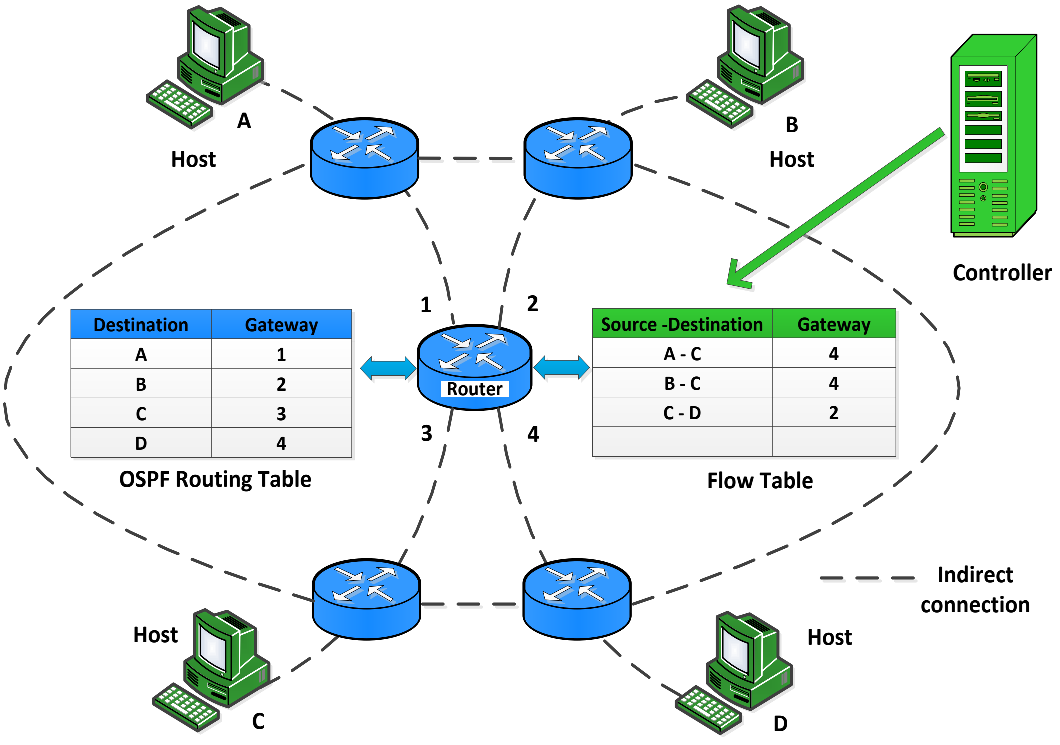
The router design represents the standard OSPF router of the Inet-OMNeT project. It is also developed with modifications in code and several extra added models to be compatible with the network requirements as explained in **chapter 3**.

4.6.1 General Description

The main function of a network layer of the OSI model (the internet layer of the DoD model) is to route packets from the source machine to the destination machine [77]. The developed router has two forwarding tables; the OSPF routing table and the flow table. Both of them are invoked by the DoD internet layer (OSI network layer). Similar to any OSPF router, the router builds its OSPF routing table by exchanging information with its neighbours and responds to the commands of the controller in terms of IP routed flows. This response is represented by constructing the flow table. Router either uses OSPF routing table or flow table to forward an IP routed packet.

4.6.2 The Flow Table Container Model of the Router

The *Flow table* model is an additional model used by the network layer. The flow table is a temporary container and has higher priority than the OSPF routing table as it substitutes the routing table in forwarding specific flows. The router establishes or removes the flow table in response to a *Control* message issued by the controller. The structure of the flow table of a router inside fictional network is as shown in Fig. 4-1.



Destination	Gateway
A	1
B	2
C	3
D	4

OSPF Routing Table

Source -Destination	Gateway
A - C	4
B - C	4
C - D	2

Flow Table

Fig. 4-1 Flow table structure of a router within fictional network

4.6.3 Routing Table vs. Flow Table

The differences between the routing table and the flow table are:

- The routing table is constructed according to the OSPF routing protocol through the exchanged OSPF messages while the flow table is constructed according to the controller's instructions.
- The routing table does not change unless there is a topology change while the flow table can change according to the change in flows' pattern.
- When forwarding a data packet, the routing table matches the destination address only, while the flow table matches both the source and the destination addresses.
- The flow table has higher priority than the routing table.
- Finally, the routing table contains all destinations and it is larger in size than the flow table which deals with specific flows directed to specific destinations.

4.6.4 Data Packets Forwarding Scenario Inside the Router

If there is a flow table constructed inside the router, the forwarding operation of flows' *Data* packets passes through the following procedure:

- For every *Data* flow packet that arrives at the router, the *Developed IP* model of the network layer dismantles the packet. It extracts the flow *source-destination* pair from the header of the packet.
- The *Developed IP* model then matches the flow *source-destination* pair with any of the addresses of the flow table.
- If the result of the matching operation is positive, the router forwards the flow according to the output gateway of the flow table that meets the *source-destination* addresses.
- If the result of the matching operation is negative, the router continues to forward the packet of the flow according to the OSPF routing table by matching the *destination* address only.
- If the destination address of the packet does not exist in the OSPF routing table, the router discards the packet as usual.

In the case where there is no flow table inside the router, the router directly forwards the incoming flow according to the routing table.

4.6.5 Methodology in Dealing with Data Packets of a Flow

After the routers complete building of their routing tables and the controller finishes the setup of its database and algorithms, many of the source hosts commence sending flows consisting of *Data* packets to other destination hosts. The source host informs the head-end router the forwarding method of forwarding whether IP or MPLS.

At commence of a new flow or when there has been an obvious change in data rate of an existing flow, the following procedure takes place:

The head-end router, the node where a flow enters the networking domain, analyses the received packet at its *Developed IP* model which is part of the *Network layer* model. It dismantles the head of the packet to extract the *source-destination* pair of the flow. If the *source–destination* pair does not exist in the stored database of the router, this means that this flow is a newly started flow. The head-end router calculates the length of the packet and stores the new flow information in its data base. The head-end router also creates a *Flow status updating* message, inserts the new flow information into it and sends it to the controller. The form of *Flow status updating* message was shown in Fig. 3-15 of **chapter 3**.

If the *source–destination* pair exists in the stored database of the router, this means that this flow exists and was previously identified to the controller. Thus, the head-end router compares the length of the flow packet with its counterpart stored at the data base. If there is an obvious difference between them, it stores the updated flow information in its data base. The head-end router also creates a *Flow status updating* message, inserts the updated flow information into it and sends it to the controller.

In the case where there is no obvious difference between the length of the current flow packet and its previous one (the difference is less than 5% from the stored value on the router's data base), this means that there is no change in flow's data rate and the head-end router does not create any *Flow status updating* message.

In all cases, the head-end router forwards the packets of the flow by default according to the OSPF routing table (or according to the flow table if the flow was previously diverted) through the appropriate gate.

When a router calculates the length of the packet in its *Developed IP* model, it takes into consideration (adds) the number of bits that to be added when the packet is encapsulated at the next PPP model. The added bits include: 8bits (flag), 8bits (address), 16 bits (control) and 16 bits (Frame check sequence) [18]. The total calculated length of the packet is very equal to that practically crossing the link of the physical layer.

4.7 The Analysis Tools of the Controller

The controller is the device that has the responsibility of monitoring the traffic across the network and it deals with any congestion problem. It is a real time device that has the ability of dealing with an arbitrary changing traffic. The structure of the controller represents the DoD model as explained in **chapter 3** before. The basic operation of the controller depends on Flows Distribution Test (FDT) and Flows Distribution Algorithm (FDA).

The controller activates FDT to check the distribution of the flows over the pre-determined paths and if there is any congestion problem detected, it activates FDA to solve it. Flow will be rejected from distribution over a path when at least one of the links along the path from source to destination does not satisfy the requested bandwidth. FDA re-distributes specific flows to remove the congestion. Both FDT and FDA are applied inside the controller as they share some of its models. The models of the controller that contain FDA and FDT are:

- *Flows Manage* model.
- *Flows Checker* model.
- *Flow Exchange Algorithm* model.
- *Remove Flow* model.
- *Flow Re-Divert* model.
- *Flow Compare* model.

The decision of the controller is represented by the *Control* message which can be either a *Confirmation decision* message or a *Flow table decision* message which are created at the *Inform decision* model, encapsulated at the *UDP* and the *Network layer* models and transmitted to outside from the *Ethernet* model. Fig. 4-2 shows the structure of the *Confirmation decision* message.

Flow Source	**Flow Destination**	**Head-end Router Address**	**code**

Fig. 4-2 Structure of the controller confirmation decision message

The structure of the controller *Flow table decision* message is shown later in Fig. 4-6.

4.8 Flows Distribution Test (FDT) Operation

When the controller receives the *Flow status updating* message from the router, it extracts the flow's information from it. Flows information consists of: flow's source address, flow's destination address and flow's data rate. The controller internally examines the network behaviour against the flow (the flow forwarding operation across the routers according to the pre-stored information). The examination operation is called Flows Distribution Test (FDT). According to this operation, the controller notices if there is any congestion that may occur at any link inside the network due to the flow distribution. FDT is passing through the following procedure:

- Define a network with specific topology. The network consists of number of nodes (routers) and number of links.
- Define ***A*** and ***B*** as two nodes that belong to the network.
- Define a flow of specific data rate commencing from node ***A*** and terminating at node ***B*** passing through some other nodes and links on its way from ***A*** to ***B***.
- The data rate of flow ***A-B*** suffers from an obvious data rate change (either increase or decrease).
- FDT checks the current distribution of flow ***A-B*** by identifying the currently used path to forward flow ***A-B***.
- If flow ***A-B*** is currently passing through the OSPF path, FDT applies the steps of **section 4.8.1**.
- Else if the flow is currently passing through any of the non OSPF paths, FDT applies the steps of **section 4.8.2.**

4.8.1 FDT Operation on OSPF Path Distribution

This section is for testing flows that are just begin sending *Data* packets or flows that are currently passing through the OSPF paths and suffer from obvious changes in their data rates.

- If flow ***A-B*** that suffers from change in data rate is passing via the OSPF path, FDT checks if it is available to continue forwarding it with its new data rate via the OSPF path without congestion.
- FDT Identifies the path where the flow is currently passing through (the path consist of all nodes and links that the flow passing through from source node ***A*** to destination node ***B*** and in this case it is the OSPF path).
- FDT replaces the flow's previous information (old data rate) from each link via the identified path with the flow's current information (new data rate). FDT tests the residual bandwidth of each link via the path.
- If the residual bandwidth of all links over the path is greater than zero, this means that the path can handle the flow without any congestion. FDT adopts the OSPF path between nodes ***A*** and ***B*** as the main employed path that used to transport the *Data* packets of flow ***A-B***. The controller sends the head-end router (the router where flow ***A-B*** commences) an *Approval control* message to confirm using the path. FDT terminates the operation.
- If the residual bandwidth of one link or more over the path is less than zero, this means that the OSPF path cannot handle the flow without any congestion. Therefore, it is better to choose an alternative path to forward the flow. The new path should avoid the congested links. FDT activates FDA of **section 4.10** to solve the congestion problem.

4.8.2 FDT Operation on None OSPF Path Distribution

This section is for testing flows that are currently passing through non OSPF paths and suffer from obvious changes in their data rates. If a flow that suffers from a change in its data rate is passing via non OSPF path (previously diverted), FDT checks if it is available to re-forward it with its new data rate via the OSPF path without congestion and if possible, it activates the flow table removal procedure. If it is not possible to forward the flow over the OSPF path, FDT tests the possibility of carrying on forwarding it over the currently used path.

- FDT identifies the path where flow ***A-B*** is currently passing through (in this case it is one of the non OSPF paths).

- FDT subtracts the flow's previous information (old data rate) from the other flows passing through each link via the identified none OSPF path.
- FDT identifies the OSPF path between nodes ***A*** and ***B***.
- FDT adds the flow's current information (new data rate) to the other flows passing through each link via the OSPF path. Then, it tests the residual bandwidth of each link via the path.
- If the residual bandwidth of all links over the OSPF path is greater than zero, this means that the OSPF path can handle flow ***A-B*** without any congestion. FDT adopts the OSPF path between nodes ***A*** and ***B*** as the main employed path that used to transport the *Data* packets of flow ***A-B***. The controller sends a *Flow table decision* message of *Removal flow table* to the head-end router to remove the existing flow table regarding the flow. This returns flow ***A-B*** forwarding from the non OSPF path to the OSPF path. The procedure of creating and removing the flow table is illustrated later in **section 4.10.1**. After returning flow ***A-B*** to be forwarded via the OSPF path, FDT terminates the operation.
- If the residual bandwidth of one link or more over the OSPF path is less than zero, this means that flow ***A-B*** cannot be forwarded via the OSPF path without congestion. Then, FDT checks the availability of continue forwarding flow ***A-B*** via the currently used none OSPF path.
- FDT adds the flow's current information (new data rate) to the other flows passing through each link via the currently used none OSPF path. Then, it tests the residual bandwidth of each link across it.
- If the residual bandwidth of all links over the path is greater than zero, this means that the currently used none OSPF path can handle the flow without any congestion. FDT adopts the currently used none OSPF path between nodes ***A*** and ***B*** as the main employed path that used to transport the *Data* packets of flow ***A-B***. The controller informs the head-end router to continue forwarding via the currently used none OSPF path by sending a *Confirmation decision* message. FDT terminates the operation.
- If the residual bandwidth of one link or more via the currently used none OSPF path is less than zero, this means that it cannot handle flow ***A-B*** without any congestion. Therefore, it is better to choose an alternative path to forward the flow. The new path should avoid the congested links. FDT activates FDA of **sections 4.10.2 - 4.10.5** to solve the congestion problem. The FDT procedure is also shown in Fig. 4-3.

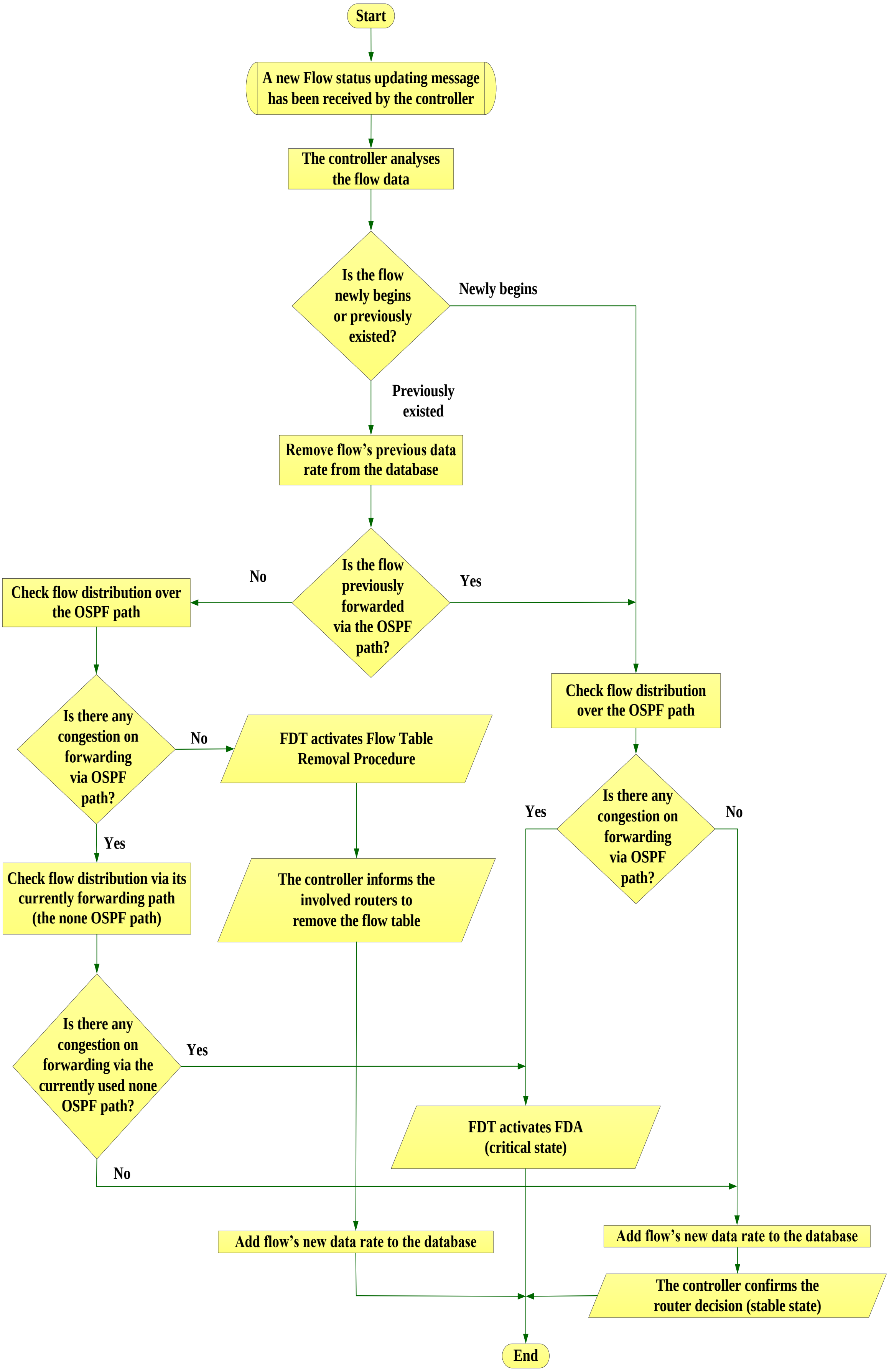

Start
A new Flow status updating message has been received by the controller
The controller analyses the flow data
Is the flow newly begins or previously existed?
Newly begins
Previously existed
Remove flow's previous data rate from the database
Is the flow previously forwarded via the OSPF path?
No
Yes
Check flow distribution over the OSPF path
Check flow distribution over the OSPF path
Is there any congestion on forwarding via OSPF path?
No
FDT activates Flow Table Removal Procedure
Yes
No
Is there any congestion on forwarding via OSPF path?
Yes
Check flow distribution via its currently forwarding path (the none OSPF path)
The controller informs the involved routers to remove the flow table
Is there any congestion on forwarding via the currently used none OSPF path?
Yes
FDT activates FDA (critical state)
No
Add flow's new data rate to the database
Add flow's new data rate to the database
The controller confirms the router decision (stable state)
End

Fig. 4-3 FDT algorithm procedure

4.9 Normal Flows Distribution (Stable State)

The normal flows' distribution state represents the best network condition. The controller always drives the network to reach the stable state. In this state, there is no possible congestion to occur at any link inside the network. All the links over the path where the flow is supposed to pass can easily handle the flow.

- Define a network with specific topology and consists of number of nodes and number of links.
- FDT tests the residual bandwidth of each link over the network.
- If the residual bandwidth of all links is greater than zero, this means that the network is in stable state.
- Hence, the recorded packet loss is almost zero. The bandwidth usage is efficient. The head-end routers as well as the other routers over the network continue forwarding flows' *Data* packets according to their OSPF routing tables or according to the previously defined paths without any problem.

Example, suppose network of Fig. 4-4 where host 1 (H1) commences sending data packets of specific data rate to host 2 (H2). H1 is directly connected to router 1.

Router 1, the head-end router of flow H1-H2, informs the controller about flow (H1–H2) by using a *Flow status updating* message. When receiving the *Flow status updating* message, the controller de-capsulates it and extracts the OSPF path that flow H1-H2 should passes through from its data base. The path passes through routers 1-3-4-2.

The controller applies its FDT to test the congestion over the links on the path of flow H1-H2. The result is: No congestion at any link. Thus, the controller confirms the forwarding decision of router 1 regarding flow H1-H2 by sending a *Confirmation decision* message to router 1. The routers over the H1-H2 OSPF path continue forwarding flow H1-H2 according to their OSPF routing table.

After a while, host 4 (H4) increases its data rate that is directed to host 8 (H8). Router 4 informs the controller about the current situation of flow H4-H8. The controller extracts from its data base the stored distribution path of flow H4-H8. Flow H4-H8 was previously forwarded via the OSPF path passing through routers 4-3-1-8. FDT removes the flow's old data rate from the path then calculates the congestion status by adding the current data rate to the path. The result is: No congestion at any link. The controller sends router 4 a *Confirmation decision* message to continue forwarding flow H4-H8 according to the OSPF routing table via the OSPF path and stores the current distribution status of flow H4-H8 in its data base. Fig. 4-4 shows flows normal distribution inside 9 routers network.

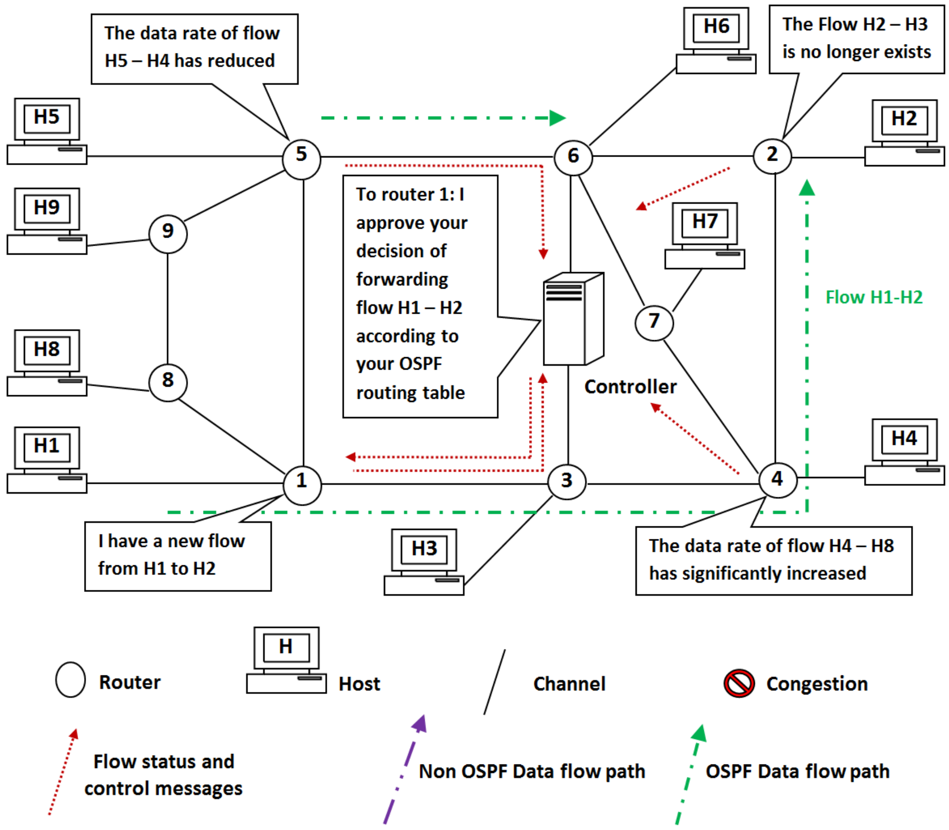

The data rate of flow H5 – H4 has reduced
The Flow H2 – H3 is no longer exists
H5
H6
H2
H9
H7
H8
H1
H4
H3
5
6
2
9
7
8
1
3
4
To router 1: I approve your decision of forwarding flow H1 – H2 according to your OSPF routing table
Flow H1-H2
Controller
I have a new flow from H1 to H2
The data rate of flow H4 – H8 has significantly increased
Router
H
Host
Channel
Congestion
Flow status and control messages
Non OSPF Data flow path
OSPF Data flow path

Fig. 4-4 Flows normal distribution inside 9 routers network

The *Confirmation decision* message is a *Control* message. It also can be considered as some kind of *Acknowledge* message and it is created at the *Inform decision* model of the controller.

The router receives *Confirmation decision* message. This stops the timer from re-sending a copy from the previously sent *Flow status updating* message to the controller.

This situation represents low to moderate applied data rates. The controller has no effect on router’s forwarding decision as it only collects data, investigates the flows distribution, stores the data at its database and confirms the head-end router decision. Neither congestion nor packet drop has been recorded.

In the network of low applied traffic where there is no rule to the controller, the recorded throughput is the same for both simulations of with and without controller. Fig. 4-5 shows the throughput line which is almost identical for both simulations.

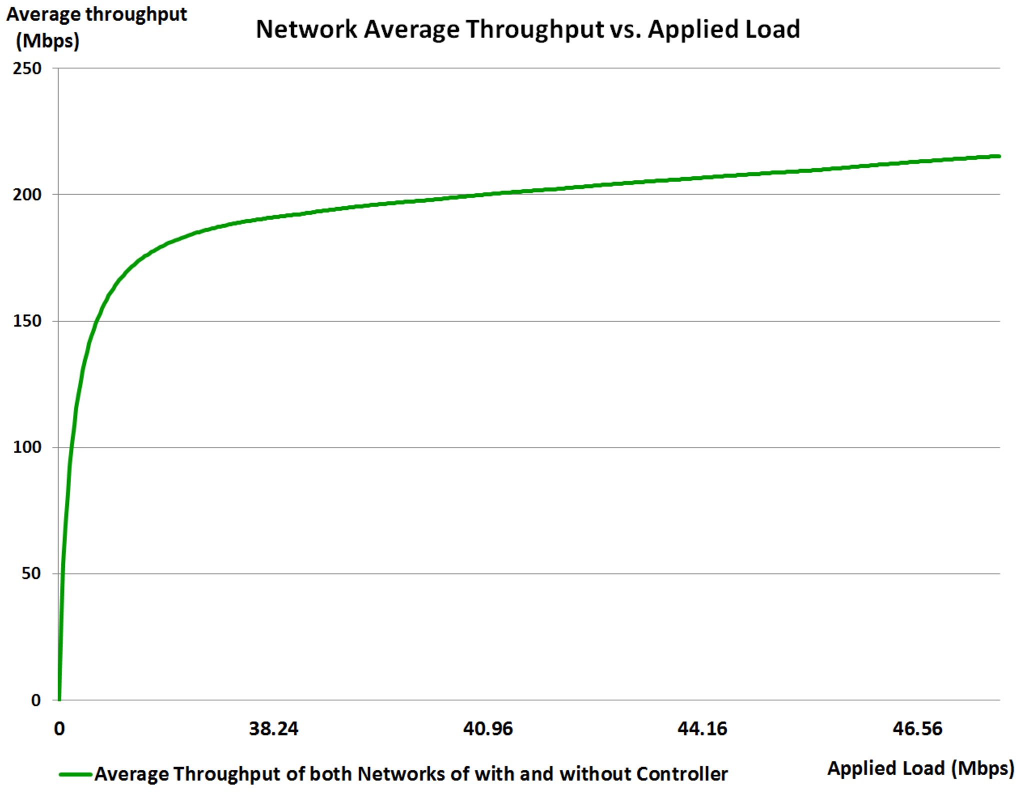


Fig. 4-5 Network throughput recorded on low applied data rate

4.10 Congested Flows Re-Distribution (Critical State and FDA)

The flows travelling across the network are not always sent on low or moderate data rates. The data rate of some flows may increase rapidly from the sender host. By notifying the controller about the increase of any flow data rate, the controller repeats applying FDT by removing the old flow data from its data base and examining the effect of the distribution of flow with new data rate and notices the congestion occurrence over one or more links inside the network.

- Define a network with specific topology and consists of number of nodes and number of links.
- FDT tests the residual bandwidth of each link over the network.
- If the residual bandwidth of one link or more over the network is less than zero, this means that there is a congestion problem somewhere inside the network and the network is in critical state. The duty of the controller is to drive the network to reach the stable state.
- FDT activates FDA to solve the problem. FDA employs one of four placements subsequently:
    - Diverting the flow of high data rate.
    - Diverting the non-high data rate flow.
    - Re-Diverting the previously diverted flow.
    - Multiple flows diverting.

If the current placement solves the congestion problem, the controller applies it without continuation to the next placement. The result of the placement is changing the decision of specific routers related to forwarding specific flows. The controller creates a *Flow table decision* message and sends it to the head-end router, which responds to it by applying the specific procedure of creating the required flow table.

4.10.1 Flow Table Creation/Removal Procedure

The controller informs the head-end router to divert the flow over specific bypass path by sending a *Flow table decision message/Establish flow table* to create a flow table inside the router. The *Flow table decision* message is type of the *Control* messages mentioned in **chapter 3**. Its structure is illustrated in Fig. 4-6.

Flow Source	**Flow Destination**	**Flow Path:** $\boldsymbol{N_1\ N_2\ N_3\ N_4\ \dots\dots\dots\dots N_k}$	**Establish/ Remove**	**code**

Fig. 4-6 Flow table decision message structure

Flow Source: Host address where the flow starts.

Flow Destination: Host address where the flow ends.

$\boldsymbol{N_1\ N_2\ N_3\ N_4}\ \dots\dots\dots\dots\dots \boldsymbol{N_k}$: Addresses of routers where the flow path passes through.

$\boldsymbol{N_1}$: The address of the head-end router.

$\boldsymbol{N_k}$: The address of the tail-end router.

Establish/ Remove bit: If *True* then the order is to establish a flow table over the assigned path else if *False,* the order is to remove the flow table related to the path.

code: Random number created by the controller to verify that the *Control* messages are received successfully. The controller stores the flow *code* when it establishes the flow table and uses the flow *code* again to remove the flow table.

- The controller creates a *Flow table decision* message and loads it with the flow information, flow path and code. It sets the Establish/ Remove field to *True*.
- The controller sends the *Flow table decision* message to the head-end router (address = N_1) as it is the first router in the path.
- The head-end router dismantles the *Flow table decision* message, allocates its location inside the received flow path and compares the address of the next router over the received flow path with the address of the next router according to its previously established OSPF routing table when dealing with same flow.
- If the comparing result is identical, the head-end router forwards the *Flow table decision* message to the next router of the flow table (N_2).
- If the comparing result is different, the head-end router creates a temporary flow table to divert the flow through, marks the flow table with the flow code and forwards the *Flow table decision* message to the next router of the flow path (N_2).
- Upon receiving the *Flow table decision* message from the previous router, the current router repeats the comparing and the flow table creating operations (similar to the procedure at the head-end router), forwards the *Flow table*

decision message to the next router and sends an *Acknowledge* message that contains the flow code to the previous router to confirm receiving the *Flow table decision* message.

- This operation is repeated until reaching the tail-end router.
- Upon receiving the *Flow table decision* message from the pen-ultimate router, the tail-end router (address = N_k) informs both the controller and the pen-ultimate router that the establishment of the temporary flow path has been completed by sending two *Acknowledge* messages containing the flow code to both of them.
- All devices activate a re-sending timer and after sending or forwarding the *Flow table decision* message, the received *Acknowledge* message with flow code abolishes the timer.
- *The Flow table decision* message is created at the *Inform Decision* model of the controller and finally received and dismantled at the *Operations* model of the router (created at the application layer of a device and received at the application layer of another device).
- The controller uses the *Flow table decision* message to remove the existing flow table by setting the *Establish/Remove* field to *False*.

The structure of the *Acknowledge* message is as shown in Fig. 4-7.

Flow Source	**Flow Destination**	**Message Sender Address**	**Message Receiver Address**	**code**

Fig. 4-7 Acknowledge message structure

4.10.2 Diverting the Flow of High Data Rate

FDA activates this procedure to deal with flow passing via the OSPF path and suffers from congestion due to rise in its data rate.

- Define a network with specific topology. The network consists of number of nodes (routers) and number of links.
- Define ***A*** and ***B*** as two nodes that belong to the network.
- Define a flow of specific data rate commencing from node ***A*** and terminating at node ***B*** passing through some other nodes and links on its way from ***A*** to ***B***.
- The data rate of flow ***A-B*** suffers from an obvious data rate increase.
- FDT checks the availability of carrying on forwarding the flow via the currently employed path but finds this causes congestion at one link or more.
- FDT activates FDA. FDA identifies the congested links.
- FDA checks if there are any previously diverted flows and those pass through the congested links. If there are any, FDA tries to return the previously diverted flow to its original OSPF path apart from the congested links as to be described later FDA at the placement of: *Re-Diverting the Previously Diverted Flow* of **section 4.10.4**. Else, FDA continues with this procedure.
- FDA allocates all possible paths between ***A*** and ***B***.
- This placement represents diverting the flow via a bypass path that is no longer than the OSPF path. FDA chooses the paths that are no longer than the currently used OSPF path.
- Commencing from the shortest path, FDA examines the distribution of flow ***A-B*** via each path. This examination tests the residual bandwidth of each link across the currently tested path regarding the new data rate of flow ***A-B*** as well as the data rate of the other flows passing through the link.
- If the residual bandwidth of all links over the path is greater than zero, this means that the path can handle the flow without any congestion. FDA adopts this path as the main path to forward the flow from ***A*** to ***B***. The controller records the flow's information represented by the chosen path and allocated bandwidth over the path in its database. It marks the flow from ***A*** to ***B*** as diverted flow. FDA terminates the operation.
- The path that is adopted to forward the flow between ***A*** and ***B*** is one of the non OSPF paths. The controller activates the *Flow Table Creation Procedure* of **section 4.10.1** by sending a *Flow table decision message/Establish flow table* to

the routers across the chosen path to create flow tables inside them. FDA terminates the operation.

- When the data rate of the flow reduces to a value that the OSPF path can handle without any congestion, the controller informs the head-end router and the involved routers to return the diverted flow to be forwarded via its original OSPF path and to remove the flow tables from the routers of the bypass path as explained before in FDT of **section 4.8.2**.
- If there is no available path that is no longer than the OSPF path or the residual bandwidth of such path cannot handle flow's data rate then, FDA continues to the next placement test: *Diverting the Non High Data Rate Flow* that explained in **section 4.10.3**.

Example**,** suppose network of Fig. 4-4 where host 1 (H1) increases the data rate it sends to host 2 (H2) at simulation time **T1**. Router 1, the head-end router of flow H1-H2, informs the controller about the data rate increase of flow (H1–H2) by sending a *Flow status updating* message. The controller receives the *Flow status updating* message. It extracts the stored distribution path of flow H1-H2 from its data base. Flow H1-H2 is now forwarded via the OSPF path passing through routers 1-3-4-2. FDT removes the flow's old data rate from the OSPF path then calculates the congestion status by adding the current data rate to the path. FDT finds that the distribution of flow H1-H2 via the OSPF path with its new data rate causes congestion at link 3-4 as shown in Fig. 4-8.
FDT activates FDA, which examines flow H1-H2 distribution via the non OSPF paths. FDA finds several paths that can handle flow H1-H2 without congestion:

- 1-5-6-2
- 1-5-6-7-4-2
- 1-8-9-5-6-2
- 1-8-9-5-6-7-4-2

FDA adopts the path that is no longer than the current OSPF path. The only path that can handle flow H1-H2 without congestion is the path that is passing through routers 1-5-6-2 as it traverses 4 routers, which makes it no longer than the OSPF path 1-3-4-2 of 4 routers either. The controller randomly chooses a specific flow *code* that represents flow H1-H2. It constructs a *Control* message of type the *Flow table decision* message of creating a flow table, inserts the new path and the flow *code* inside it and sends it to router 1 (the head-end router of flow H1-H2).

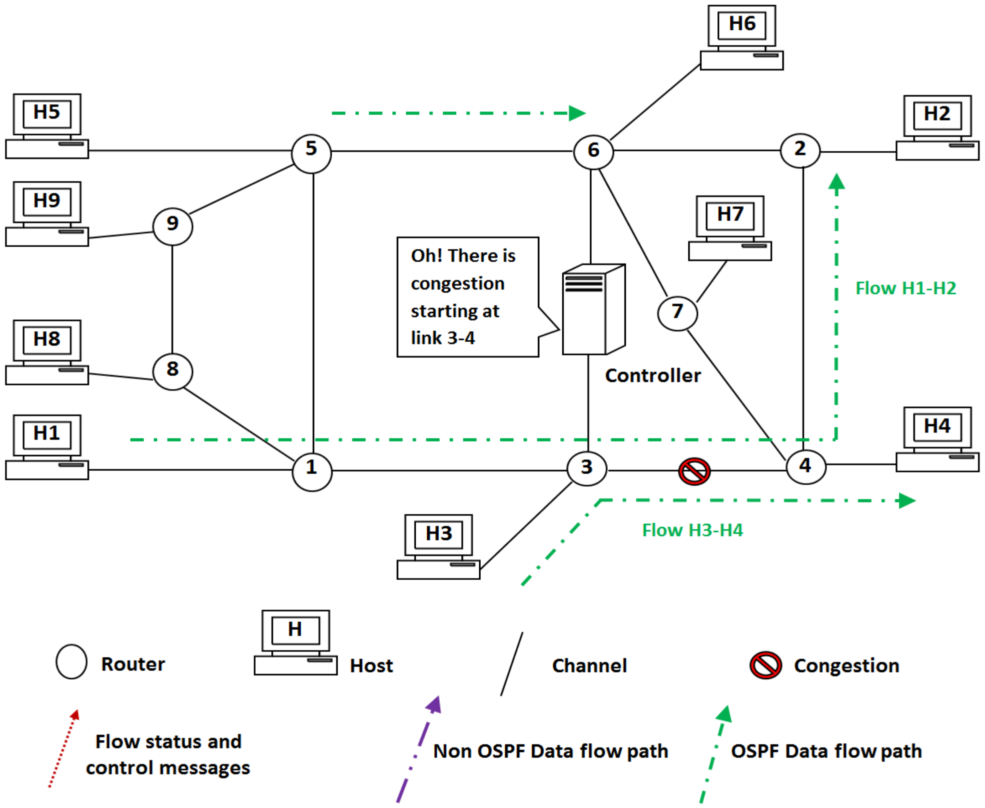


Fig. 4-8 Congestion begins at link 3-4

Router 1 receives the *Flow table decision* message, dismantles it and compares the next hop according to its routing table related to flow H1-H2 (router 3) with the next hop according to the path suggested by the controller (router 5). Router 1 finds that the decisions are different. Therefore, it constructs the flow table related to flow H1-H2.

Router 1 forwards *Flow table decision* message to router 5 as it is the next hop of the flow path. When receiving the *Flow table decision* message, router 5 immediately sends an *Acknowledge* message to router 1 that contains the flow H1-H2 *code* to prevent it from re-sending the *Flow table decision* message and at the same time, it repeats the comparing operation of the next hop regarding destination H2 between the controller's decision (router 6) and its existing routing table (router 6). Router 5 finds that the controller decision related to flow H1-H2 is compatible with its routing table. Therefore, there is no need to create a flow table.

Router 5 forwards the *Flow table decision message* to router 6, which repeats the same comparing and forwarding operations. The final router that receives the *Flow table decision message* is router 2 (the tail-end router of flow H1-H2). Router 2 sends two *Acknowledge* messages; one to router 6 and the other to the controller. The *Acknowledge* message contains flow H1-H2 *code* to confirm the successful construction of the new path that serves flow H1-H2. The related routers of the network commence forwarding the forthcoming *Data* packets of flow H1-H2 via path 1-5-6-2 and as shown in Fig. 4-9 avoiding the congestion at link 3-4.

The results that measure the QoS under same applied traffic show an improvement in performance of the network with a controller compared to that of without controller. The results of Packets Delay Variation (PDV) recorded at H2 show that the received packets of flow H1-H2 in simulation of controller of Fig. 4-10 are more organized and better harmonized than those of no controller simulation of Fig. 4-11.

H2 records PDV of the data packets of flow H1-H2 that successfully arrive to the destination. There is no PDV recorded for the packets that drop along the path.

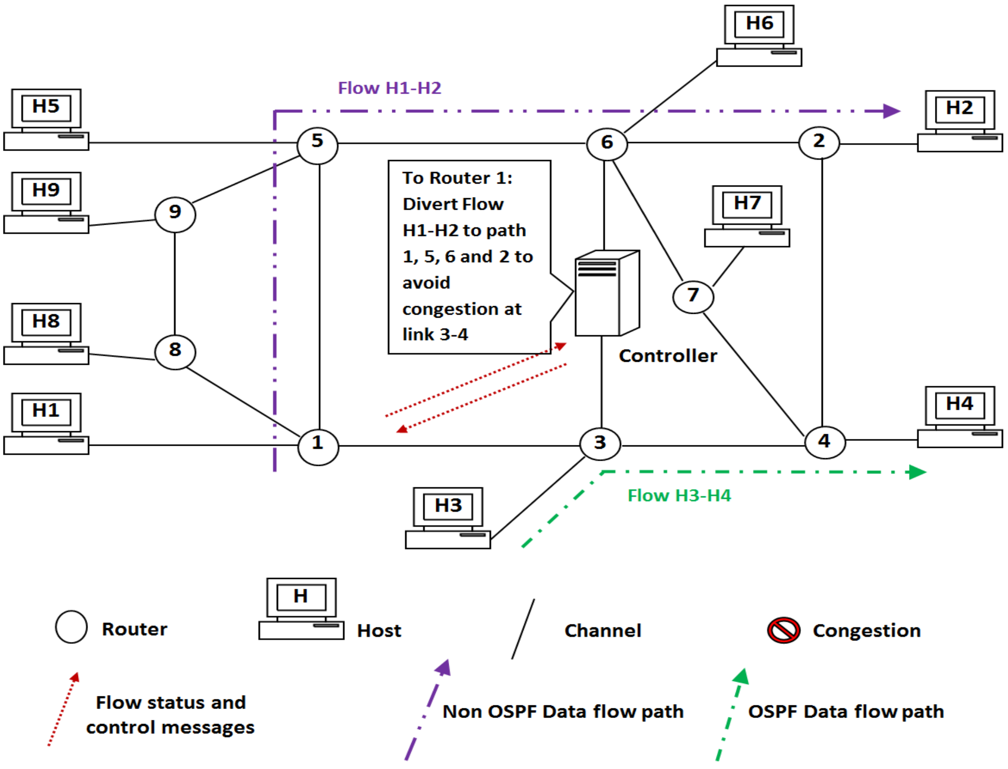


Fig. 4-9 Distribution of flow H1-H2 after diversion

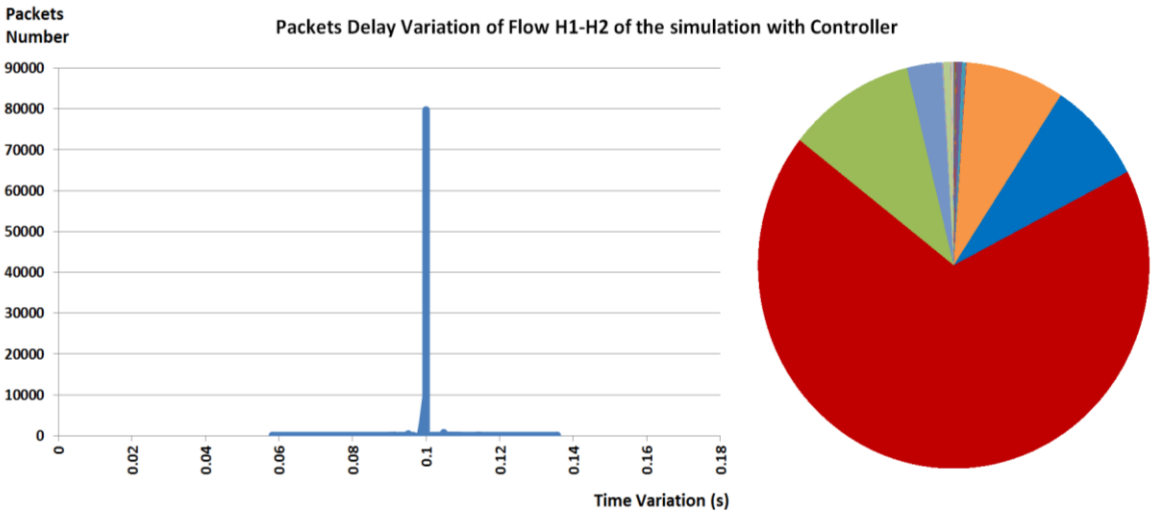


Fig. 4-10 Packets delay variation of flow H1-H2 of the simulation with controller

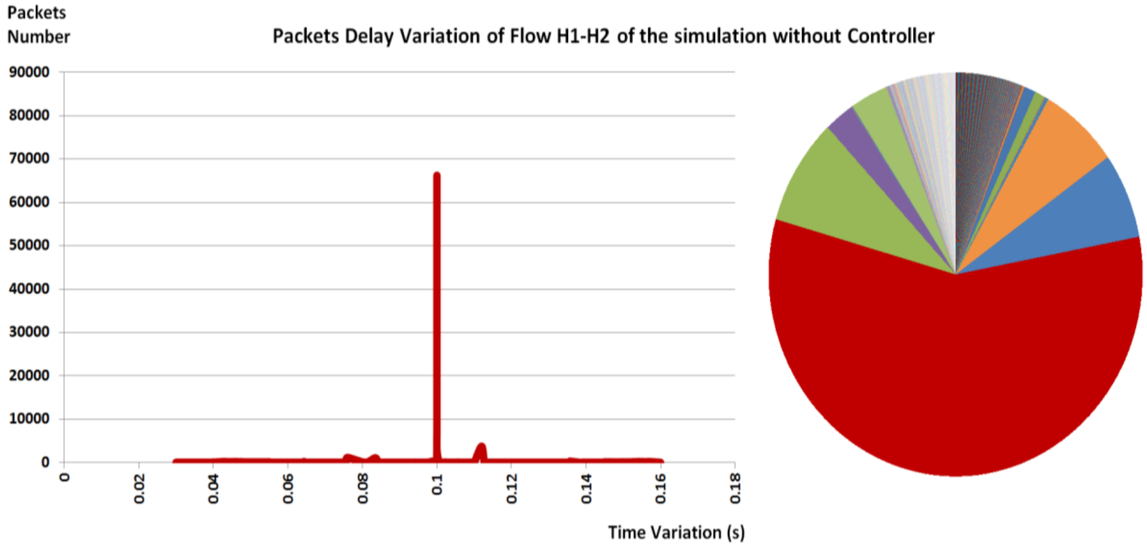


Fig. 4-11 Packets delay variation of flow H1-H2 of the simulation without controller

The latency plot of flow H1-H2 is as shown in Fig 4-12. Latency shows that the diverted packets of flow H1-H2 consume less time in its travelling journey from source H1 to destination H2 than those passing through the OSPF path (no controller) under congestion status. It can be noticed that the diverted flow has less propagation delay and more stability.

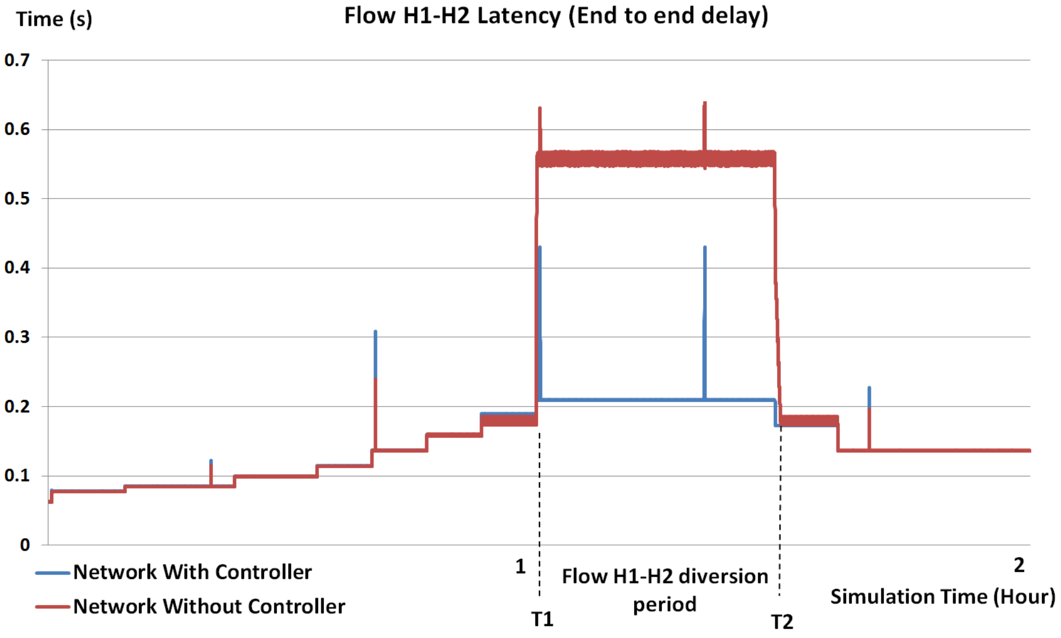


Fig. 4-12 Flow H1-H2 latency recorded for both simulations of with and without controller

At simulation time **T2**, host 1 (H1) reduces the data rate it sends to host 2 (H2). Router 1 informs the controller via *Flow status updating message*.

After receiving the *Flow status updating message*, the controller extracts the stored distribution path of flow H1-H2 from its database. Flow H1-H2 was previously forwarded via the non OSPF path passing through routers 1-5-6-2. FDT removes the flow’s previous data rate from the path then calculates the congestion status by adding the current received data rate over the OSPF path. The controller notices that the network can handle forwarding flow H1-H2 via the OSPF path 1-3-4-2 without any congestion.

According to this information, the controller constructs a *Flow table decision* message of active *Removing a flow table* bit. It inserts the previously established alternative flow path (1-5-6-2) and *code* into it. It sends the message to the head-end router, router 1, which removes the flow table of flow H1-H2. Router 1 forwards *Flow table decision* message*/Remove flow table* to router 5. Router 5 does not have a flow table of flow H1-H2. Therefore it does nothing except forwarding *Flow table decision* message*/Remove flow table* to the next hop (router 6). The operation is repeated until reaching the tail end router (router 2), which contacts the controller with an *Acknowledge* message containing flow *code* and confirming the removal operation of flow table of flow H1-H2.

The average time of all *Data* packets that travel from all sources to all destinations (total network delay) is recorded by the network listener model and the result is shown in Fig. 4-13.

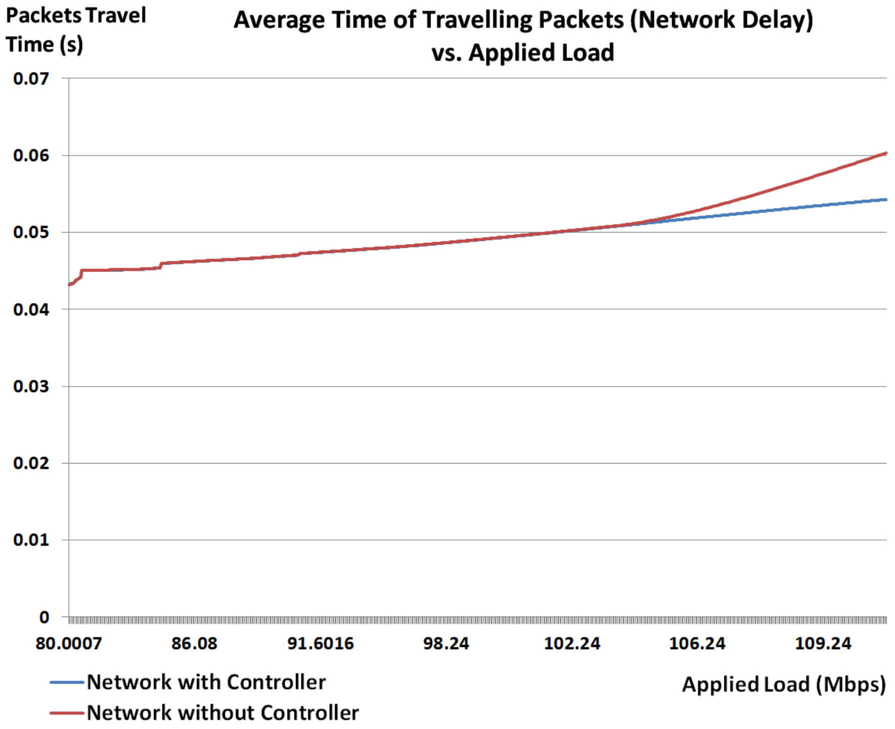


Fig. 4-13 Network delay (with and without controller)

The average time of all *Data* packets of network with controller is less than that without controller for higher applied load. The number of dropped packets is less in the simulation with controller, which results in a higher throughput recorded as well.

4.10.3 Diverting the Non High Data Rate Flow

The data rate of the flows is changeable during the simulation time. If the residual bandwidth of an OSPF path cannot occupy the current flow data rate, the controller diverts the flow via another bypass path. There are several conditions to choose the new path. First condition is the length of the path. It is not preferable to divert any flow over path which is longer than its OSPF path. Therefore, before taking the flow diversion decision, the controller applies testing operation to choose which flow it should divert. The controller applies this placement if the previous placement (*Diverting the Flow of High Data Rate*) could not solve the congestion problem within network constraints.

- For the network that the previous placement could not solve its congestion problem, FDA continues forwarding the flow from ***A*** to ***B*** via its OSPF path. It adds its data rate to the path in the controller's data rate.
- FDA tries to divert another flow to remove the congestion from the path. It identifies the congested links.
- FDA distinguishes the flows that are repeated across the congested links.
- Define flow ***C-D*** as a flow begins at node ***C***, terminates at node ***D*** and passes through the congested links. Both ***C*** and ***D*** are part of the network.
- FDA identifies the path of flow ***C-D***. Then, it subtracts the flow's data rate from the path. If the subtracted data rate removes the congestion, FDA continues finding alternative path to flow ***C-D***. If not, FDA choses another flow that to be diverted.
- Suppose extracting the data rate of flow ***C-D*** removes the congestion. FDA identifies all possible paths between ***C*** and ***D***. FDA chooses the paths that are no longer than the currently used OSPF path.
- Commencing from the shortest path, FDA examines the distribution of flow ***C-D*** via each path. This examination tests the residual bandwidth of each link across the currently tested path.
- If the residual bandwidth of all links over the path is greater than zero, this means that the path can handle the flow without any congestion. FDA adopts this path as the main path to forward flow ***C-D***. The controller records the flow's information represented by the chosen path and allocated bandwidth over the path in its database. It marks the flow from ***C*** to ***D*** as diverted flow. FDA terminates the operation.
- If there is no available alternative path to divert flow ***C-D***, FDA applies the

previous steps on another flow and so on.

- If all flows have been tested without solving the congestion problem, then FDA continues to the next step: *Re-Diverting the Previously Diverted Flow* of **section 4.10.4.**

Diverting any flow over non OSPF path that is longer than the OSPF path is not preferable because this increases flow travelling time which increases network delay and the flow that travels over many nodes may share links over which it traverses with more other flows than if it forwarded over less number of nodes; this increases the network bandwidth occupation and the congestion probability.

Example**,** suppose network of Fig. 4-4 where host 3 (H3) increases the data rate it sends to host 4 (H4) at simulation time T1. Router 3, the head-end router of flow H3-H4, informs the controller about the upserge in data rate of flow (H3–H4) by sending a *Flow status updating* message. The controller receives the *Flow status updating* message. It extracts from its data base the stored distribution path of flow H3-H4. Flow H3-H4 was previously forwarded via the OSPF path passing through routers 3-4.

FDT internally removes the flow’s old data rate from the path then calculates the congestion status by adding the current data rate to the OSPF path. FDT finds that the distribution of flow H3-H4 with its new data rate via the OSPF path causes congestion at link 3-4. Therefore, it activates FDA which during its procedure to solve the congestion problem finds several paths that can handle flow H3-H4 without congestion:

- 3-1-5-6-2-4
- 3-1-5-6-7-4
- 3-1-8-9-5-6-2-4
- 3-1-8-9-5-6-7-4

FDA adopts the path that is no longer than the current OSPF path (3-4) but there is no existing path that fulfils this requirement. FDA adds the flow data rate to its distribution map (via the OSPF path). Then it allocates all the flows that are passing through link 3-4. In addition to flow H3-H4, the flows passing through link 3-4 are:

- Flow H1-H4 – passes through the: OSPF path 1-3-4.
- Flow H1-H2 – passes through the: OSPF path 1-3-4-2.

FDA applies the hops number and the bandwidth conditions when it tests diverting those flows via alternative (non OSPF) paths. It finds that the only flow that can be diverted through a path that is no longer than its OSPF path and removes the congestion from link 3-4 is flow H1-H2. The controller informs the involved routers to divert flow H1-H2. Flow H1-H2 is now forwarded via path 1-5-6-2 to avoid the congestion at link

3-4 and as was shown in Fig. 4-9.

The results of Packets Delay Variation (PDV) recorded at H4 show that the received packets of flow H3-H4 in simulation of controller of Fig. 4-14 are more organized and better harmonic than those of no controller simulation of Fig. 4-15. The latency plot of flows H3-H4 and H1-H2 are as shown in Figs 4-16 and 4-17 respectively. Latency shows that the diverted packets of flow H1-H2 consume less time in its travelling journey from source to destination than those continue passing through the OSPF path (no controller). The *Data* packets of flow H3-H4 also consume less time in its travelling journey from source to destination in the network of controller than those of the network without controller. Diverting flow H1-H2 away from the congested link 3-4 provided better fluency to all flows passing through link 3-4. For Figs 4-16 and 4-17 and at simulation time **T2**, host 3 (H3) reduces the data rate it sends to host 4 (H4). Router 3 informs the controller via *Flow status updating message*. The controller later (At simulation time **T3)** returns flow H1-H2 to be diverted via the OSPF path 1-3-4-2.

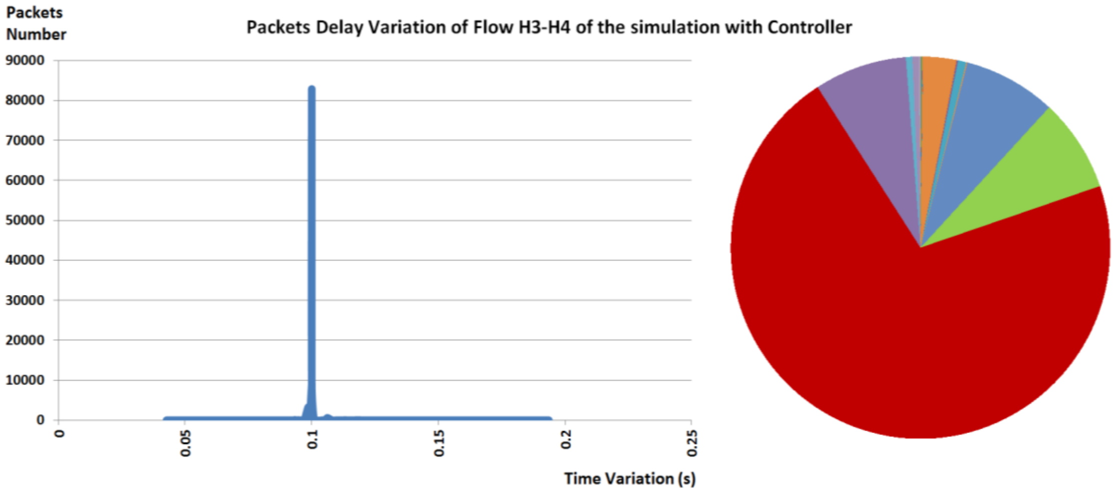


Fig. 4-14 Packets delay variation of flow H3-H4 of the simulation with controller

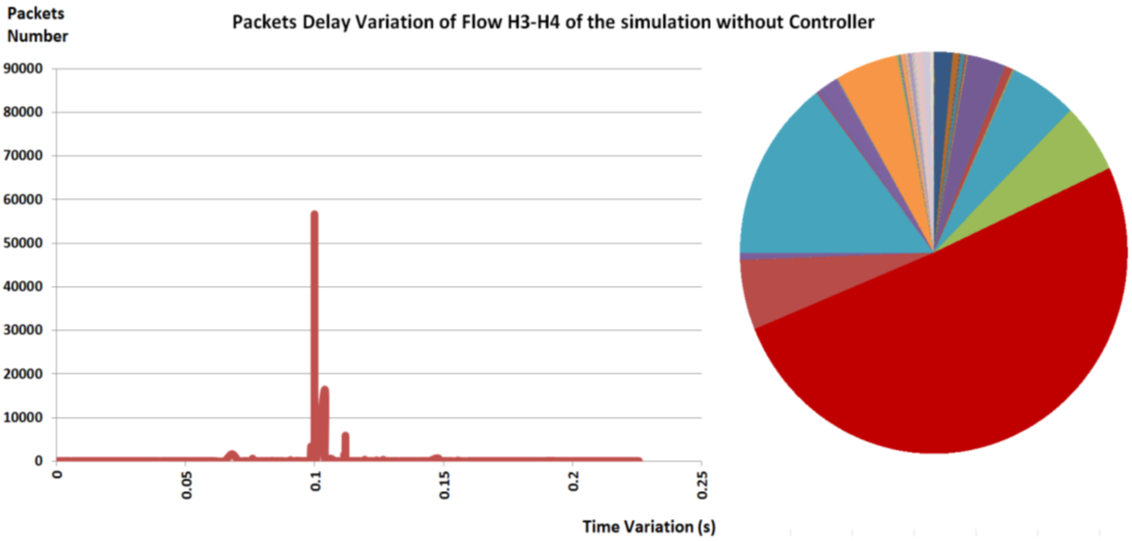


Fig. 4-15 Packets delay variation of flow H3-H4 of the simulation without controller

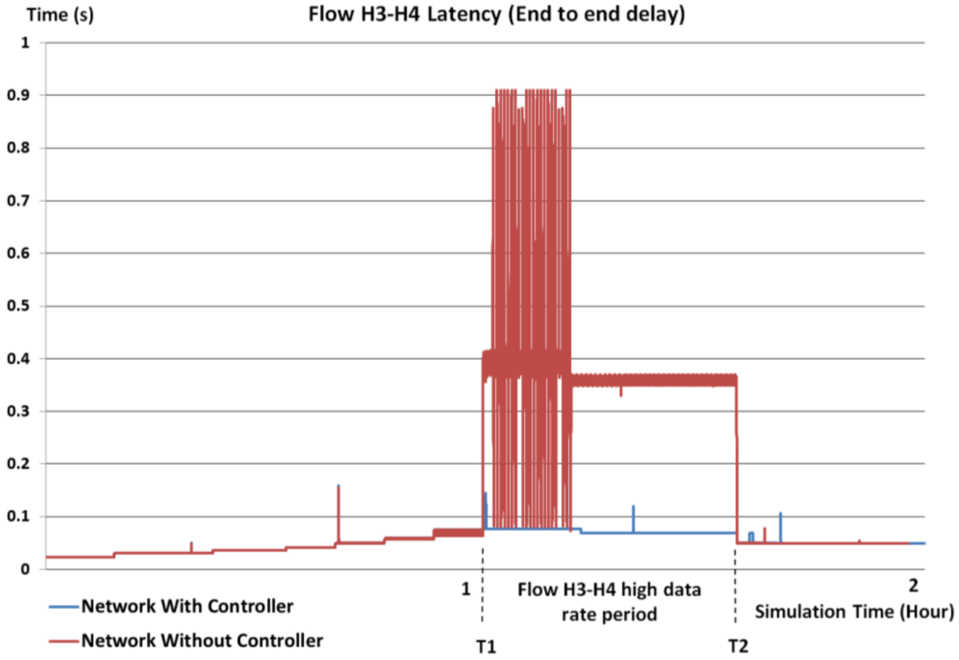


Fig. 4-16 Flow H3-H4 latency recorded for both simulations (with and without controller)

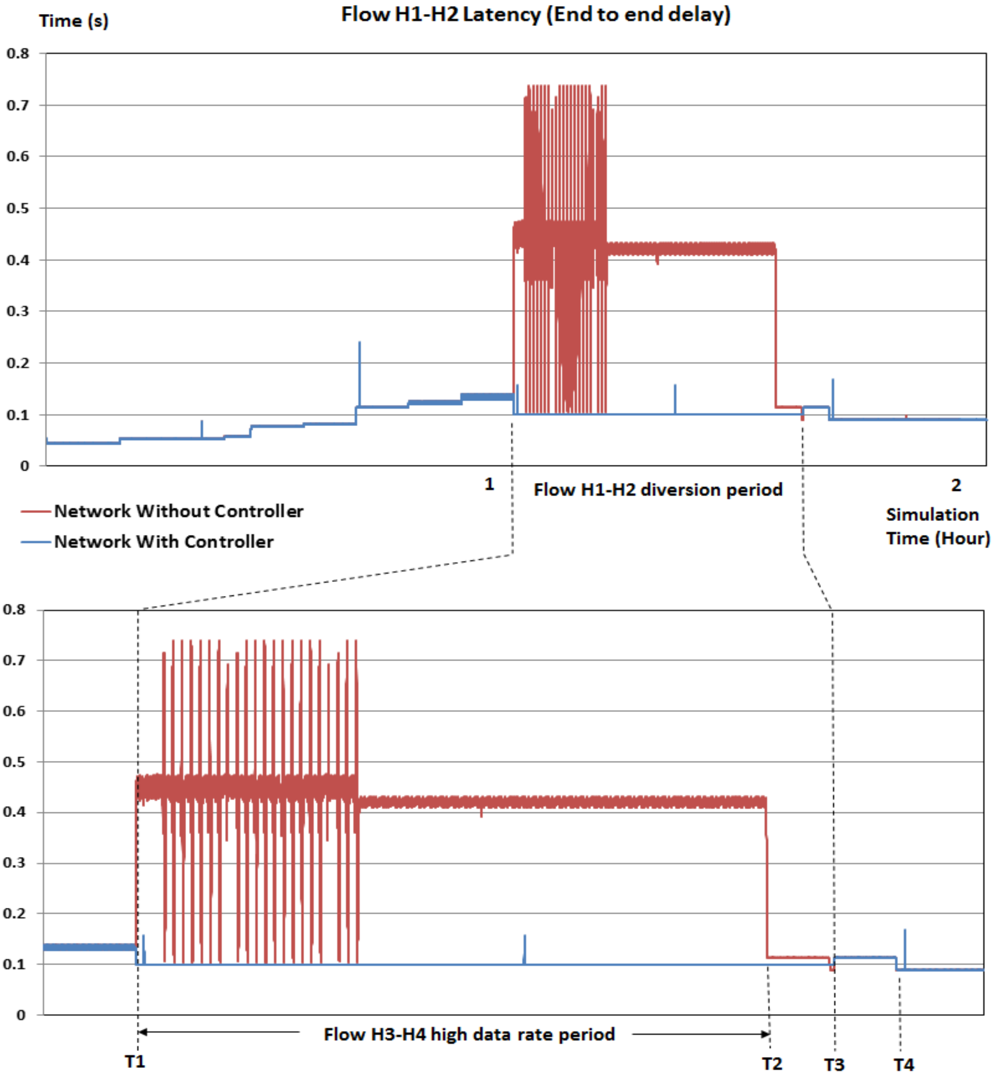


Fig. 4-17 Flow H1-H2 latency recorded for both simulations (with and without controller)

The total network delay is illustrated in Fig. 4-18 below where the network with controller transports *Data* packets with less delay under higher applied traffic.

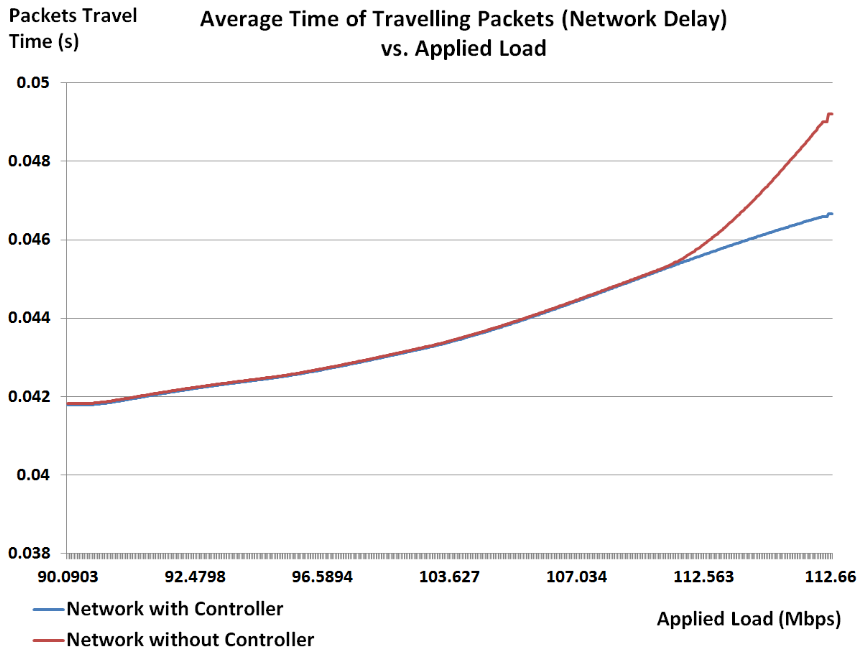


Fig. 4-18 Network delay (with and without controller)

4.10.4 Re-Diverting the Previously Diverted Flow

Sometimes, diverting many flows from their OSPF paths to other paths causes conflict in the flows distribution and consumes a lot of mathematical operations inside the controller which leads to occupy more links with more of the available bandwidth. Therefore, it is better to continue forwarding flows over the OSPF paths by avoiding the diversion of more flows over other paths as much as possible. This placement discusses the possibility of re-diverting the previously diverted flows to avoid the congestion.

- Define a network suffers from congestion problem somewhere which involved FDT to activate FDA. When FDA identifies the congested links, it finds previously diverted flows and those flows pass through the congested links.
- FDA checks the possibility of returning the previously diverted flows to their original OSPF paths as described in the procedure: *Returning the diverted flow to its original OSPF path* of **sub-section 4.10.4.i**.
- If this procedure solves the problem, FDA terminates the operation.
- Else, FDA applies the procedure: *Re-diverting the previously diverted flow via a longer path* that avoids all the congested links described in **sub-section 4.10.4.ii**. This procedure is also applied if none of the previous placement: *Diverting the*

Non High Data Rate Flow of **section 4.10.3** could not solve the congestion problem.

- If this procedure solves the problem, FDA terminates the operation. Else, FDA continues to the last placement: *Multiple Flows Diverting* of **section 4.10.5**.

i. Returning the diverted flow to its original OSPF path

The normal flows' distribution goes via their OSPF paths. The controller is provided with a timer that works every several minutes. When the timer reaches to zero, the controller checks whether the diverted flows can be forwarded again via their normal OSPF paths. Sometimes, the controller needs to activate this placement to check returning of specific flow to its original OSPF path before the timer expires. This may prevent congestion on the bypass path.

- FDA checks the possibility of returning the previously diverted flows to be forwarded via their original OSPF paths.
- Define a network with specific topology. The network consists of number of nodes (routers) and number of links.
- Define a flow of specific data rate commencing from node ***A*** and terminating at node ***B*** passing through some other nodes and links on its way from ***A*** to ***B***. The data rate of flow ***A-B*** suffers from an obvious data rate increase.
- FDT checks the availability of carrying on forwarding the flow via the currently employed path but finds this causes congestion at one link or more.
- FDT activates FDA. FDA identifies the congested links and the flows currently pass through them. It finds that one of the flows (flow ***C-D***) has been previously diverted and now it passes through the congested links.
- FDA aims to return flow ***C-D*** to be forwarded through its OSPF path and if possible, it activates the *Flow Table Removal Procedure* of **section 4.10.1** to return it to its original path. FDA terminates the operation.
- If FDA cannot return any of the previously diverted flows to their original OSPF paths without congestion, FDA continues to apply the second step of this placement: ***Re-diverting*** *the previously diverted flow* ***via longer none OSPF path*** *that avoids the congested links* as illustrated in the following **sub-section 4.10.4.ii.**

ii. Re-diverting via longer none OSPF path

If the OSPF path cannot handle the previously diverted flow, then the controller should re-divert it to avoid congestion on its currently used bypass path. During the re-diverting operation, the controller allocates the other bypass paths from the shortest to the longest avoiding the OSPF path and the congested link/ links of the currently used path. This placement is applied at *Flow Re-Divert* model of the controller.

- FDA checks the possibility of re-diverting the previously diverted flows to be forwarded via longer non OSPF paths.
- Those alternative non OSPF paths avoid the congested links.
- If applying this step removes the congestion, FDA terminates the operation.
- If this step cannot solve the congestion problem, FDA applies its the last placement: *Multiple Flows Diverting* **section 4.10.5** to solve the congestion problem.

Example, suppose network of Fig. 4-19 where flow H1-H2 has been previously diverted (at simulation time **T1**) to a bypass path passes through routers 1-5-6-2 to avoid congestion that previously occurred at link 3-4. At simulation time **T2**, host 6 (H6) increases the data rate it sends to host 2 (H2). Router 6, the head-end router of flow H6-H2, informs the controller about the data rate increase of flow (H6–H2) by using *Flow status updating* message.

The controller receives the *Flow status updating* message. It extracts from its data base the stored distribution path of flow H6-H2. Flow H6-H2 is currently forwarded via the OSPF path passing through routers 6-2. FDT removes the flow's old data rate from the path then calculates the congestion status by adding the current data rate to the OSPF path. FDT finds the distribution of flow H6-H2 with its new data rate that causes congestion at link 6-2. FDA investigates all the flows passing through the congested link 6-2 which are:

- Flow H1-H2 – passes through the: previously diverted detour path 1-5-6-2.
- Flow H5-H2 – passes through the: OSPF path 5-6-2.
- Flow H6-H2 – passes through the: OSPF path 6-2.

FDA finds that flow H1-H2 was previously diverted at simulation time **T1** to avoid congestion at link 3-4. FDA adds the new data rate of flow H6-H2 to the OSPF path (which passes through link 6-2) and removes the data rate of flow H1-H2 from the bypass path (which also passes through link 6-2) and notices the congestion state.

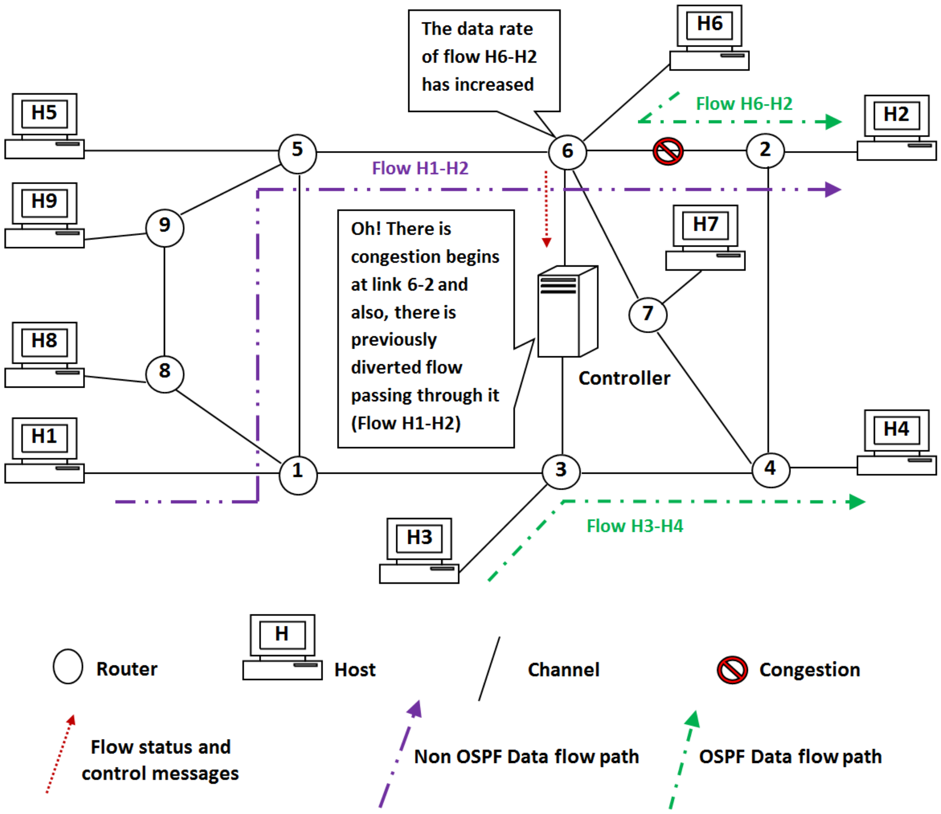

The data rate of flow H6-H2 has increased
H6
H5
H2
Flow H6-H2
5
6
2
Flow H1-H2
H9
9
Oh! There is congestion begins at link 6-2 and also, there is previously diverted flow passing through it (Flow H1-H2)
H7
7
H8
8
Controller
H1
H4
1
3
4
H3
Flow H3-H4
Router
H
Host
Channel
Congestion
Flow status and control messages
Non OSPF Data flow path
OSPF Data flow path

Fig. 4-19 Allocating the previously diverted flow passing through currently congested link

In our example, removing flow H1-H2 from link 6-2 is enough to remove the link congestion. The first step of FDA is testing the distribution of flow H1-H2 via its OSPF path via routers 1-3-4-2. If returning the flow to its OSPF path does not cause any congestion problem at another place over the network, the controller establishes a *Flow table decision message/Remove flow table* to remove the flow tables related to flow H1-H2 from the routers where they exist. The network commences forwarding flow H1-H2 via its OSPF path through routers 1-3-4-2 and as was shown before in Fig. 4-4.

In this example, the applied flows and their distribution do not allow flow H1-H2 to return passing through its OSPF path (there will be congestion at link 3-4). FDA investigates if there are any other bypass paths that can handle flow H1-H2 without passing through the congested links 6-2 and 3-4. It finds the following paths:

- 1-5-6-7-4-2.
- 1-8-9-5-6-7-4-2.

It adopts the path 1-5-6-7-4-2 to forward flow H1-H2 and to avoid the congestion at links 6-2 and 3-4. The controller sends *Flow table decision* message/*Remove flow table* to remove flow table path 1-5-6-2 followed by *Flow table decision* message/*Establish flow table* to create flow table path 1-5-6-7-4-2. The network forwards flow H6-H2 via OSPF path 6-2 and flow H1-H2 via detour path 1-5-6-7-4-2 as shown in Fig. 4-20.

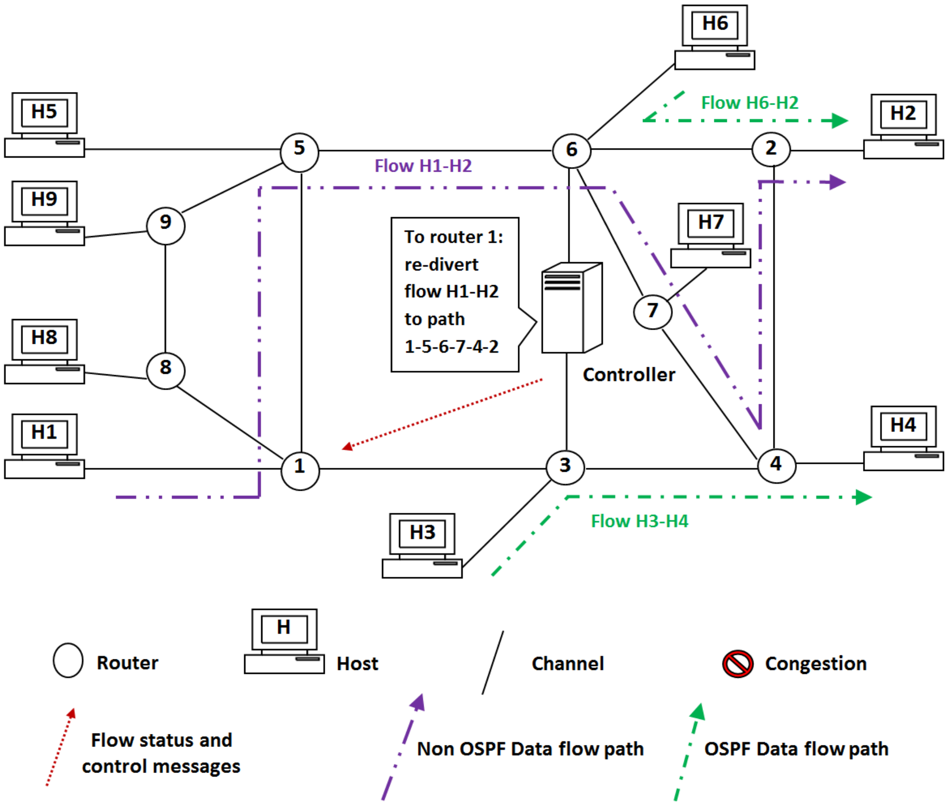


Fig. 4-20 Re-diverting the previously diverted flow H1-H2

The latency of flows H1-H2, H3-H4 and H6-H2 are shown in Figs 4-21, 4-22 and 4-23 respectively.

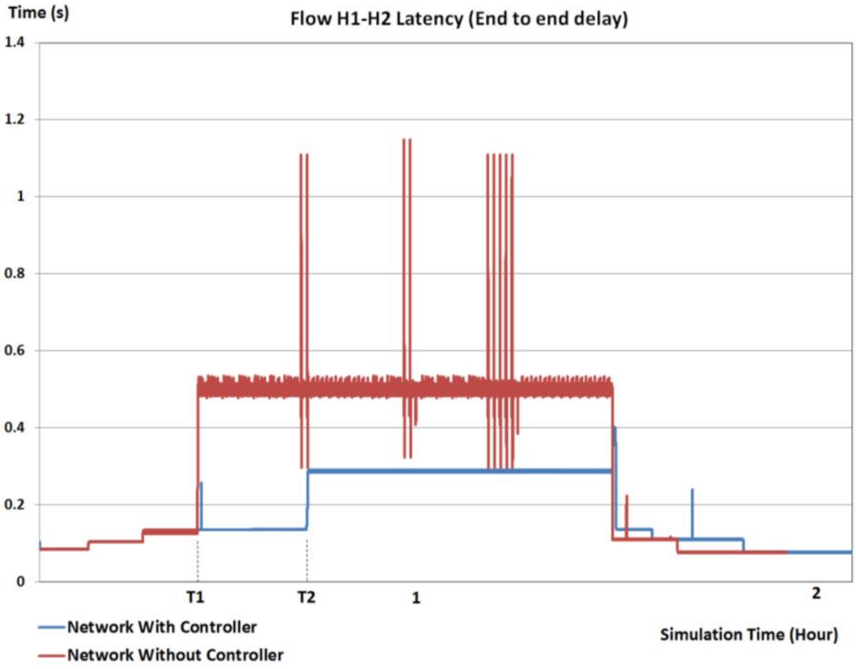


Fig. 4-21 Flow H1-H2 latency (with and without controller)

It can be noticed that the controller has diverted H1-H2 twice. The first diversion was to avoid congestion at link 3-4 and the second diversion (re-diversion) was to avoid congestion at link 6-2.

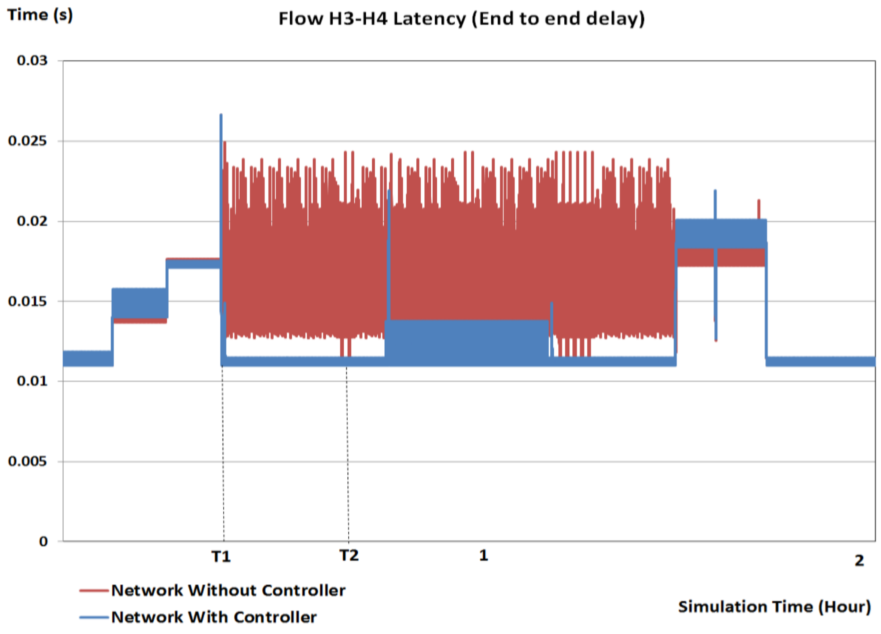


Fig. 4-22 Flow H3-H4 latency (with and without controller)

Fig. 4-23 shows that the latency of flow H6-H2 of the network without controller has been badly affected by the congestion at link 3-4 even though, flow H6-H2 passes through path 6-2 apart from the congested link 3-4. The reason behind this influence is the *route flapping* destructive effect.

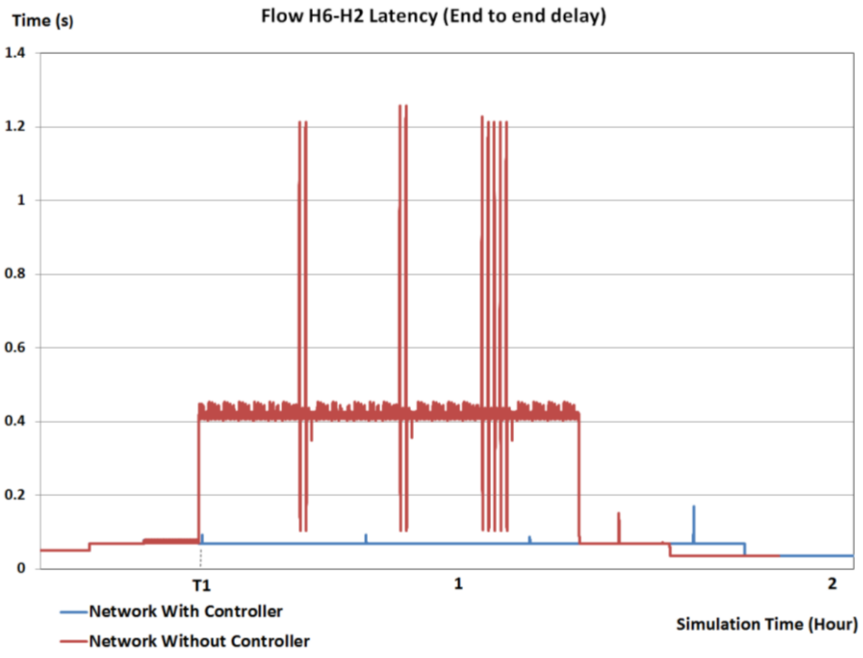


Fig. 4-23 Flow H6-H2 latency (with and without controller)

Re-diversion of flow H1-H2 provides better bandwidth usage and prevents *route flapping* effect that looks obvious on the distribution of flow H6-H2 in no controller simulation of Fig. 4-23. The improvement in total network delay is shown in Fig 4-24. The network with controller transports *Data* packets with less delay under higher applied traffic.

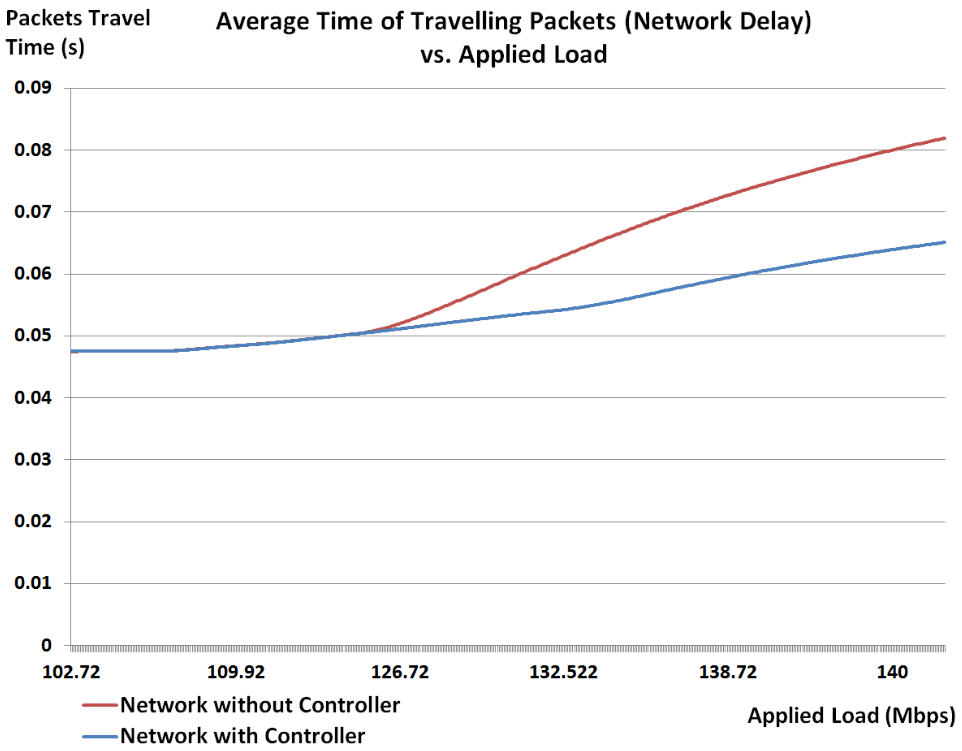


Fig. 4-24 Network delay (with and without controller)

4.10.5 Multiple Flows Diverting

The controller always attempts to reach with the network to the stable state. Sometimes, diverting or re-diverting of one flow cannot lead to a stable state. Therefore, the controller may divert more than one flow at more than one place until reaching to the stable state. The controller applies this placement at its *Flow Exchange Algorithm* model.

- For the network that suffers from congestion problem because the increase of the data rate of flow ***A***-***B*** and none of the previous placements could not solve its congestion problem, FDA continues forwarding flow ***A***-***B*** via its OSPF path. It adds its data rate to the path in the controller's data rate.
- FDA tries to divert another flow to remove the congestion from the path. It identifies the congested links. FDA distinguishes the flows that are repeated across the congested links. It finds flow ***C-D*** as one of the flows passing through the congested links. Both ***C*** and ***D*** are nodes belong to the network.
- FDA identifies the path of flow ***C-D***. Then, it subtracts its data rate from the path. Suppose the subtracted data rate removes the congestion from the currently congested links.
- FDA diverts flow ***C-D*** to an alternative path that is no longer than its currently used OSPF path.
- Diverting flow ***C-D*** to the alternative path removes the congestion from specific place of the network but creates another congestion at alternative place.
- FDA distinguishes the flows that are repeated across the newly identified congested links. It finds flow ***E-F*** as one of the flows passing through the congested links. Both ***E*** and ***F*** are nodes belong to the network. FDA identifies the path of flow ***E-F***. Then, it subtracts the data rate from the path.
- If the subtracted data rate removes the congestion from the currently congested links, FDA continues with the procedure. FDA diverts flow ***E-F*** to an alternative path that is no longer than its currently used OSPF path. Diverting flow ***E-F*** to the alternative path removes the congestion from latter congested links.
- The result is there is no congestion at any link inside the network. FDA terminates the operation. FDA marks flows ***C-D*** and ***E- F*** as diverted flows.

Regarding this placement, it can be said that FDA diverts two flows via bypass paths no longer than their OSPF paths to clear the way to third flow to pass via OSPF path without congestion. In the case where none of the mentioned placements solves the congestion problem, FDA abolishes all the previously performed calculations then tests all the paths that can carry the flow of high data rate starting from the shortest to the longest. When FDA finds one, it diverts the flow via it regardless the length.

Example**,** suppose network of Fig. 4-25 where host 1 (H1) increases the data rate it sends to host 5 (H5) at simulation time **T1**. Router 1, the head-end router of flow H1-H5, informs the controller about the data rate increase of flow (H1–H5) by sending a *Flow status updating* message.

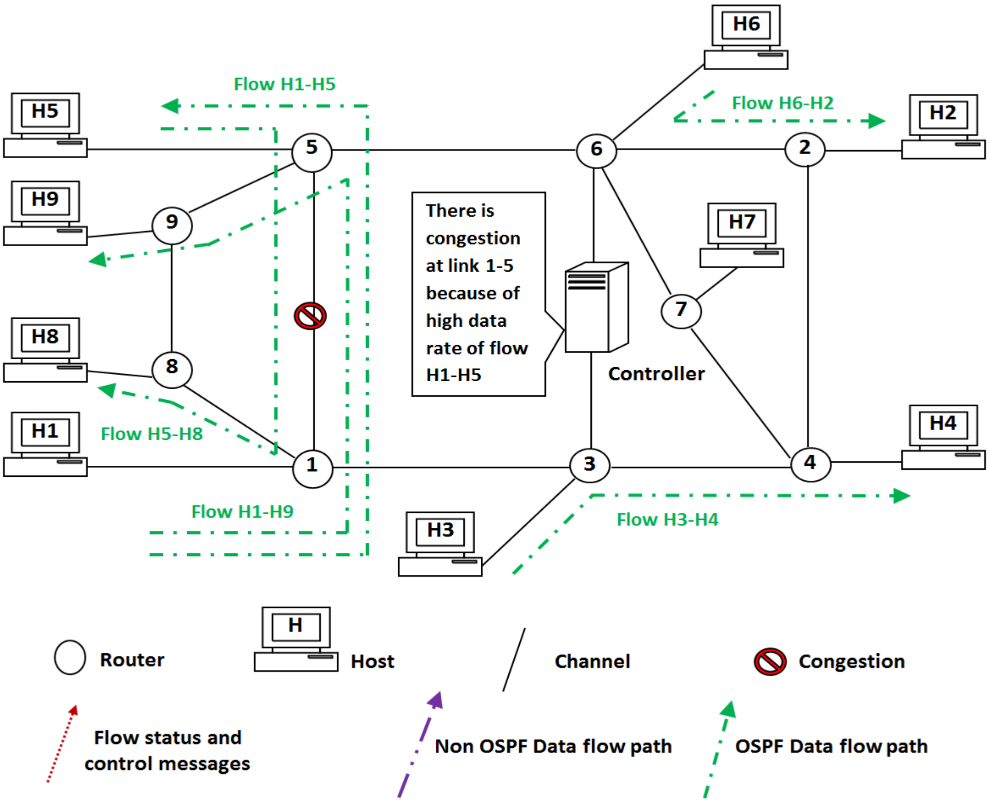


Fig. 4-25 Network with congestion at link 1-5

The controller’s FDT examines the distribution of flow H1-H5 and finds a beginning of congestion at link 1-5. FDT activates FDA. FDA tries to find a solution that removes the congestion. After testing the flows, the only flow that can be forwarded to a bypass path that is no longer than its OSPF path is flow H1-H9. Flow H1-H9 is now passing through OSPF path 1-5-9. It can be diverted to a bypass path 1-8-9 to remove the congestion from link 1-5. However, diverting flow H1-H9 to path 1-8-9 causes congestion at link 1-8 as shown in Fig 4-26. Therefore, the FDA repeats the scenario of solving the congestion problem at link 1-8. It finds that diverting flow H5-H8 from its OSPF path 5-1-8 to a bypass path 5-9-8 will remove the congestion from link 1-8.

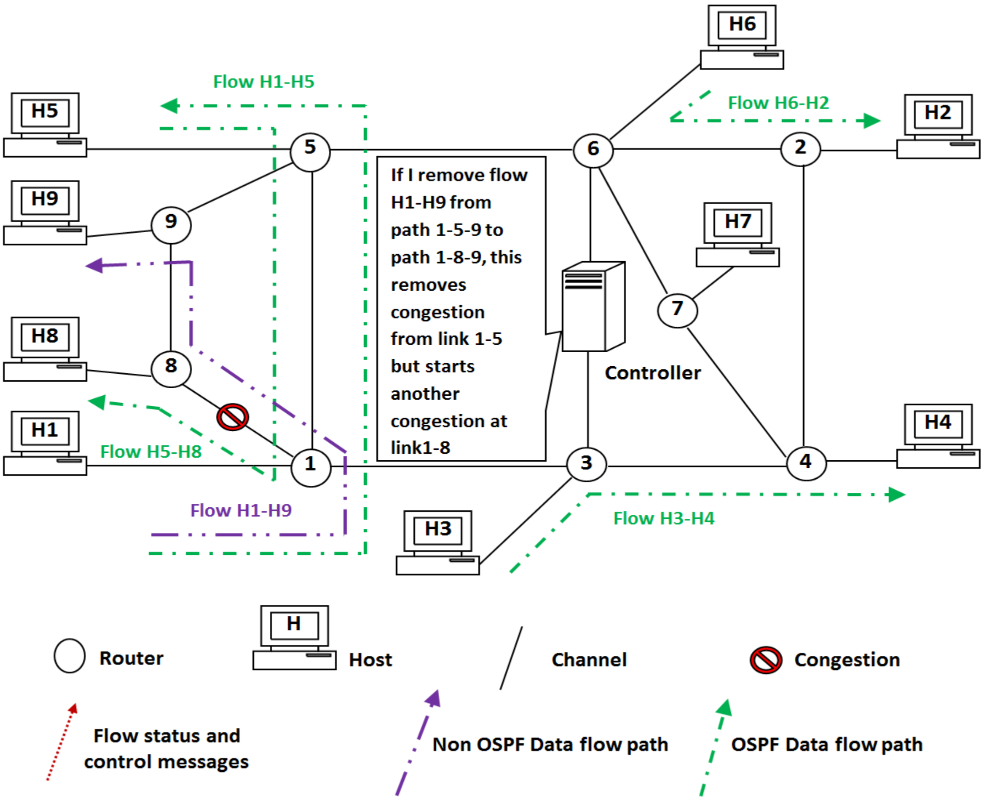

H6
Flow H1-H5
Flow H6-H2
H5
H2
5
6
2
If I remove flow H1-H9 from path 1-5-9 to path 1-8-9, this removes congestion from link 1-5 but starts another congestion at link1-8
H9
9
H7
7
H8
8
Controller
H4
H1
Flow H5-H8
1
3
4
Flow H1-H9
H3
Flow H3-H4
H
Router
Host
Channel
Congestion
Flow status and control messages
Non OSPF Data flow path
OSPF Data flow path

Fig. 4-26 Network with congestion at link 1-8

Thus, it diverts flow H1-H9 to the bypass path 1-8-9 and flow H5-H8 to the bypass path 5-9-8. The new flows distribution without congestion is as shown in Fig 4-27. After flow H1-H5 returns to normal data rate, the controller removes the established flow tables.

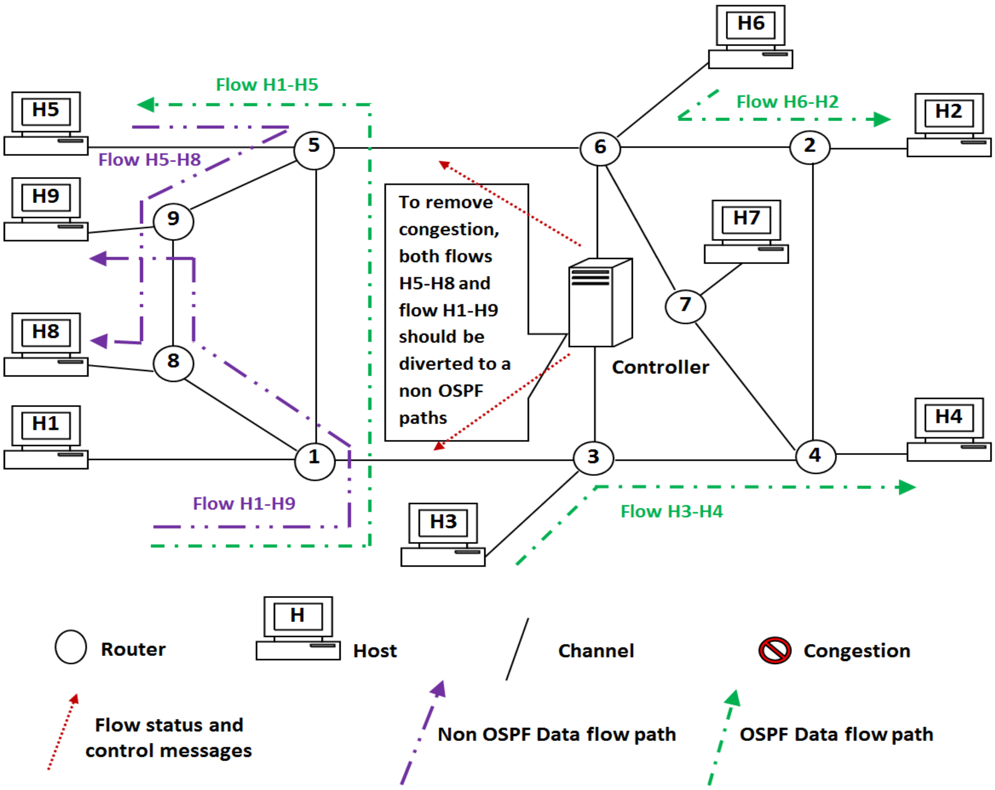


Fig. 4-27 Network without congestion

The network returns flows forwarding to be according to the OSPF routing tables. Fig. 4-28 shows the latency of flow H1-H5 for both simulations with and without controller under the same applied load. Fig. 4-29 shows the packets' delay variation of flow H1-H5 in the simulation where the controller exists and Fig. 4-30 shows the packets delay variation of flow H1-H5 in the simulation where no controller exists. Figs. 4-31 and 4-32 respectively show the latency of flows H1-H9 and H5-H8. The destructive effect of the congestion is obvious in the higher and un-inform latency that flows suffers from. All of Figs. 4-28, 4-31 and 4-32 represent the results of two simulations of with and without controller under the same applied load.

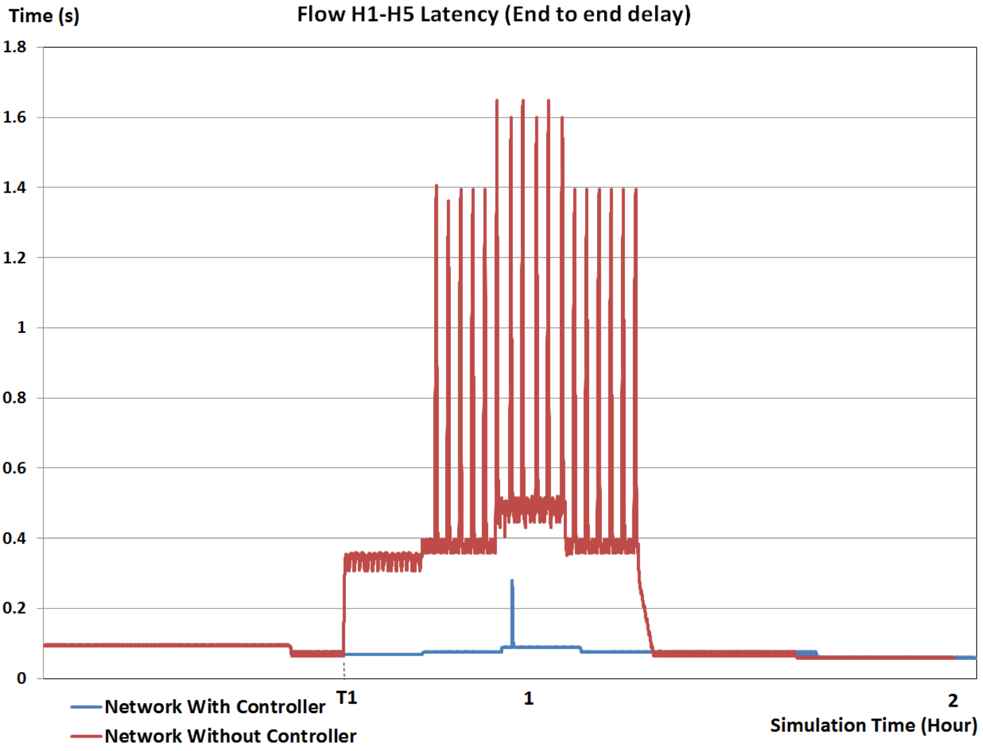


Fig. 4-28 Flow H1-H5 latency (with and without controller)

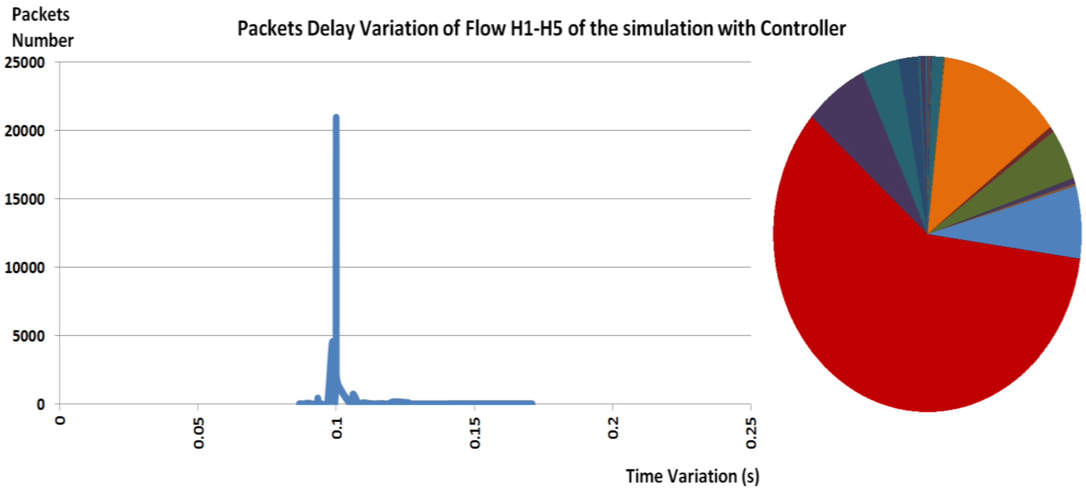


Fig. 4-29 PDV of flow H1-H5 (with controller)

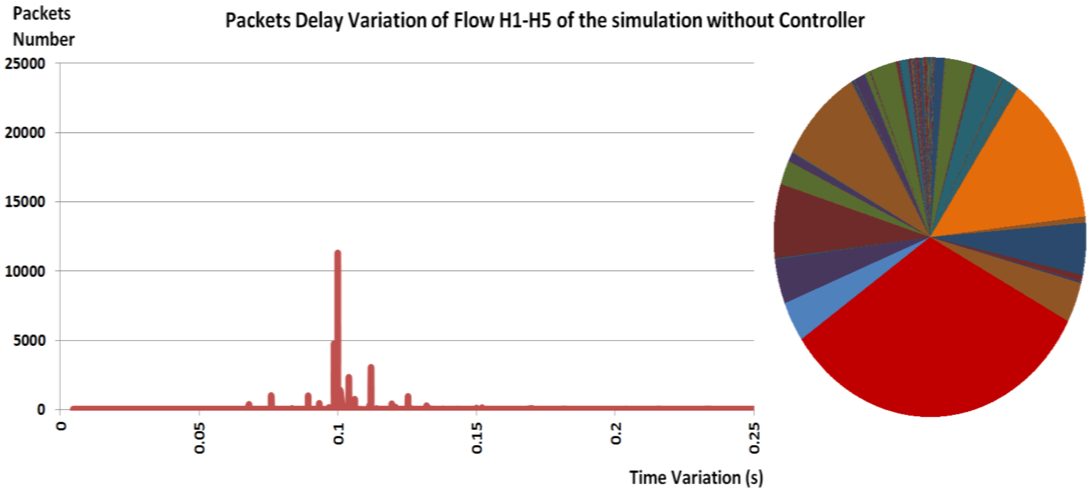


Fig. 4-30 PDV of flow H1-H5 (without controller)

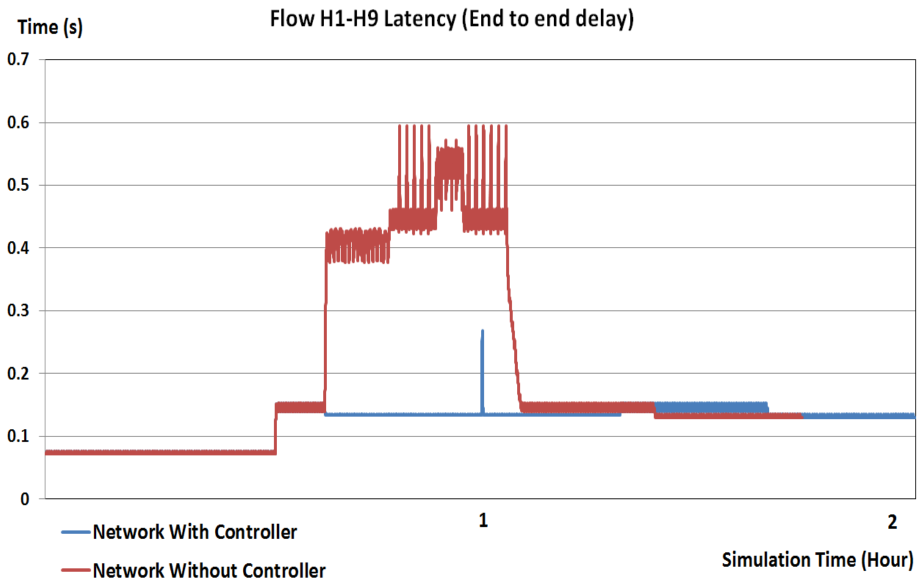


Fig. 4-31 Flow H1-H9 latency (with and without controller)

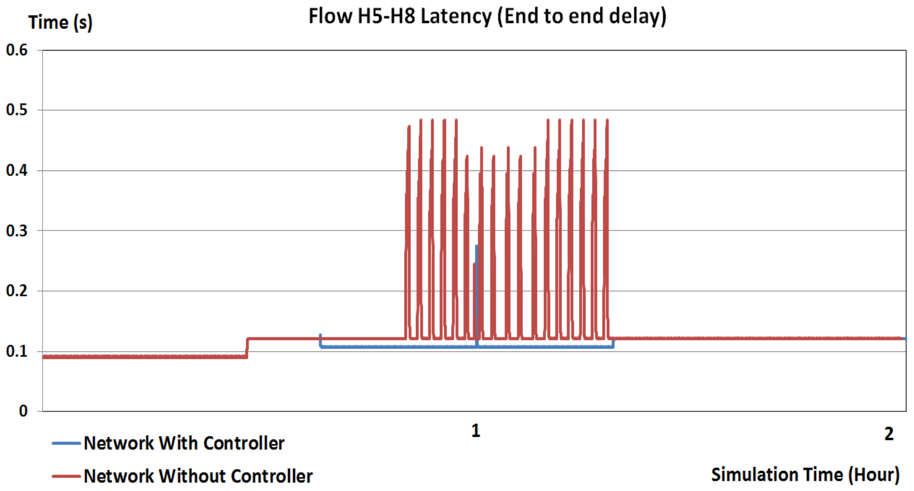


Fig. 4-32 Flow H5-H8 latency (with and without controller)

Fig. 4-33 represents the total recorded network delay when applying this placement where the network with controller has less total network delay.

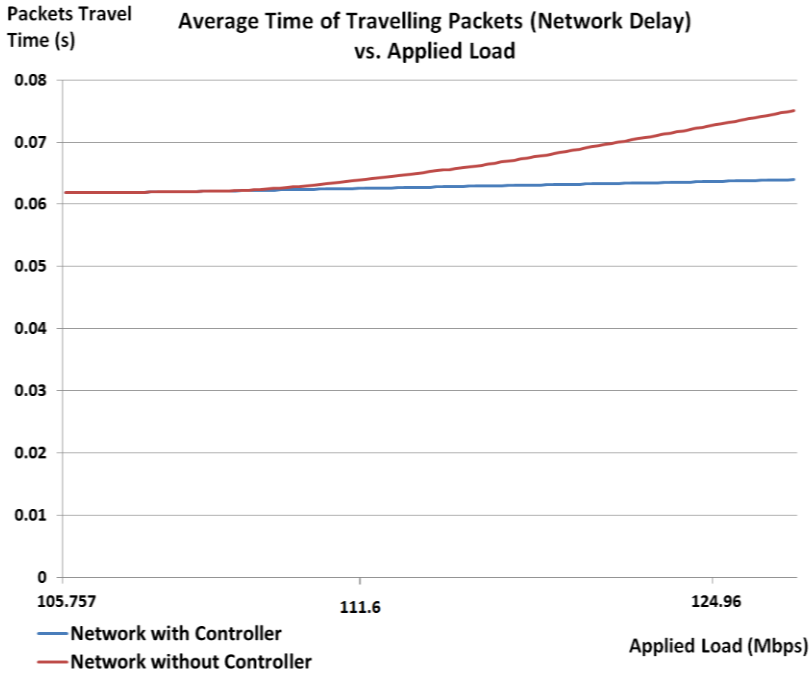


Fig. 4-33 Network delay (with and without controller)

It is obvious from all plots that the network of controller performs better than that of without controller.

4.11 Residual Bandwidth Safety Factor

In residual bandwidth calculations, there is about 5 percent of extra bandwidth added by the controller as a *safety factor*. This bandwidth compensates for the minor changes in flows' data rates, which are not reported to the controller. The residual bandwidth allocated within the *safety factor* is also employed to carry the *Real time updating* messages and the *Control* messages. The head-end router informs the controller about the change in flow's data rate only when it exceeds the *safety factor* value.

4.12 Throughput Results

For all FDA placements, there is higher average throughput recorded of the network with controller than that without controller regarding the same applied traffic load. Fig. 4-34 shows the average throughput recorded in: *Re-diverting the previously diverted flow - Re-diverting via longer None OSPF Path* of **sub-section 4.10.4.ii**.

The difference in throughput recorded values represents the packets dropped from the queues due to the congestion. When the applied traffic returns to a moderate value, the throughput plots with and without controller simulations become parallel to each other.

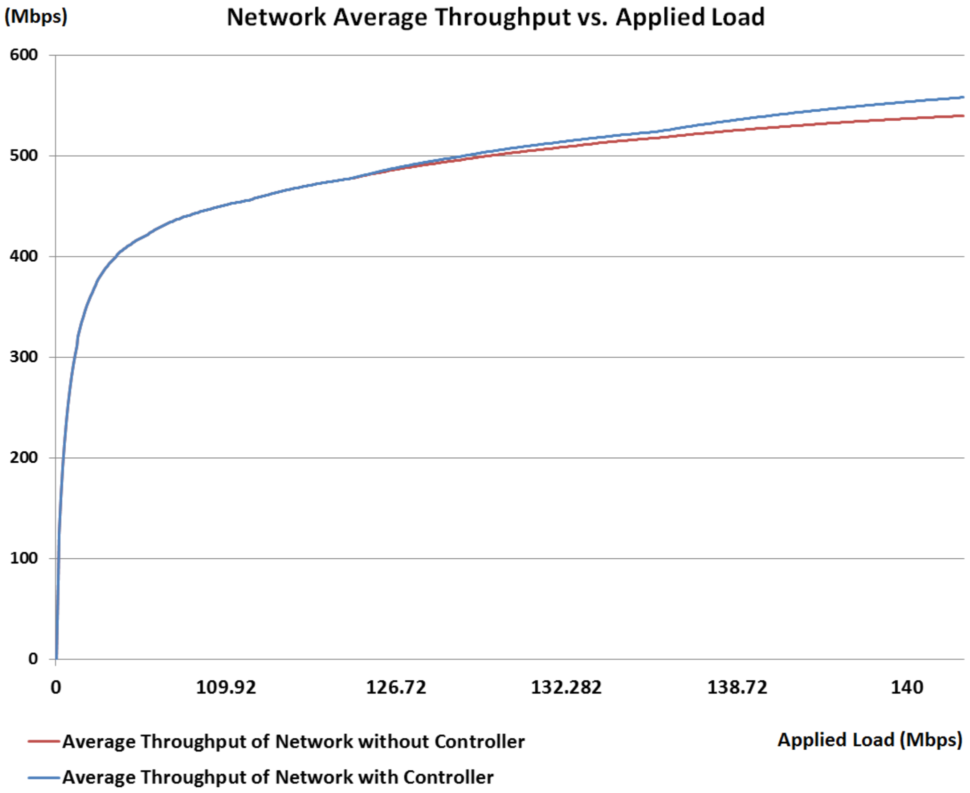


Fig. 4-34 Average throughput recorded for both with and without controller simulations along with same applied data rate

For all results obtained from all cases, the existence of the controller and Applying FDA provided the network with the following improvements:

1. Eliminating or reducing packets drop as much as possible.
2. Reducing network delay by choosing the best available paths, which reduce flow delay.
3. Regulating the Packet Delay Variation (PDV) by reducing the queueing delay effect.
4. Providing better usage of the available bandwidth.
5. Avoiding diverting flows out of their OSPF path unless necessary.
6. Preventing the *route flapping* effect.

4.13 Summary of Chapter 4

- The approach discussed in this chapter improves the performance of computer networks regarding the distribution of the IP routed flows by creating temporary flow tables in the network layer of some routers and use those the flow tables in forwarding specific flows instead of the OSPF routing table.
- The controller improves the performance of the network up to a specific level of applied data rates. When exceeding that level, the controller fails to protect all packets from dropping. However, the network with controller still behaves better than its counterpart without controller.
- The controller behaviour depends on the topology, the available resources, the applied traffic pattern and its data rate.
- In a router, there can be different flows directed to the same destination. Some of flows are forwarded according to the flow tables via specific gateways while the others are forwarded according to the OSPF routing table via the usual gateway.
- The location of the controller inside the network affects the time consumed to inform the controller of the updates of flows' status change and to receive the changes related to flows distribution, which are represented by the *Control* messages. The controller should be placed at the middle of the network and to be near as much as possible to most of the routers.
- The controller only intrudes to deal with congestion problem. This provides semi-management to the network and more independence to the routers than the SDN system. The limited number of *Control* messages issued by the controller saves more bandwidth.
- Increasing the *safety factor* value reduces the number of the *Flow status updating* messages sent to the controller but it also reduces the accuracy of bandwidth allocation to the flows.
- In cases of the controller's breakdown or the failure of all the links that connect the network with the controller, the network works as a normal OSPF network. The routers can update their routing tables (if there is a topology change) without needing the controller. In SDN, the routers fully depend on the controller.
- The congestion effect does not badly affect the flows passing through the congested links only. It extends to affect the flows passing through many other links. This because of the *route flapping* effect.

Chapter 5

Improvement of MPLS Technology by Adopting the SDN Notion

Introduction to Chapter 5

This chapter discusses the Multi-Protocol Label Switching (MPLS) technology and the improvement that our research adds to it. It represents the second contribution of our research. At the beginning, it explains the most common aspects of the MPLS system from the technical point of view. It explains the theory of the MPLS and gives several basic definitions related to it. Later, it shows some of the problems and challenges facing the MPLS technology and discusses briefly our proposed solution to deal with them. Then, it illustrates some of the development efforts that achieved by other researchers. After that, it shows in details the methodology used to apply our proposed solution in improving the MPLS performance. Finally, it displays the results obtained from applying our solution which confirms the improvement added to the system.

Before reading this chapter, we recommend reading the white paper of title: “Advanced topics in MPLS-TE deployment” [9] published by Cisco systems incorporated. This chapter and the next chapter (**chapter 6**) show an improvement to the technology illustrated in the mentioned publication.

5.1 MPLS Theory

The gigantic growth of the internet and the intranet networks creates a continuous challenge to service providers and equipment suppliers in terms of enormous escalation in traffic [9]. This growth is accompanied by a growth in routing table size.

In conventional network layer forwarding mechanisms, when a packet traverses the network, each router extracts all the information relevant to forwarding the packet from its header. The extracted information is used in the routing table lookup operation to detect the next hop for the packet. Each router along the path repeats this complicated table lookup operation [25]. This operation may take long time if the routing tables are large and the routers lie in a huge network. MPLS technology provides the solution.

The MPLS is a new technology that combines the intelligence of routing with the performance of switching. During the entrance of packets to the MPLS domain, labels are attached on the packets, and the label (instead of the IP header) determines the next hop. Labels are taken off at the egress of the MPLS domain [9]. MPLS benefits can be summarized in applications of: Virtual Private Networking (VPN), Traffic Engineering (TE), Quality of Service (QoS) and Any Transport over MPLS (AToM). MPLS reduces the forwarding overhead on the core routers [78].

The MPLS path is also known as Label Switched Path (LSP), LSP begins at head-end router and terminates at tail-end router [9]. Packets crossing the LSP have the same label [38].

5.2 MPLS Traffic Engineering

MPLS Traffic Engineering (MPLS-TE) comprises [79] the application of MPLS technology and scientific concepts to the measurement, modeling, characterization, and control of Internet traffic and the application of such knowledge and techniques to achieve specific performance. Features of MPLS Traffic Engineering are:

- Enhances standard IGPs, such as OSPF, to automatically map packets onto the appropriate traffic flows.
- Transports traffic flows over a network utilizing MPLS forwarding.
- Transports packets using MPLS forwarding passing through a multi-hop MPLS.
- Sets the routes for traffic flows via a network based on the resources that the traffic flow requires and the available resources of the network.
- Recovers from link or node failures by adapting to the new constraints given by the changed topology.

5.3 Some Fundamental Definitions of the MPLS Technology

Below, some of the important definitions related to the MPLS technology which will be dealt with later in this chapter.

5.3.1 RSVP

The Resource Reservation Protocol (RSVP) is a network control protocol. It dwells at the place of a transport layer protocol within the OSI model seven layer protocol stacks [80]. The benefit of using RSVP to create MPLS tunnels is that it enables the allocation of resources along the path [81]. The RSVP maintains the following data structures:

i. Path State control Block (PSB) list

PSB holds path state from the *RSVP-TE PATH* message for each session and sender pair [82].

ii. Reservation State control Block (RSB) list

RSB holds a reservation request that arrived within a particular *RSVP-TE RESV* message, corresponding to the triple: session, next hop, and filter spec list [82]. The filter spec list is a data structure object.

iii. RSVP messages

There are two fundamental RSVP message types: *RESV* and *PATH* [80]. The frame of the RSVP message contains a common header, followed by a body consisting of a variable number of variable-length, typed "objects" [80].

5.3.2 MPLS Tunnel Related Objects

i. Label object

The generic MPLS label is an unsigned integer in the range from 0 through 1048575. The term "*same label*" refers to the identical label value obtained from the identical label space [81].

ii. Explicit Route Object (ERO)

ERO specifies explicit path information. It consists of a series of variable length data items called sub-objects [81].

iii. Record Route Object (RRO)

Paths are recorded through the RRO. RRO can exist in both *RSVP-TE PATH* and *RSVP-TE RESV* messages [81].

5.3.3 Label Information Base (LIB) and Label Forwarding Information Base (LFIB)

The Label Switching Router (LSR) uses the Label Information Base (LIB) as database to store labels learned from other LSRs, as well as labels assigned by the local LSR [39].

The LSR uses Label Forwarding Information Base (LFIB) table to forward labeled packets. LFIB selects only one of the possible outgoing labels from all the possible remote bindings existing in the LIB and installs it. Choosing the remote label depends on which path is the best path found in the routing table [37] pp 35-36. Thus, LFIB represents the data structure used by switching functions to switch labeled packets [39].

5.3.4 Forwarding Equivalence Class (FEC)

A Forwarding Equivalence Class (FEC) is a group or flow of packets that are forwarded in the same manner via the same path. All packets that belong to the same FEC have the same label. The ingress (head-end) router decides which packets belong to which FEC; as it is responsible of classifying and labeling the packets [37] pp 31. The MPLS network needs a distribution protocol to distribute the labels between LSRs [37] pp 40.

5.4 Tunnel Establishment Procedure in Conventional Networks

MPLS-TE creates and maintains LSPs by using Resource Reservation Protocol (RSVP)

[25]. The Constrained Shortest Path process starts at the head-end router to create a LSP by using *RSVP-TE* messages [9]. The establishment of the LSP tunnel uses two types of RSVP messages: The *RSVP-TE PATH* and *RSVP-TE RESV* messages. The head-end router sends the *RSVP-TE PATH* messages to the tail-end router, which replay by sending the *RSVP-TE RESV* messages which take the same opposite path to the sender. The routers across the path consider the bandwidth factor in creating the tunnel. If the available bandwidth of any link across the path is not enough to fulfill the requirements of the tunnel, *RSVP-TE PATH* message will stop at that point (without carrying on to tail-end router) and a *PATH Error* message will be sent to head-end router without tunnel creation [37]. The router that supports MPLS is called Label Switching Router (LSR) [9].

Similar to any network forwarding system, MPLS is divided into control plane and data plane. The MPLS control plane protocols like Label Distribution Protocol (LDP) employs IP routing to set up the label switched paths but the MPLS data plane does not use IP routing in data forwarding operation [83].

5.5 Research Problems

LSRs use routing protocols extensions to create and maintain a TE Link State database (TE-LSDB) [9]. Constraint Based Routing (CBR) algorithm is used to find the best path for an LSP tunnel. Constraint Based Routing (CBR) algorithm may also be called Constrained Shortest Path First (CSPF) algorithm. CSPF is applied on the head-end router [8]. CSPF uses either the IGP metric or link metric to find the shortest path [84]. Through CSPF, the head-end router uses Traffic Engineering (TE) topology database in finding the path to destination [85].

Even though the path discovered to the destination is the shortest, it does not mean it is the best to be used during the recent time. Some of remote links can be occupied by or allocated to other routed flows. Reserving the bandwidth of those links for a MPLS path affects the distribution of other forwarded flows. This may cause a link congestion problem.

The bandwidth size of a MPLS Traffic Engineering (TE) tunnel is automatically adjusted by *Cisco IOS MPLS AutoBandwidth allocator*, which works as shown in Fig. 5-1. The re-adjustment of tunnel bandwidth depends on the largest average output rate noticed during a certain interval X. This involves monitoring the applied flow data rate every interval (several minutes), records the highest value and uses it as a reference to adjust the tunnel bandwidth during the next interval Y, which takes longer time (hours) [9]. At period N, a dot represents the highest bandwidth noticed during this period. This value will then be used at period N+1 to signal the adjusted LSP; the network provider is capable of bandwidth management and optimizing traffic [9].

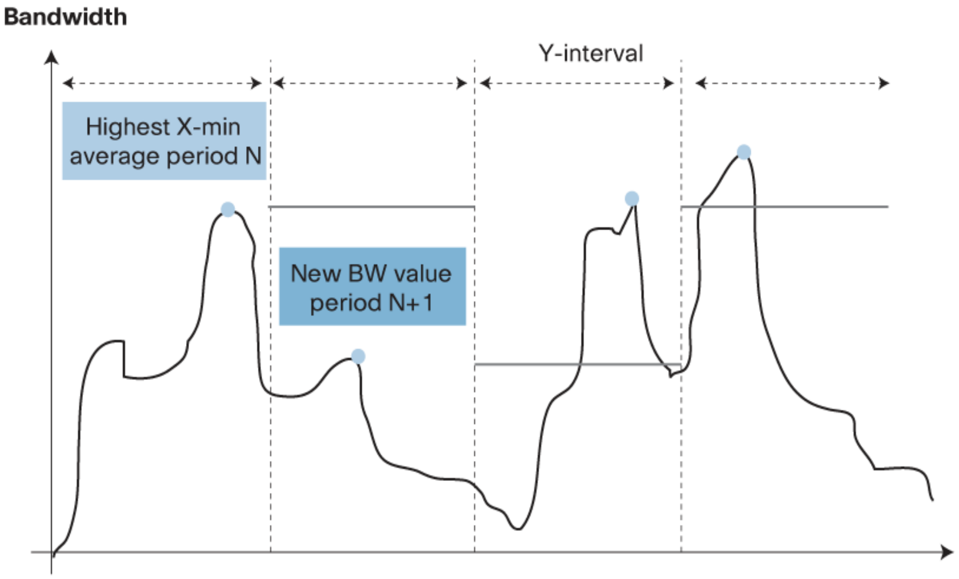


Fig. 5-1 Illustration of Cisco MPLS bandwidth allocator adjusting bandwidth over time [9]

The tunnel bandwidth re-adjustment is not instantaneous. If the expectation of tunnel bandwidth of the next interval does not meet the real change of flow applied data rate, either the reserved bandwidth will be more than the required, which may consider a waste in using the bandwidth of the network, or the reserved bandwidth is less than required, which may cause link congestion.

Tunnel path restoration is performed by the head-end router. At the event of congestion or topology change, the head-end receives notification by RSVP-TE that the path cannot be maintained. Immediately, it constructs a new TE database after removing the faulty links or area of congestion. This operation takes from **2-3** seconds [9]. This is considered a long time of outage. Furthermore, if there are several tunnels that share an area of congestion and those tunnels were established by different head-end routers then there should be coordination among them on how to deal with their congested tunnels (which tunnel path should be diverted and which should not).

From the above discussion, several problems are identified:

- The link congestion problem.
- The delay in tunnel bandwidth re-adjustment.
- The inaccuracy in tunnel bandwidth re-adjustment.
- The delay in path restoration (inflexibility of the path of the tunnel) when dealing with the congestion problem.
- The necessity of coordination among the head-end routers.

5.6 Research Objectives

The aim of this chapter is to enhance the performance of the OSPF/Multi-Protocol Label Switching (MPLS) routed network by adding some features of the Software Defined Networking (SDN) in a modified approach that reduces the time consumed in tunnel establishment, maintenance and restoration. The bandwidth reserved for the tunnel is almost equal to the data rate of the flow passing through it. This improves the network performance in dealing with the critical systems and provides better resources utilization.

5.7 Related Work

There are several development efforts on MPLS technology in both conventional and SDN networks.

5.7.1 Tunnel Establishment and Maintenance Efforts

MPLS performance development has been studied in conventional routing networking, especially the Open Shortest Path First (OSPF) routing protocol.

All the proposed algorithms that are used to dynamically establish MPLS tunnels are applied at the ingress of the tunnel (the head-end router) [10-12]. This assumes the existence of signaling protocols like RSVP-TE or CR-LDP are responsible for the updating of the information related to topology changes and residual bandwidth.

The Minimum Interference Routing Algorithm (MIRA) proposed in [10] tries to decrease the possibility of selecting a route that "interferes" with future requests of other *ingress-egress* pairs. It uses the residual bandwidth as the main factor on its calculations by identifying the critical links belonging to the other *ingress-egress* pairs in the network and assigning higher weights for these critical links. It applies Dijkstra algorithm on the graph to avoid the critical links. According to [10], the *residual bandwidth* along a link is defined as the difference between the total bandwidth of a link and the sum of the LSP demands that are routed through that link.

Light Minimum Interference Routing (LMIR) algorithm [11] is a development of MIRA. LMIR attempts to optimize resource utilization with low computational complexity. Similar to MIRA, LMIR finds paths with lowest capacities to determine the critical edges, assigns weight to them and executes Dijkstra algorithm to select the non-critical edges.

Another development to MIRA is proposed by Zhu et al in [86] and named BU-MIRA. BU-MIRA considers the current bandwidth available of the links in the network in addition to the traffic flow distribution. It aims to minimize the interference among the competing flows by balancing the number and quantity of flows carried by a link to attain an efficient routing of bandwidth guaranteed LSPs. BU-MIRA uses specific criteria for creating a weighted graph, which avoids routing LSPs on critical links if possible. A weighted graph is established where the critical links have a weights increasing function along with their criticality, which defer loading the critical links whenever possible. Zhu et al define the critical links as the links with the feature that whenever an LSP is routed over them, the max flow values of one or more *ingress-egress* pairs of routers decreases.

A Hop-Constrained Adaptive Shortest-Path Algorithm (HCASP) is an algorithm for routing of MPLS bandwidth-guaranteed tunnels proposed in [12]. HCASP has two goals: the first is to choose MPLS path with a limit on the length so as not to largely exceed the length of the shortest-hop path between the *ingress-egress* pair; the second is to give preference to less loaded links during the MPLS tunnel creation time to improve the load balance.

Informing the head-end router by using RSVP-TE with feedback is discussed in [38, 87], where there is an ability to attach actual link bandwidth availability information at every link that the signaling message traverses. This feature is added to the signaling protocol and it allows the head-end router to receive very up-to-date information, which is attached to its topology database. The information obtained from the feedback message is used on further source route computations. Those feedback messages may consume a portion of the bandwidth and some processing if the number of the created tunnels is huge.

All proposed algorithms are applied at the head-end routers, which need for a coordination among them to achieve better bandwidth utilization. Our proposed algorithm is applied at the controller which provides central management to flows distribution and better bandwidth occupation.

Another privilege of our approach is that it does not need any feedback messages as the proactive controller knows the flows distribution and all links’ residual bandwidth during the real time. It is familiar with the amount and the location of the congestion as well which makes it enable to behave proactively.

5.7.2 MPLS and OpenFlow

The OpenFlow protocol 1.0 standard version does not support the MPLS technology [83]. However, the research of [83] discusses the design and implementation of an experimental extension of OpenFlow 1.0 to support MPLS. Because of this extension, the OpenFlow switch without IP routing capability can now forward MPLS on the data plane. The switch data plane in OpenFlow is designed as a flow table in which there are three columns: rules, actions, and counters. The rules column determines the flow. Each rule consists of elements from a ten-tuple of header fields. The most important extensions to the OpenFlow include:

- Increasing the size of the tuple used for flow identification.
- Adding of the MPLS header modification actions (push, pop, and swap).
- Adding a counter to the OpenFlow statistics that is incremented every time a virtual port drops a packet due to the expiration of the TTL.

Our approach differs from this approach as it develops a conventional OSPF/ MPLS routed network by adopting the SDN notion.

In [88], Tu et al. suggest a method of splicing MPLS and OpenFlow tunnels based on SDN paradigm. A global view controller that is provided with information collector, path translator and command installer is responsible of generating MPLS label forwarding rules and OpenFlow entries inside the routers.

5.7.3 Other MPLS Development Efforts

In [89], Hao et al. present an algorithm that selects paths for QoS traffic and best-effort traffic. The framework of traffic engineering consists of: optimal computing, getting the path set from the result of the optimal computing and on-line routing, which selects paths for QoS traffic. The on-line routing algorithm selects narrowest shortest paths for QoS traffic and light load path for the best-effort traffic.

In our approach, the controller is flexible in choosing or changing the path depending on the real time requirements of the network.

Establishment of MPLS tunnel that traverses multiple OSPF areas is discussed in [90].

5.8 Network Model Features

This section presents the modified network and its models. The network has been designed by using OMNeT++ simulator through modifying its Inet-OMNeT project. The OMNeT++ simulator of version 4.3 supports the MPLS technology regarding establishing explicit tunnel in its Inet project.

5.8.1 Network Topologies

Our project examines several OSPF/MPLS networks of different topologies. The primary modified network consists of 9 routers and 9 hosts. It is the same network used to explain FDT and FDA of **chapter 4**. It is also shown here in Fig. 5-3. The other topologies are shown in Figs. 5-11 and 5-13. The topologies of the networks are constant during the simulation time. Neither topology change nor link or node failure occurs.

5.8.2 Network Messages

The main messages that used by the network were explained in **section 3.9** of **chapter 3** before. The most important messages that used to develop the MPLS are the RSVP standard and developed messages. The development in RSVP messaging system to improve the performance of the network is as shown below:

- No *RSVP-TE PATH* messages are used to establish the tunnel.
- *RSVP-TE RESV* messages are improved to substitute the duty of *RSVP-TE PATH* messages as well.
- *RSVP Final hop resv* messages are new type of *Control* messages that are established at the *Developed RSVP* model of the controller and received at the *Developed RSVP* model of the tail-end router.
- *RSVP Remove tunnel* messages are new types of *Control* message that are established and used in the *Developed RSVP* models. It is used to remove the existing MPLS tunnel.

The routers forward those messages according to their OSPF/IP routing tables. In the next sections, those messages will be used and their structures will be shown.

5.9 The Controller in Dealing with MPLS Technology

The controller architecture was explained in **chapter 3** and its performance when dealing with flows and congestion was viewed in FDT and FDA, which is explained in **chapter 4**. Making the controller eligible to manipulate the *MPLS* model inside the router involves providing it with an ability to affect the router's *RSVP* model. Inserting a *RSVP* model inside the controller provides it with the ability to contact its counterpart *RSVP* model of the router.

5.9.1 Controller Capabilities

The controller deals with MPLS according to one of four actions:

- Creating an MPLS tunnel.
- Re-adjusting the reserved bandwidth of the MPLS tunnel.
- Removing the existing MPLS tunnel.
- Changing the path of the MPLS tunnel.

The first and third actions are applied according to a request from the sender host, while the second and forth actions are applied according to the network's operation requirements.

5.9.2 Network Gain of Controller's Behavior

The problems discussed in **section 5.5** (Research Problems section) are solved as follows:

- The link congestion problem has been solved by using the pro-active real time controller that distinguishes congestion and selectively manages flows according to FDA.
- The controller is able to re-adjust the reserved bandwidth of the tunnel immediately and exactly according to its applied flow data rate. There is no waste in bandwidth allocation.
- The tunnel can be replaced with another tunnel that passes through another path and serves the same *source–destination* pair if the existing path is not suitable at the meantime. The creation time of the new tunnel path is short.
- There is no need for coordination among the head-end routers as the tunnel creating operation is the responsibility of the controller alone.

5.9.3 RSVP Model of Controller

The controller’s *RSVP* model represents a development to the standard *RSVP* model designed by Inet-OMNeT. It is also managed by the *Operations* model. The controller uses this model to contact and manipulate the *RSVP* model of the routers via ordering the *RSVP Final hop resv* and the *RSVP Remove tunnel* messages which are created inside this model. Both the *RSVP Final hop resv* and the *RSVP Remove tunnel* messages represent the *Control* messages of the controller regarding the MPLS technology. The controller assigns the tunnel’s identifications and attributes and adds them to those messages. The *Developed RSVP* model of the controller lies at the Transport Layer of the DoD model as was shown in Fig. 3-4 of **chapter 3** before.

5.10 The Developed MPLS Router

The router design was discussed in **chapter 3**. It represents the standard OSPF/IP-MPLS router of the Inet-OMNeT project. Its architecture was shown in Fig. 3-5 of **chapter 3** before. The models that related to the MPLS technology and serve the requirements of this chapter are: *Developed RSVP* model, *Developed MPLS* model, *Developed Simple Classifier* model and *Developed LIB table* model.

5.10.1 Developed RSVP Model of Router

The *RSVP* model of the router represents a development to the standard *RSVP* model designed by Inet-OMNeT. The code of the standard *RSVP* model has been modified by sub-classing to enable it to perform extra duties. The main improvements to the code of the RSVP model are as following:

- The *Developed RSVP* model of the router will no more read the tunnel's information from a **xml file** as the main design of the Inet-OMNeT.
- The *Developed RSVP* model of the router receives the *RSVP Final hop resv* messages that are sent by the controller, the *RSVP-TE RESV* messages that are sent by a neighbour router and the RSVP *Remove tunnel* messages that are sent by the controller or a neighbour router, reads the tunnel’s attributes from them and continues in constructing or removing the tunnel as usual.
- Both *rsb timeout* and *hello timeout* timers are abolished.
- The capabilities of the *Developed RSVP* Model of the router are extended so that it can manipulate the contents of the *LIB Table* model through the *Simple Classifier* model.

i. The reason behind removing the *hello timeout* timer

When RSVP signals a TE LSP and at the same time there is a failure somewhere along the path, then it may take long time until the failure is detected. The MPLS Traffic Engineering—RSVP Hello state timer feature discovers when a neighbour is down and initiates faster state timeout. This releases the resources such as bandwidth to be reused by other label-switched paths (LSPs) [91]. Hellos enable RSVP nodes to detect when a neighbouring node is not reachable. In our project, it is the responsibility of the controller to remove the reserved bandwidth from the routers (resources freeing). Using the *hello timeout* timer conflicts with the controller’s duties of removing the tunnel, and its reserved bandwidth.

ii. The reason behind removing the *rsb timeout* timer

The RSVP *Teardown* messages are types of RSVP messages that remove path or reservation state immediately [92]. The *rsb timeout* timer is responsible of creating the *ResvTear* message. The *ResvTear* message travels towards all senders upstream from its point of establishment [92]. This message deletes the Reservation State control Block (RSB) list regarding the specific tunnel. In our approach, this is the duty of the controller to remove the tunnel and all its identifications and attributes from the involved routers. Therefore, the *rsb timeout* timer is abolished.

5.10.2 Developed MPLS Model of Router

The *MPLS* model of router represents a development to the standard *MPLS* model designed by Inet-OMNeT. The improvement concentrates on dealing with the Fast Re-routes (FRR) mechanism that to be explained during the next chapter (**chapter 6**). It also records the data rate forwarded through each interface and sends the recorded values periodically to the router’s *Listener* model. The *Developed MPLS* model of a router tests any arrived packet and forwards it according to its type. If the packet’s type is IPv4 datagram, the *MPLS* model forwards it to the *Network layer* model of the router else if the packet’s kind is MPLS format, the *Developed MPLS* model examines its label and forwards it to the output interface according to the attached label. The labels are invoked from the *LIB Table* model.

5.10.3 Developed Simple Classifier and Developed LIB Table Models of Router

Both standard *Simple Classifier* and *LIB Table* models are modified to accept the intrusion of the *Developed RSVP* model of the router in terms of adding, removing or modifying tunnels' attributes and identifications. The *Developed RSVP* model adds or removes tunnels' attributes in the *LIB Table* model through the *Simple Classifier* model.

5.11 The MPLS Tunnel by the Controller

In principle, our network performance regarding flows distribution under OSPF/MPLS development effort is not different from its performance under OSPF/IP development effort that was discussed in **chapter 4**. The difference is that the controller here is completely responsible of creation and management of the MPLS tunnel.

For every new flow or obvious change in the data rate of an existing flow that exceeds the *Safety Factor* value, the head-end router will normally deal with the flow by sending a *Flow status updating* message that contains the flow current information to the controller.

The source host (the sender host) of the flow requests the head-end router to create a MPLS tunnel to the destination. The head-end router inserts this request of tunnel establishment inside the *Flow status updating* message that is sent to the controller.

When the controller receives the *Flow status updating* message of the head-end router, it activates its FDT. The FDT models the network and examines the flow mimicry distribution over the OSPF paths of the network. If FDT notices any possible congestion over the main OSPF path, it activates the FDA to find an alternative path to the tunnel. The controller chooses the MPLS path according to the results obtained from FDT or FDA. In addition to the tunnel establishment, the controller uses the capabilities of FDT and FDA to re-adjust the bandwidth of the tunnel or change its path (if necessary). The controller removes the tunnel if the source host requires it to do so.

5.11.1 Creating of a MPLS Tunnel

The controller allocates the path of the tunnel according to the decision of the FDT or FDA inside it. The tunnel creation procedure is as follows:

1. The controller allocates the path from source to destination. The path is a list that consists of several adjacent routers (nodes) and each router has an address. It begins at the head-end router N_1 and terminates at the tail-end router N_k. The chosen path can handle the flow without any congestion.
2. Inside the *Operations* model of the controller, the controller assigns the features and the attributes of the tunnel represented by tunnel path list, tunnel bandwidth, and flow *source-destination* pair addresses and flow's destination. The controller assigns specific values for tunnel's identifications represented by: *tunnel ID, tunnel LSP ID* and *tunnel color,* and stores all the mentioned information in specific container.

3. The controller does not give the same identifications to more than one tunnel. Each tunnel has its specific identifications which are considered global.
4. The controller creates *RSVP Final hop resv* message inside its *Developed RSVP* model; It inserts tunnel's attributes and identifications into the created message, encapsulates it with IPv4 datagram at the *Network layer* model and sends it to the tail-end router, N_k, of the tunnel. The structure of *Final hop resv* message is shown on Fig. 5-2. If this tunnel is new and not replacing an existing one, then the controller sets the field *Previous Tunnel ID* value to zero.
5. The *Final hop resv* message moves inside the network according to the IP-OSPF routing tables of the routers until reaching its destination (the tail-end router N_k).
6. The tail-end router receives the arrived *RSVP Final hop resv* message, de-capsulate and sends it to its *Developed RSVP* model.
7. The router's *Developed RSVP* model extracts the tunnel identifications from *Final hop resv* message and uses them to create Explicit Route Object (ERO), Record Route Object (RRO), Path State control Block (PSB) list and Reservation State control Block (RSB) list by sub-classing.
8. Upon receiving RSVP *Final hop resv* message, the router's *Developed RSVP* model creates internal timer messages: *Rsb Commit Timer* message and *Rsb Refresh Timer* message.
9. After receiving *Rsb Refresh Timer message*, the router *Developed RSVP* model creates a *RSVP-TE RESV* message, loads tunnel identifications into it and sends it to the penultimate router of address N_{k-1} within the path list.
10. The *RSVP-TE RESV* message created by *Developed RSVP* model represents the standard *RSVP-TE RESV* message designed by the Inet-OMNeT++. It has been modified by adding some attributes to substitute the duty of *RSVP-TE PATH* messages and carry some extra information. Therefore, *RSVP-TE RESV* message carries the same information of the *Final hop resv* message in addition to its standard design information.
11. The penultimate router receives *RSVP-TE RESV* message, de-capsulates it and repeats steps 6, 7, 8 and 9 above performed by the tail-end router. The *Developed RSVP* of the penultimate router creates a new *RSVP-TE RESV* message, loads it with tunnel attributes and sends it to the next previous router in the MPLS path list N_{k-2}.
12. This operation is repeated at each router via the MPLS tunnel path until reaching the head-end router N_1, where no more *RSVP-TE RESV* messages are issued.

13. At the head-end router, the tunnel establishment operation finishes. Tunnel attributes are also added to *Simple Classifier* model to build the *Label Information Base* (*LIB*) table. The *bindings* container vector is loaded with the *FECEntry* values represented by the *source-destination* and the tunnel's identifications. This enables MPLS model of the head-end router to push a label into the flow packets entering the MPLS domain.
14. During the tunnel's establishment procedure, every router the *RSVP-TE RESV* messages pass through sends an *Acknowledge* message that contains the operation *code* number which was inserted in the *Final hop resv* message (as well as inserted in the *RSVP-TE RESV* message) to the previous router to confirm the successful arrival of *RSVP-TE RESV* message. If the previous router does not receive the *Acknowledge* message within specific period of time after sending the *RSVP-TE RESV* message, it will re-send a copy of the *RSVP-TE RESV* message.
15. The head-end router sends another *Acknowledge* message to the controller. It also contains the operation *code* number. The *Acknowledge* message confirms the successful operation of MPLS tunnel construction. Without receiving the *Acknowledge* message within specific time, the controller will repeat the tunnel construction operation.
16. Tunnel establishment time starts when the head-end router commences to contact the controller via establishing the *Flow status updating* message that requests tunnel establishment and finishes when the head-end router receives the *RSVP-TE RESV* message related to the tunnel.

Flow Source	**Flow Destination**	**MPLS Path:** $N_1\ N_2\ N_3\ N_4\ \ldots\ldots\ldots\ldots N_k$	**Code**
Tunnel Attributes and Identifications: Tunnel Number, Tunnel BW, Tunnel ID, Tunnel LSP ID, Tunnel Colour, Previous Tunnel ID			

Fig. 5-2 RSVP Final hop resv message structure

Flow Source**:** Host address where the flow begins.

Flow Destination**:** Host address where the flow terminates.

$N_1\ N_2\ N_3\ N_4\ \ldots\ldots\ldots\ldots\ldots N_k$: Addresses of routers where MPLS path should passes through.

N_1: The address of the head-end router.

N_k: The address of the tail-end router.

Tunnel's Attributes and Identifications**:** Tunnel Number, Tunnel BW, Tunnel ID, Tunnel LSP ID, Tunnel Colour, Previous Tunnel ID.

Code: A specific number imposed by the controller. It is used in the *Acknowledge* messages to confirm the arrival of the *Control/RSVP* message as well as to confirm the successful establishment of the tunnel.

Example, suppose network shown in Fig. 5-3 where host H5 commences sending a flow of *Data* packets to host H4. H5 wants flow H5-H4 to pass through a MPLS tunnel. H5 begins sending the *Data* packets of flow H5-H4 to router 5 implying its request of building the tunnel. Router 5 sends a *Flow status updating* message to the controller illustrating flow source, flow destination, flow data rate and the MPLS tunnel establishment request. The controller applies its FDT upon the flow characteristics. The result is creating the MPLS tunnel that passes through routers 5-1-3-4. This path represents the OSPF distribution path as there is no congestion via it. The controller assigns tunnel identifications as in Table V-I:

TABLE V-I
ALLOCATED IDENTIFICATIONS TO A FIRST ESTABLISHED TUNNEL

Tunnel Number	41
Tunnel ID	92
Tunnel LSP ID	142
Tunnel Color	192
Previous Tunnel ID	0

The controller allocates those identifications to the MPLS tunnel that passes through 5-1-3-4 and carries flow H5-H4 only. No other tunnel will have any of those values unless flow H5-H4 tunnel is completely removed.

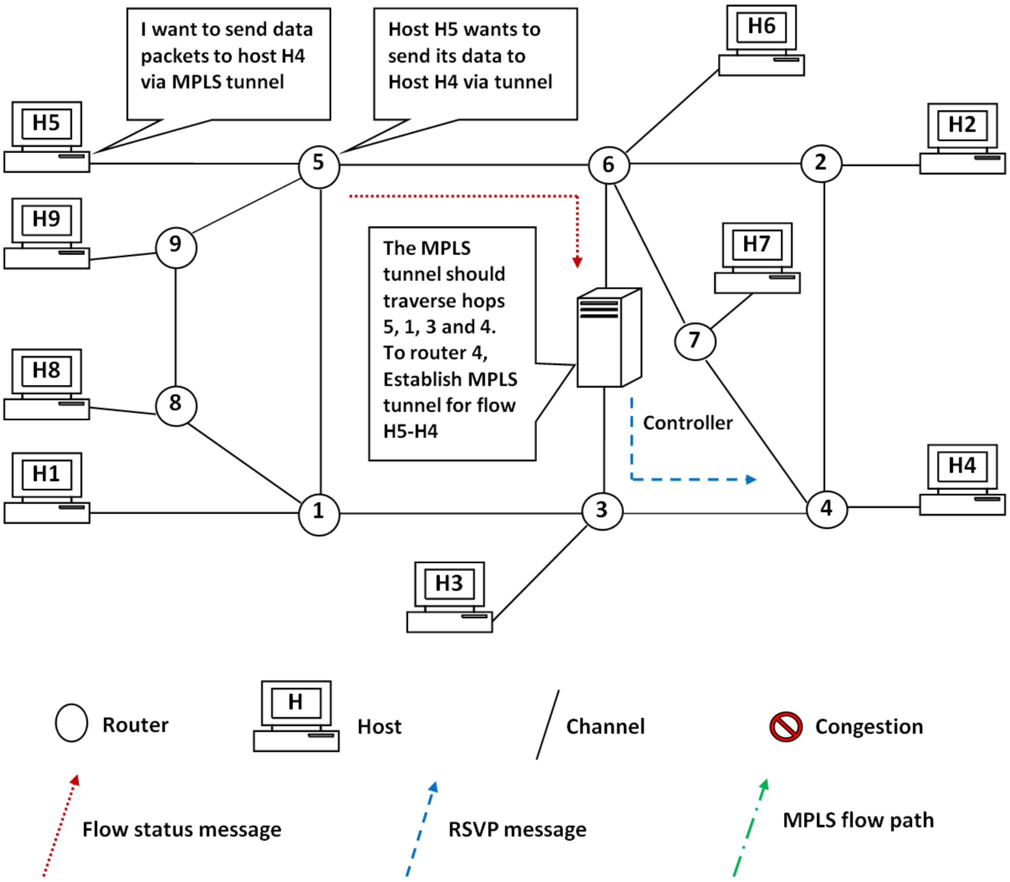

I want to send data packets to host H4 via MPLS tunnel
Host H5 wants to send its data to Host H4 via tunnel
H6
H5
H2
5
6
2
H9
9
H7
The MPLS tunnel should traverse hops 5, 1, 3 and 4. To router 4, Establish MPLS tunnel for flow H5-H4
7
H8
8
Controller
H1
H4
1
3
4
H3
Router
H
Host
Channel
Congestion
Flow status message
RSVP message
MPLS flow path

Fig. 5-3 First steps to establish MPLS tunnel that serves flow H5-H4

The controller creates *RSVP Final hop resv* message, inserts tunnel identifications and attributes inside it and sends it to router 4 (which is the tunnel tail-end router) as shown in Fig. 5-3.

Router 4 uses its *Developed RSVP* model to record the attributes and the identifications of the MPLS tunnel. The *Developed RSVP* model uses tunnel's attributes and identifications to create Explicit Route Object (ERO), Record Route Object (RRO), Path State control Block (PSB) list and Reservation State control Block (RSB) list. Then, it creates and sends a *RSVP-TE RESV* message to router 3 (which is the tunnel pen ultimate router). The operation is repeated across the tunnel path until reaching the head-end router (router 5) where the establishment of the MPLS tunnel completes successfully. Flow H5-H4 is forwarded over the established MPLS tunnel path, which passes through routers 5-1-3-4 as shown in Fig. 5-4.

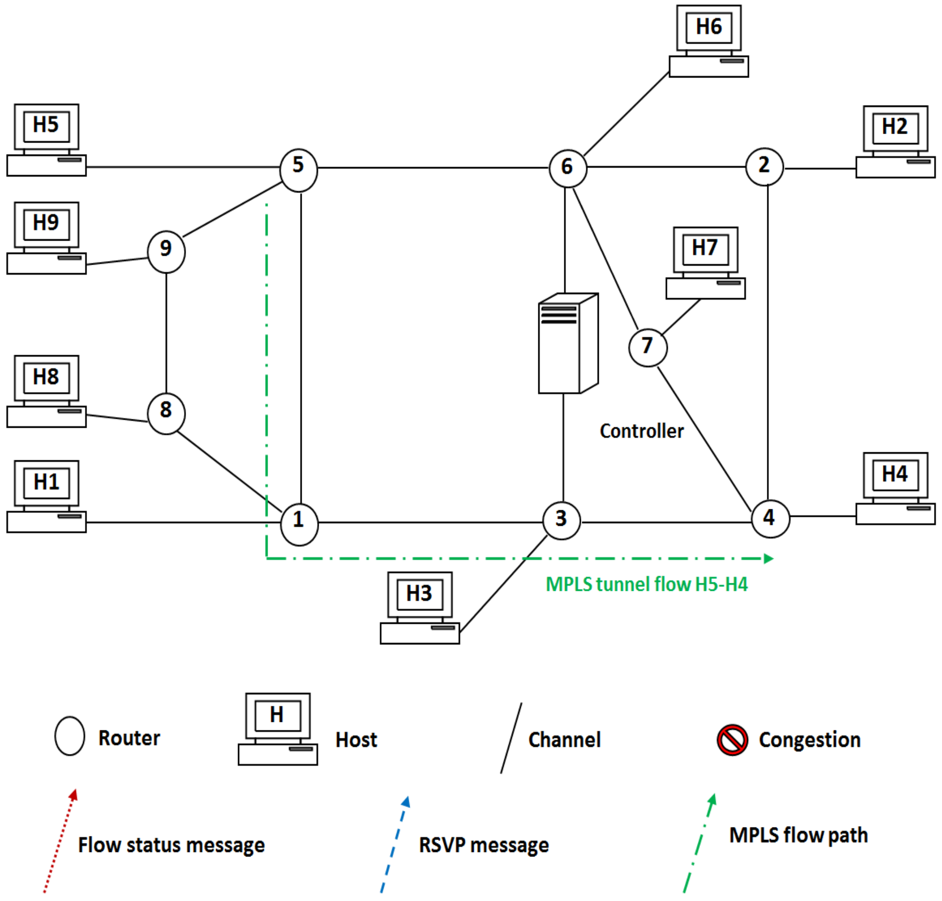


Fig. 5-4 The MPLS path of flow H5-H4

The period of time recorded to establish tunnel serving flow H5-H4 is 74.9 ms. The allocated bandwidth to the MPLS tunnel is equal to the data rate of the flow passing through it in addition to 5% of residual bandwidth *safety factor* mentioned in **section 4.11** of **chapter 4**.

5.11.2 Re-Adjusting the Reserved Bandwidth of the MPLS Tunnel

The data rate of any flow is changeable with time. Instantaneous and accurate re-adjusting of the bandwidth of the tunnel provides better bandwidth occupation. If the data rate of any applied flow across a tunnel is increased or decreased by 5% more or less than its previously measured value (more or less than the *safety factor* residual bandwidth), then the head-end router informs the controller via the *Flow status updating* message that the new bandwidth is required for the tunnel. Upon receiving the *Flow status updating* message, the controller applies its FDT to examine congestion on links.

If there is no congestion caused by the new applied data rate, then there is no need to change the path of the existing MPLS tunnel. The controller repeats the same procedure it used before in creating the tunnel. The new created tunnel will take the place of the old tunnel, which is removed from the path. The new tunnel is compatible with the old one in path, flow source and flow destination but different from it in other tunnel attributes and identifications like reserved bandwidth, tunnel ID, tunnel LSP ID and tunnel color.

Example, suppose that in the network depicted in Fig. 5-4 where host 5 and host 4 are connected via MPLS tunnel passes through path 5-1-3-4. The data rate of flow H5-H4 has risen moderately. Router 5 informs the controller about the flow status through the *Flow status updating* message. The controller applies its FDT upon the flow characteristics. It internally removes the old data rate of the flow and tests the congestion possibility upon the used path under the new data rate. The result is no problem in using the same path of the existing MPLS tunnel that passes through routers 5-1-3-4 with a higher reserved bandwidth.

The controller orders to establish a new MPLS tunnel by inserting its attributes and identifications in a *RSVP Final hop resv* message and sends the generated *RSVP Final hop resv* message to router 4 (which is the tunnel tail-end router) as shown in Fig. 5-5. The *RSVP Final hop resv* message contains the identifications of both the old tunnel and the new tunnel as shown in Table V-II.

TABLE V-II
IDENTIFICATIONS OF THE NEW TUNNEL THAT REPLACING THE EXISTING ONE

Tunnel Number	44
Tunnel ID	95
Tunnel LSP ID	145
Tunnel Color	195
Previous Tunnel ID	92

The *Developed RSVP* model of router 4 replaces the old information of the tunnel's attributes and identifications of the Explicit Route Object (ERO), Record Route Object (RRO), Path State control Block (PSB) list and Reservation State control Block (RSB) list with the new data. Router 4 sends *RSVP*-TE *RESV* message to router 3 (router 3 is the tunnel penultimate router).

The operation is repeated across the tunnel path until reaching the head-end router (router 5) where both the establishment of the new MPLS tunnel and the removal of the old MPLS tunnel complete successfully. The new tunnel takes the place of the old one but with higher reserved bandwidth. The involved routers continue forwarding flow H5-H4 of higher data rate via the same MPLS tunnel path, which passes through routers 5-1-3-4 and as shown in Fig. 5-5.

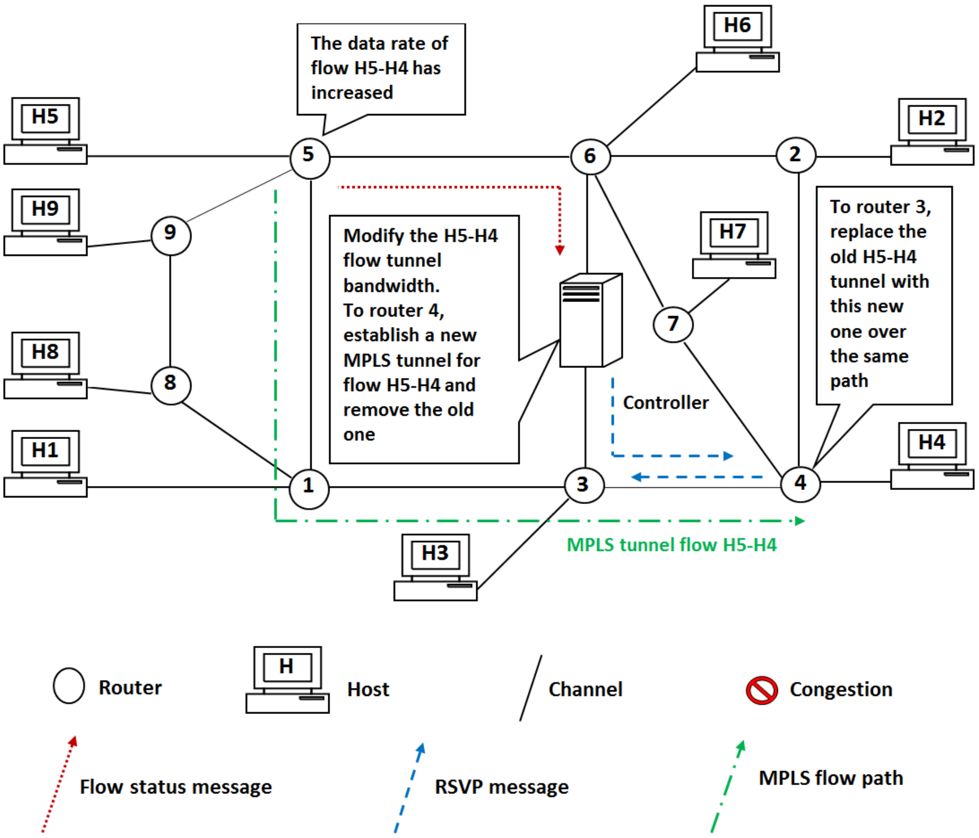

The data rate of flow H5-H4 has increased
H6
H5
H2
H9
H7
H8
H1
H4
H3
5
6
2
9
7
8
1
3
4
Modify the H5-H4 flow tunnel bandwidth. To router 4, establish a new MPLS tunnel for flow H5-H4 and remove the old one
To router 3, replace the old H5-H4 tunnel with this new one over the same path
Controller
MPLS tunnel flow H5-H4
H
Router
Host
Channel
Congestion
Flow status message
RSVP message
MPLS flow path

Fig. 5-5 Re-adjusting the reserved bandwidth of the MPLS tunnel by replacing the old tunnel with a new one of same path

The period of time recorded to re-adjust the bandwidth tunnel serving flow H5-H4 is 58.9 ms.

5.11.3 Removing the MPLS Tunnel

The sender host requests to remove the used MPLS tunnel; the head-end router conveys this request to the controller, which takes the following steps:

- The controller creates a *Remove tunnel* message, inserts tunnel's ID and path inside it and sends it to the tail-end router. The structure of the RSVP *Remove tunnel* message is shown in Fig. 5-6.
- According to the information received from the RSVP *Remove tunnel* message, the tail-end router receives and de-capsulates the arrived *Remove tunnel* message then sends it to its *Developed RSVP* model.
- The *Developed RSVP* model allocates the tunnel that the controller orders to remove. Then, it deletes all tunnel attributes and identifications from its database represented by the Explicit Route Object (ERO), Record Route Object (RRO), Path State control Block (PSB) list and Reservation State control Block (RSB).
- After the removal operation, the *Developed RSVP* model of the router creates another *Remove tunnel* message and sends it to the pen-ultimate router.
- This operation is repeated at each router across the MPLS path until reaching the head-end router where the MPLS tunnel is completely removed.
- At each router, an *Acknowledge* message supplied with *code* is sent back to the previous router to confirm the successful arrival of the *Remove tunnel* message.
- The head-end router sends another *Acknowledge* message to the controller to inform it that the tunnel has been removed. It also deletes the tunnel's identifications from the *Label Information Base* (*LIB*) table container model.
- After removing the tunnel completely, routers continue forwarding flow's *Data* packets according to the IP-OSPF routing table. The controller can re-use some of the tunnel's identifications like: Tunnel Number, Tunnel ID, Tunnel LSP ID and Tunnel Colour in establishing another tunnel.

MPLS Path: N_1 N_2 N_3 N_4 ……… N_k	**Code**
Tunnel Identifications: Tunnel Number, Tunnel ID, Tunnel LSP ID, Tunnel Colour	

Fig. 5-6 The structure of RSVP Remove tunnel message

N_1 N_2 N_3 N_4 ……… N_k : Addresses of routers where the tunnels' path passes through.

N_1: The address of the head-end router.

N_k: The address of the tail-end router.

Tunnel Identifications**:** Tunnel Number, Tunnel ID, Tunnel LSP ID, Tunnel Colour.

Example, in the network depicted in Fig. 5-4, suppose host H5 wants to remove the existing tunnel to host H4. Host H5 implies its request inside the flow's *Data* packets. Router 5 receives the request then sends a *Flow status updating* message to the controller. The controller sends *Remove tunnel* message to router 4 (the tunnel tail-end router). Router 4 identifies the tunnel that the controller orders to remove, which is the tunnel that serves flow H5-H4. It deletes all the identifications and the attributes of the tunnel from its database completely. It performs this operation at its *Developed RSVP* model.

Router 4 continues removing the old MPLS tunnel that passes over routers 5-1-3-4 through sending *Remove tunnel* message to router 3, which repeats the same scenario. This operation is repeated until reaching the head-end router (router 5), which sends an *Acknowledge* message to the controller to inform it that the tunnel has been removed completely. The illustrated operation is depicted in Fig. 5-7.

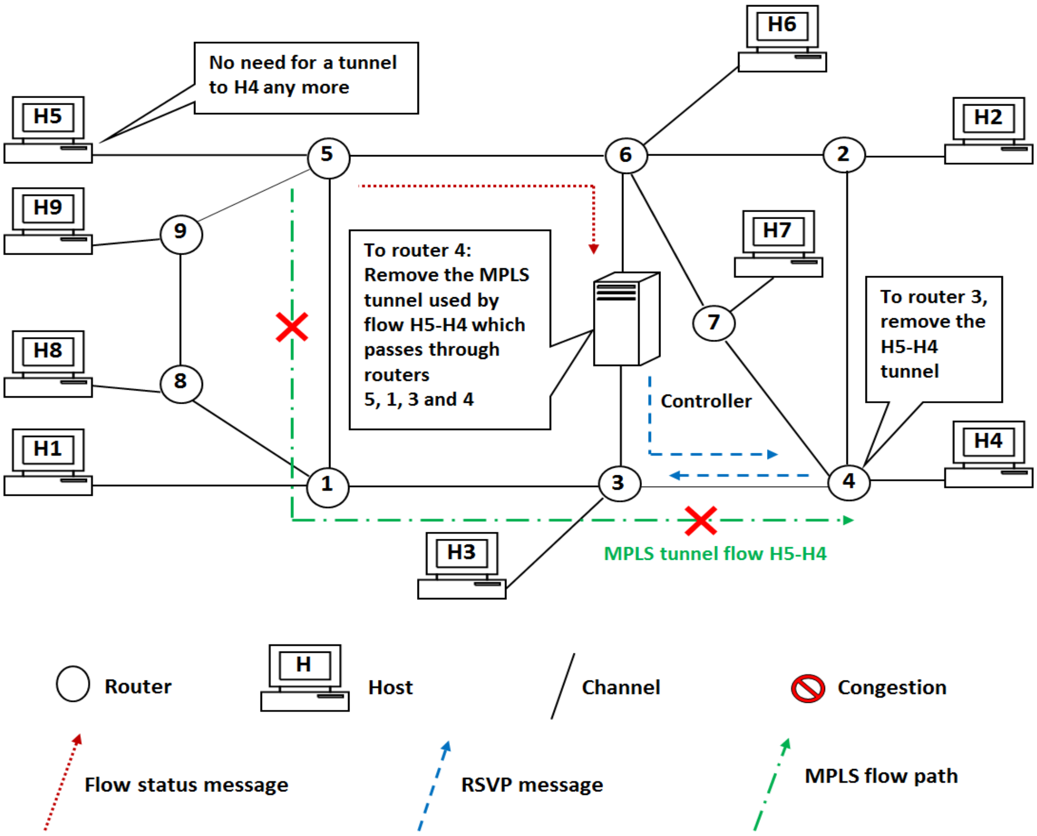

No need for a tunnel to H4 any more
H6
H5
H2
5
6
2
H9
9
H7
To router 4: Remove the MPLS tunnel used by flow H5-H4 which passes through routers 5, 1, 3 and 4
To router 3, remove the H5-H4 tunnel
7
H8
8
Controller
H4
H1
1
3
4
H3
MPLS tunnel flow H5-H4
Router
H
Host
Channel
Congestion
Flow status message
RSVP message
MPLS flow path

Fig. 5-7 The process of removing of existing MPLS tunnel

5.11.4 Changing the Path of the MPLS Tunnel

Using the same path to serve a tunnel may not provide the optimal flows distribution. Sometimes, the requirements impose changing the path of the tunnel to avoid congestion or to achieve better usage to the available resources. Changing the path of the tunnel involves removing the old existing tunnel and establishing a new one of different path. Changing the path of a MPLS tunnel passes through the following procedure:

If the data rate of an applied flow across a tunnel upsurges extremely, the head-end router informs the controller through a *Flow status updating* message about the new bandwidth required for the tunnel. Upon receiving the *Flow status updating* message, the controller applies its FDT which notices congestion begins occurring on the path. The controller applies FDA to find an alternative path instead of the existing one to avoid the congestion. By using both scenarios of creation and removal of MPLS tunnel mentioned in **sections 5.11.1** and **5.11.3** illustrated before, the controller creates the new tunnel over the new suggested path (where there is no expected congestion) and at the same time, it removes the existing old tunnel. The flow commences using the new established tunnel.

Example, suppose that in the network depicted in Fig. 5-4, the data rate of flow H5-H4 has risen extremely. Router 5 informs the controller about the flow status through a *Flow status updating* message as usual. The controller applies its FDT upon the flow's characteristics. The result of the FDT shows that there is congestion over link 3-4. The controller applies FDA to find an alternative path to prevent the congestion. The new proposed path of the MPLS tunnel (specified by the FDA and serves flow H5-H4) should pass through routers 5-6-2-4 instead of the old MPLS tunnel path which passes through routers 5-1-3-4.

The new path represents an alternative path from the OSPF path due to the congestion that occurred via the latter one. The controller orders to establish the new MPLS tunnel by inserting its attributes, identifications and path in a *RSVP Final hop resv* message and to remove the old MPLS tunnel by inserting its identifications and path in a *RSVP Remove tunnel* message. The controller sends both *RSVP Final hop resv* and *RSVP Remove tunnel* messages to router 4 (the tail-end router). Router 4 repeats the scenario of creating new MPLS tunnel over routers 5-6-2-4 through sending *RSVP TE RESV* message to router 2 and so on. It also begins removing the old MPLS tunnel that passes over routers 5-1-3-4 through sending *RSVP Remove tunnel* message to router 3 and so on. Both the creation and removal operations continue until reaching the head-end router (router 5). The illustrated operation is depicted in Fig. 5-8.

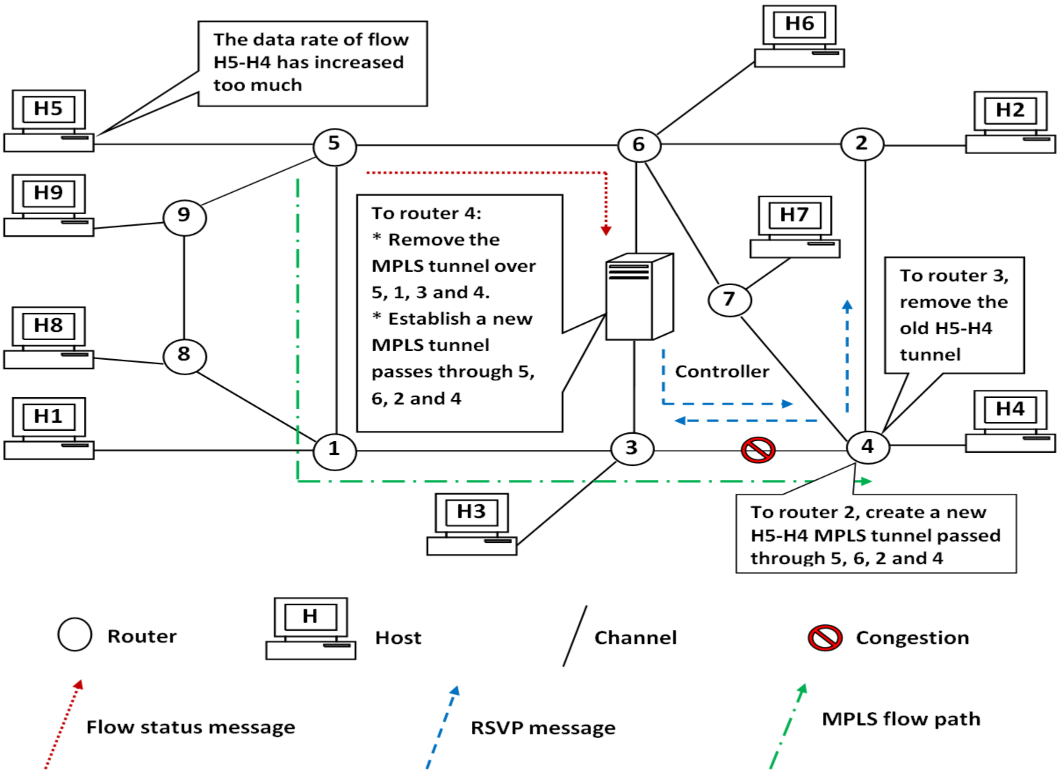

The data rate of flow H5-H4 has increased too much
H6
H5
H2
5
6
2
H9
9
To router 4:
* Remove the MPLS tunnel over 5, 1, 3 and 4.
* Establish a new MPLS tunnel passes through 5, 6, 2 and 4
H7
7
To router 3, remove the old H5-H4 tunnel
H8
8
Controller
H4
H1
1
3
4
H3
To router 2, create a new H5-H4 MPLS tunnel passed through 5, 6, 2 and 4
Router
H
Host
Channel
Congestion
Flow status message
RSVP message
MPLS flow path

Fig. 5-8 The process of changing the path of MPLS tunnel

Flow H5-H4 begins using the new alternative tunnel of path passing through routers 5-6-2-4 as show in Fig. 5-9. The period of time recorded to establish the new tunnel of the alternative path is 83 ms.

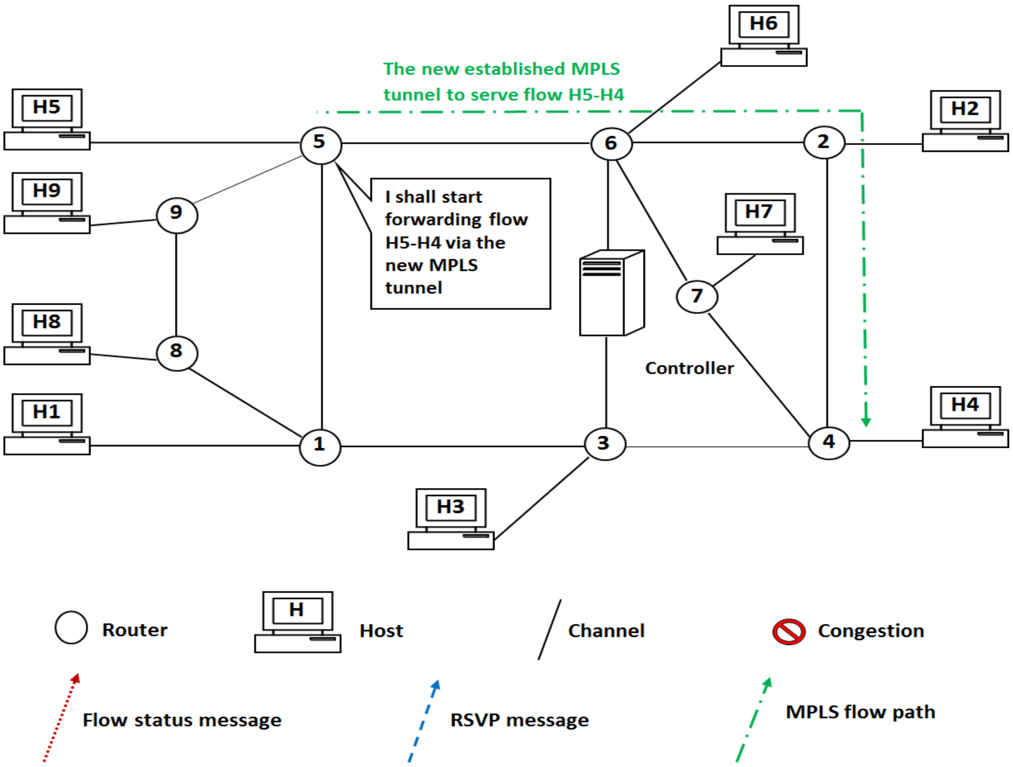

H6
The new established MPLS tunnel to serve flow H5-H4
H5
H2
5
6
2
H9
9
I shall start forwarding flow H5-H4 via the new MPLS tunnel
H7
7
H8
8
Controller
H4
H1
1
3
4
H3
Router
H
Host
Channel
Congestion
Flow status message
RSVP message
MPLS flow path

Fig. 5-9 The H5-H4 flow passes over the detour MPLS tunnel path 5-6-2-4

5.12 Results of 9 Routers Network

The results shown in Fig. 5-10 represent the tunnel creation time duration for flow H5-H4 of the 9 routers' network during three different periods of simulation time that meets three different applied load values. The first tunnel creation time value was recorded when all flows began together. At that time, the controller was busy with many mathematical operations in calculating the paths related to the flows as it was receiving many status messages that included many requests to create tunnels. The second creation time value of tunnel was recorded during moderate applied load over the network, while the third value was recorded during high applied load.

The tunnel creation time of flow H5-H4 and the network applied load status are also shown in Table V-III.

TABLE V-III
TUNNEL CREATION AND REPLACEMENT TIME OF 9 ROUTERS NETWORK

Time duration of tunnel creation	**Applied load status**	**Controller action**
74.9 ms	Start	Tunnel first creation (via main path)
58.9 ms	Moderate	Tunnel bandwidth updating (via main path)
83 ms	High	Creating a tunnel of an alternative path (restoration by a detour path)

It can be noticed that tunnel's creation time and restoration time for all cases is a small value compared to the requirements of modern communications. End to end delay (Latency) for flow H5-H4 is also an acceptable value.

The total simulation time of the network of 9 routers lasts for one hour and several minutes.

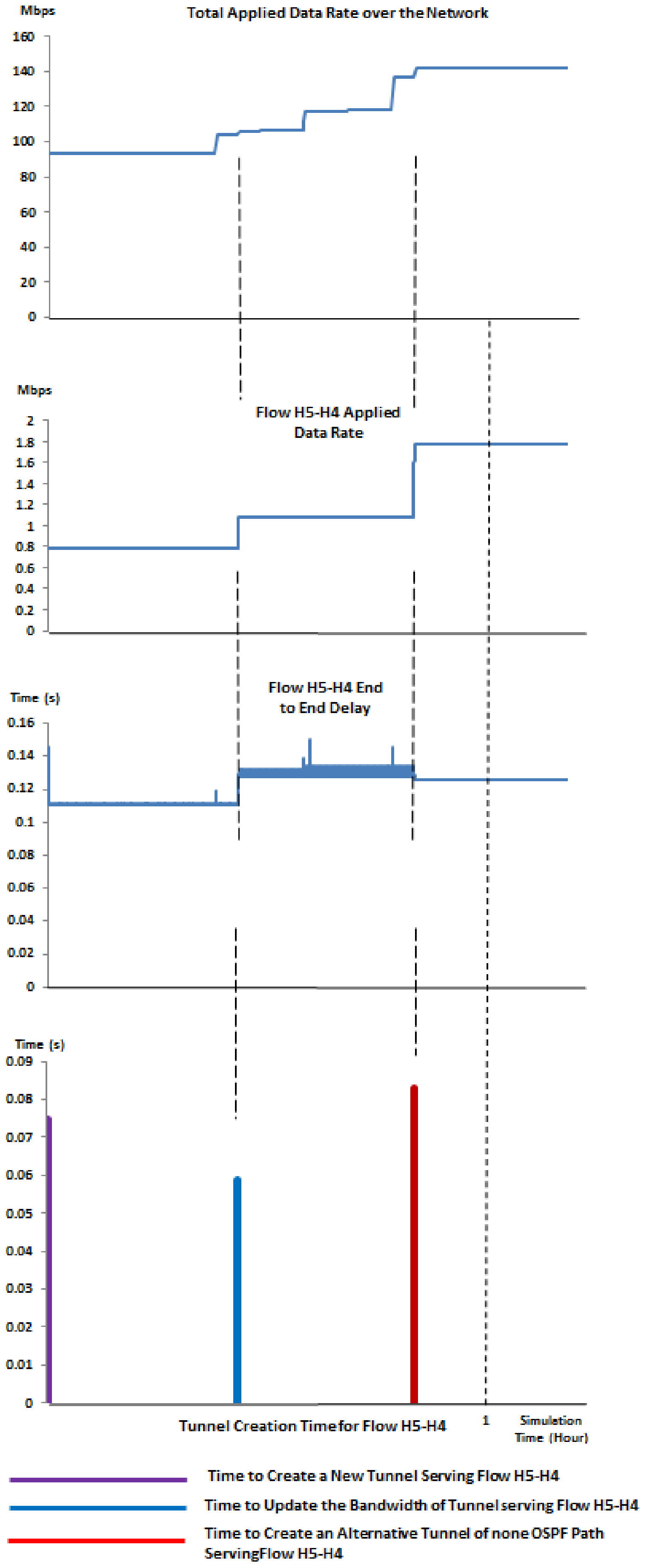


Fig. 5-10 End to end delay and tunnel creation time for flow H5-H4 along with its applied data rate and the total applied data rate of 9 routers network

5.13 More Complicated Topologies

Testing our methodology on larger networks of more complicated topologies confirms the validity of our proposal. The networks of bigger topologies are:

- 13 Routers Network.
- 18 Routers Network.

5.13.1 13 Routers Network

The network of Fig. 5-11 represents a network consists of 13 routers and each router is connected to a host that has the same number. Each host contacts one or more of the other hosts by applying flows of changeable values of data rates during simulation time.

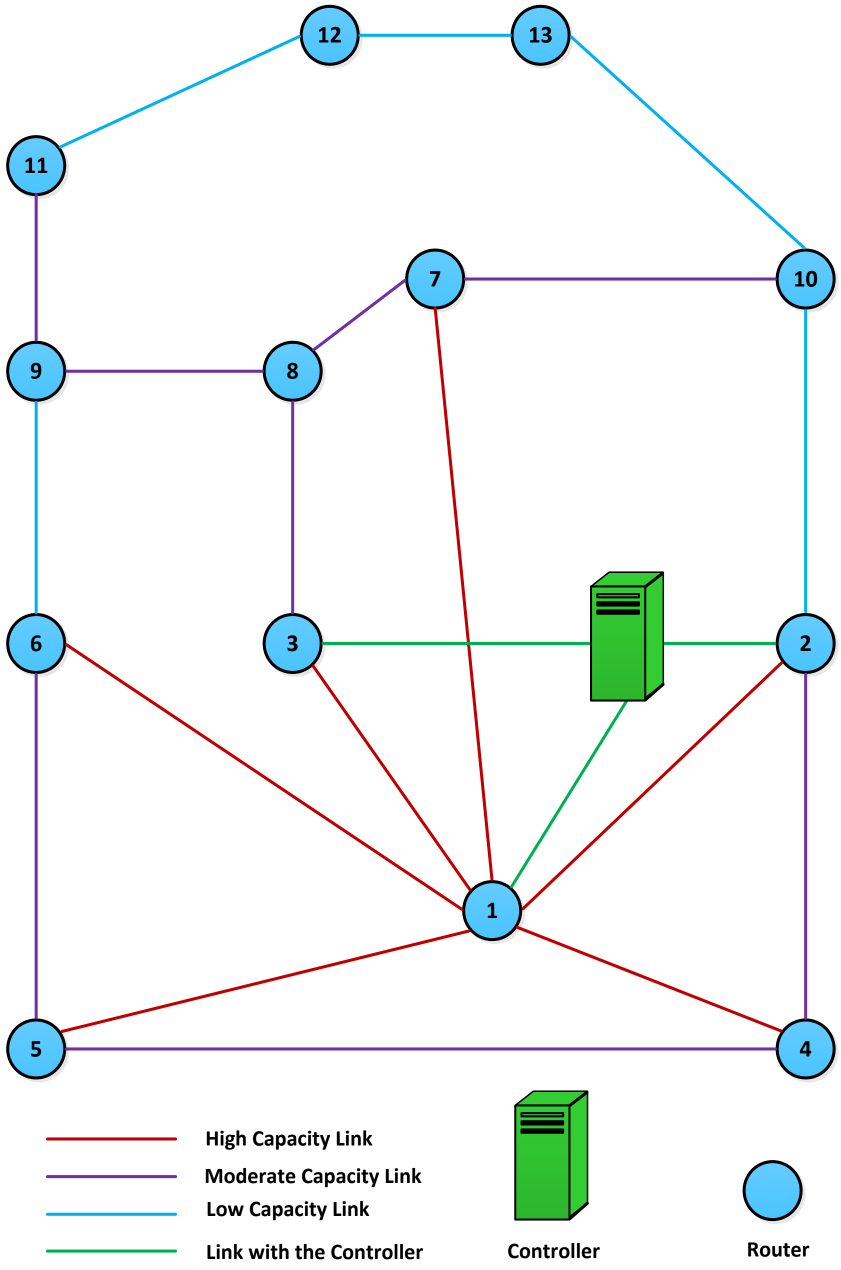


Fig. 5-11 Network of 13 routers

For the first hour of the simulation and during its first 15 minutes, the hosts subsequently commence applying load to the network. The controller establishes the MPLS tunnels from the head-end routers to the tail-end routers as required by applying its FDT.

During the next 15 minutes of the simulation, some hosts increase their applied data rates to higher values but still under a tolerable amount. The controller applies its FDT to establish new tunnels with higher bandwidth passing via the same used paths.

During the third 15 minutes of the simulation, some hosts increase their applied data rates to high values. The controller applied its FDT and FDA to establish new tunnels with higher bandwidth. Some of new tunnels take the same paths of their ancestors and others take different paths to avoid congestion.

During the last 15 minutes of the simulation, the hosts which were sending high data rates reduce their applied load to moderate values. The controller returns the diverted flows to their original paths.

Finally and during the last few minutes after the first hour, hosts continue applying the same moderate data rate values where there is no need to establish or change the path of any MPLS tunnel. The total simulation time of the network of 13 routers lasts for one hour and several minutes.

The results shown in Fig. 5-12 represent the durations of time recorded to create or restore (change of path) of all the tunnels in the simulation of 13 routers network of Fig. 5-11. The time consumed to establish new tunnels during the first 15 minutes of the simulation is shown in the violet lines. The time consumed to update the bandwidth of the existing tunnels with other tunnels passing through the same paths during the second 15 minutes is shown in the blue lines and the time consumed to update the bandwidth of the existing tunnels by replacing them with alternative tunnels of different paths during the third 15 minutes is shown in the red lines.

The longest value of time that was recorded to create the longest replacement (detour) in terms of the number of hosts for the MPLS tunnel of path that passes through routers (*4 – 2 – 10 – 13 – 12 – 11*) under high applied load equals to 178 ms.

The longest value of time that was recorded under the highest applied load to create a replacement (detour) for the MPLS tunnel path that passes through routers (*10 – 13 - 12 - 11 – 9),* equals to 173 ms.

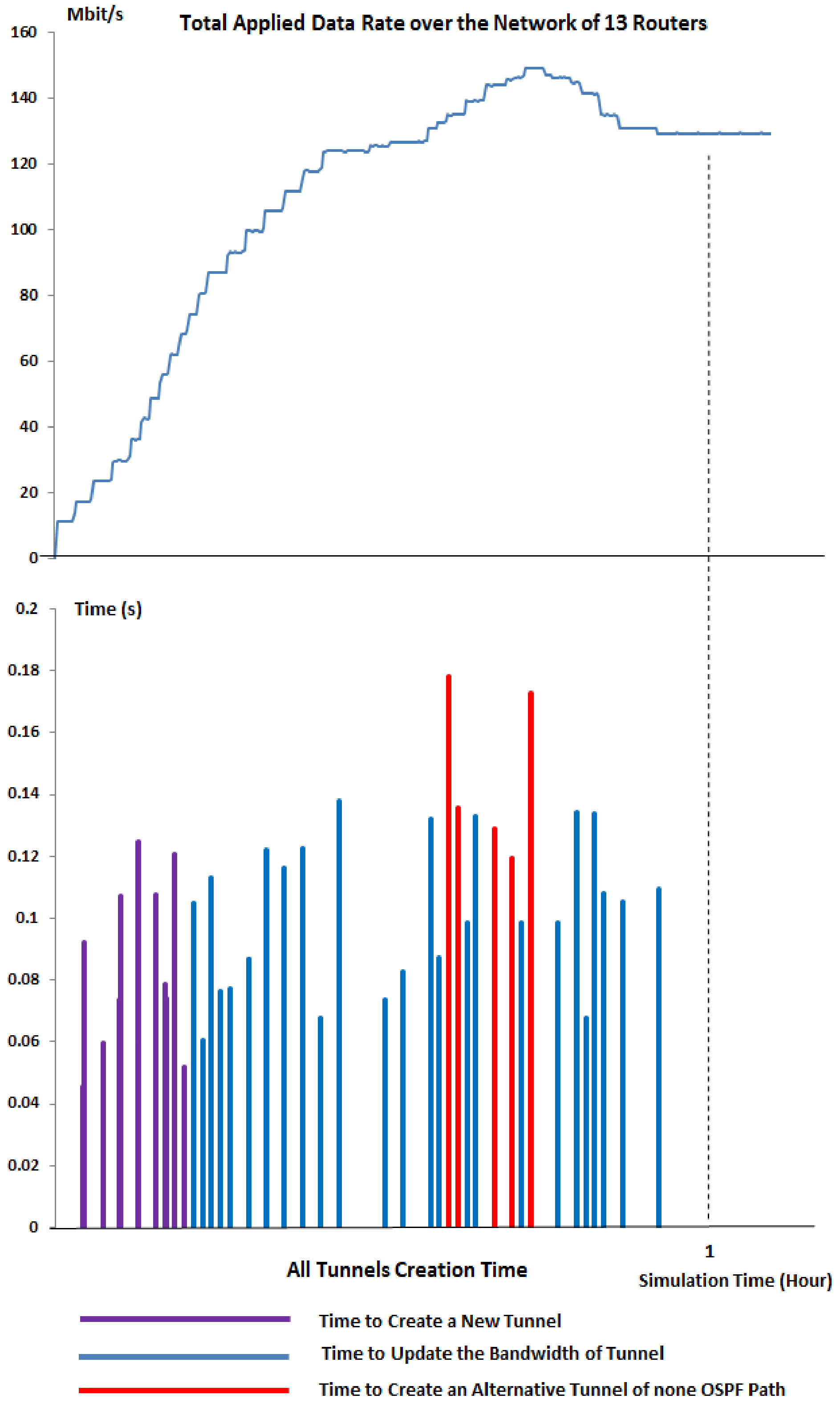


Fig. 5-12 Durations of time to create or restore tunnels in 13 routers network

Some of the created tunnels of 13 routers network and their information of Fig. 5-12 are also shown in Table V-IV.

TABLE V-IV
Some of the Created Primary Tunnels in 13 Routers Network

Flow Source (Host)	**Flow Destination (Host)**	**Primary MPLS tunnel Path (Routers)**	**Time duration of tunnel creation**	**Applied load status**	**Controller action**
1	*10*	*1 – 2 – 10*	35 ms	Low	First establishment
1	*8*	*1 – 3 – 8*	43 ms	Moderate	First establishment
6	*10*	*6 – 1 – 2 – 10*	73.5 ms	Low	First establishment
6	*10*	*6 – 1 – 2 – 10*	78 ms	Moderate	Bandwidth update
6	*10*	*6 – 1 – 2 – 10*	87 ms	High	Bandwidth update
11	*7*	*11 – 12 – 13 - 10 - 7*	137.8 ms	Moderate	Bandwidth update
13	*5*	*13 - 10-2 – 1 - 5*	104 ms	Moderate	Bandwidth update
10	*9*	*10 – 7 - 8 - 9*	118.5 ms	Moderate	First establishment
10	*9*	*10 – 13 - 12 - 11 – 9*	173 ms	High	Replacement with detour
10	*9*	*10 – 7 - 8 - 9*	135.6 ms	Moderate	Replacement with original (OSPF)
12	*4*	*12 – 13 – 10 - 2 – 4*	125 ms	Low	First establishment
12	*4*	*12 – 13 – 10 - 2 – 4*	140.2 ms	Moderate	Bandwidth update
4	*11*	*4 – 1 – 6 – 9 – 11*	92.3 ms	Low	First establishment
4	*11*	*4 – 2 –10 – 13 – 12 – 11*	178 ms	High	Replacement with detour
4	*11*	*4 – 1 – 6 – 9 – 11*	100.5	Moderate	Replacement with original (OSPF)

5.13.2 18 Routers Network

The network of Fig. 5-13 represents a network that consists of 18 routers. Each router is connected to a host that has the same router's number. After building the routing tables, the routers inform the controller about the network's topology, resources and their routing tables as usual. Then, the hosts of the network commence sending *Data* packets to some of their counterparts. At the beginning, the controller establishes many tunnels within a short period. It uses its FDT to allocate the paths and the messaging system to establish the tunnels as usual.

During simulation time, some hosts increase the data rates of the flows they send. The controller responds to those requirements by updating bandwidth of the existing tunnels that exceeds the *safety factor* value of 5 percent from the previous value. It continues establishing alternative tunnels via the same paths of their ancestors until reaching the point that some of the utilized paths cannot handle the current data rates of their occupying flows without congestion. At this point, the controller activates its FDA to establish alternative tunnels of alternative paths (some are longer paths).

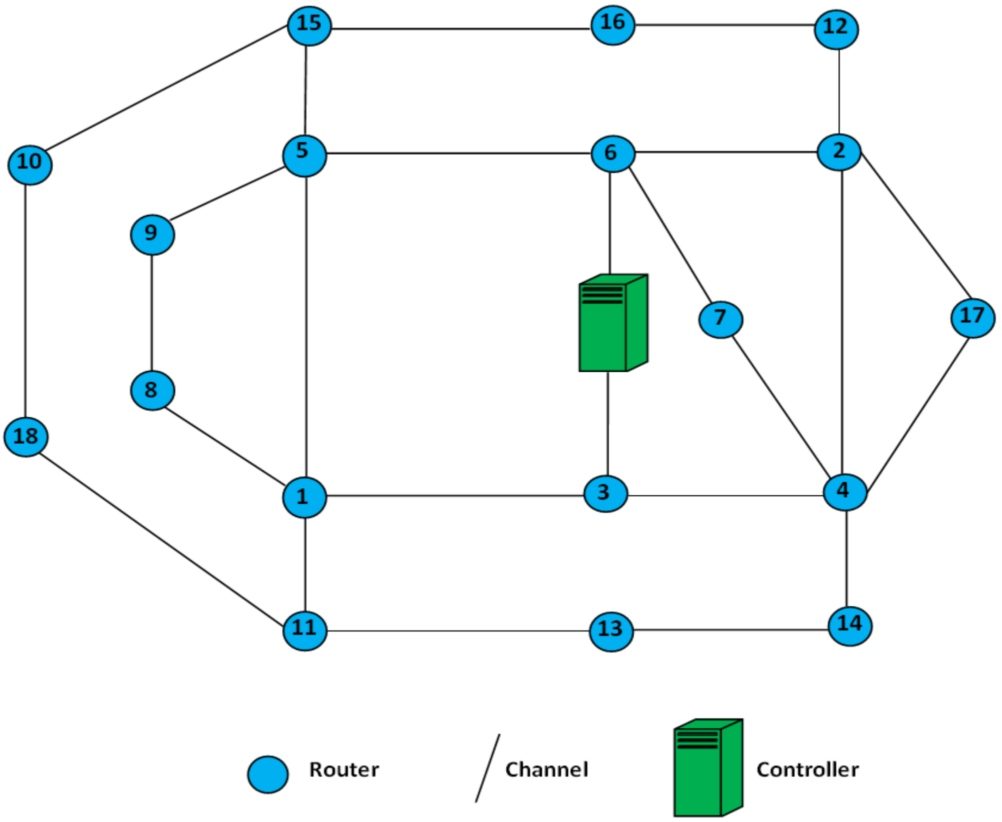


Fig. 5-13 Topology of 18 routers network

The total applied load and the time consumed to establish the related tunnels of the 18 routers network of Fig. 5-13 are as shown in Fig. 5-14. Some of the created tunnels at the network of 18 routers are also shown in Table V-V.

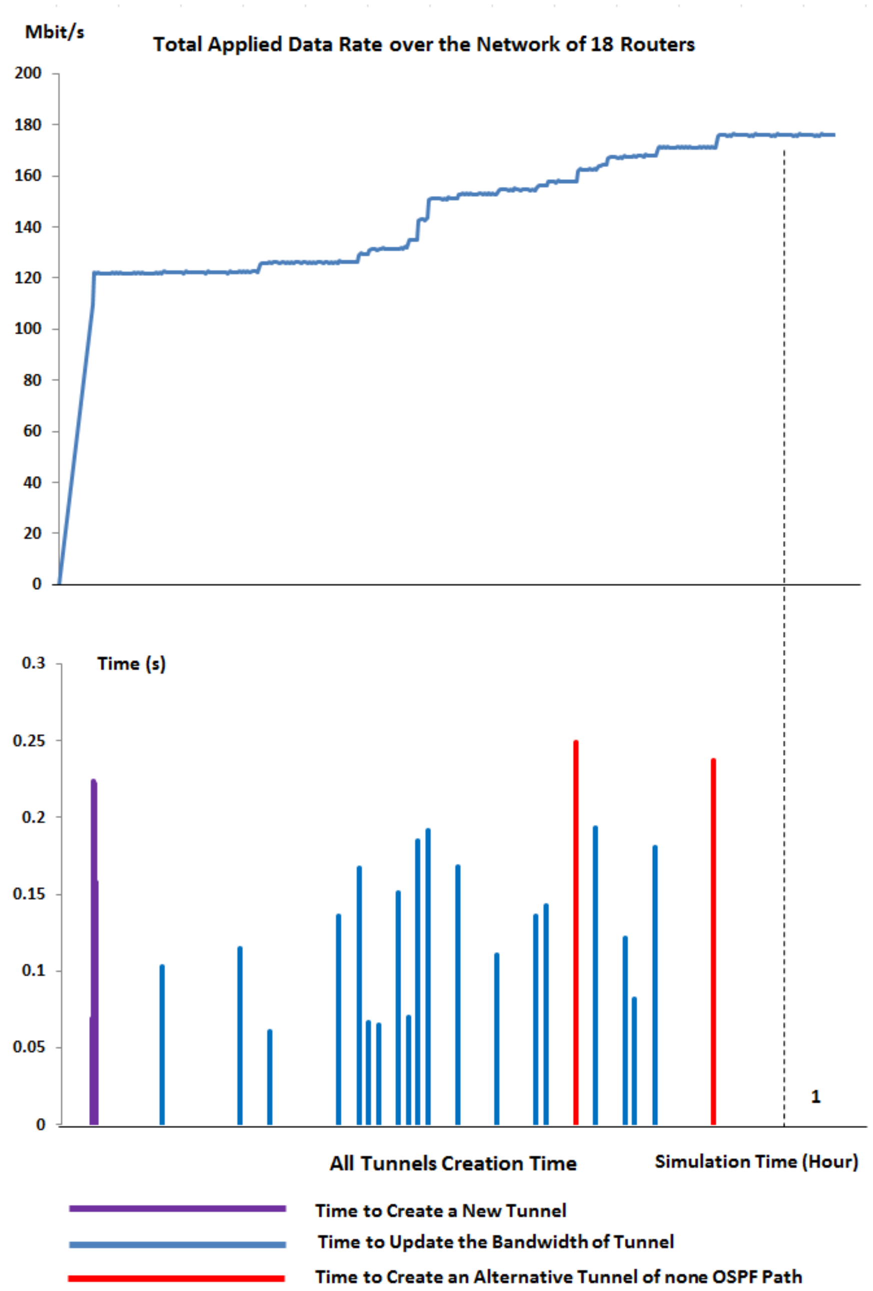


Fig. 5-14 Durations of time to create or restore tunnels in network of 18 routers

TABLE V-V
SOME OF THE CREATED TUNNELS IN 18 ROUTERS NETWORK

Flow Source (Host)	**Flow Destination (Host)**	**MPLS tunnel Path (Routers)**	**Time duration of tunnel creation**	**Applied load status**	**Controller action**
H 1	*H 2*	*1-3-4-2*	75.6 ms	Start	First establishment
H 1	*H 2*	*1-3-4-2*	69.6 ms	Moderate	Bandwidth update
H 9	*H12*	*9-5-6-2-12*	112.2 ms	Moderate	Bandwidth update
H 10	*H 17*	*10-15-16-12-2-17*	205.3 ms	Start	First establishment
H 10	*H 17*	*10-18-11-13-14-4-17*	244 ms	High	Replacement with detour

5.14 Summary of Chapter 5

- The controller under this approach establishes the primary MPLS tunnel, re-adjust its bandwidth instantaneously and accurately, change its path during very short time or remove the MPLS tunnel. The dynamic establishment and management of the MPLS tunnel through using the controller provided better usage of the available bandwidth. By fast updating the tunnel requirements, the flow that traverses the tunnel suffers less from outage probability. As the controller behaves proactively and it knows the flows' distribution and the residual bandwidth over all network links, tunnel creation time and tunnel restoration time are reduced, as tunnel creation operation is easier and there is no possibility of *PATH Error* messages to exist.
- Tunnel creation time is affected by several factors like hops number, applied load value, applied load pattern, the degree of the complexity of the topology and the distance away from the controller. The network takes a longer time to create or change the path of the MPLS tunnel, if the new path passes via more number of hops or there is high applied load over the network. However, it is still much less than the value declared in [9] which range between 2-3 seconds. If the controller tries to establish several tunnels at the same time, this may cause a small extra delay as well.
- The result is a flexible MPLS tunnel in the terms of path and reserved bandwidth. The path of the established tunnel that serves the same *source-destination* pair may pass completely or partially through the main OSPF routing table path or pass through very different path. The tunnel has a global *identification numbers* as there is no possibility that two or more tunnels have the same *identification numbers*.
- The location of the controller affects the tunnel creation time. Placing the controller's position at the middle of the network makes it closer to the largest number of routers and enables it to create and restore the largest number of tunnels with less processing time.
- The controller of our project recognizes the location and the amount of the congestion when it receives the *Flow status updating* messages and applies its FDT. This makes our system faster in dealing with the congestion problem than the other systems which wait for the congestion to occur then re-act after receiving feedback messages informing the congestion problem.

Chapter 6

Soothing the Effect of Link Failure by Adopting the SDN Notion

Introduction to Chapter 6

This chapter deals with the link failure in both OSPF/IP and OSPF/MPLS routed networks. The following sections show the link failure problem and its destructive effect on the network performance. The later sections show the proposed solutions and the efforts of other researchers on dealing with it. After that, it shows a scenario of achieving a successful link failure restored with network recovery in an OSPF/IP routed network implemented by OMNeT++ simulator through its Inet-OMNeT++ project. Then, we explain our methodology in dealing with the link failure in OSPF/IP network by viewing the ideas, implementing them and getting the results. Finally, we explain our methodology in dealing with link failure in OSPF/MPLS network by proposing our algorithm, implementing it and getting the results, which is the most important part of this chapter. This chapter represents the third contribution of our research.

6.1 Network Failure

Network failure is represented by link failure or node failure. Link failure and node failure are some of the problems that cause degradation to network performance. Network failure duration can extend from seconds to weeks. Some reasons behind failures are hardware malfunctions, accidental cable cuts, natural disasters (example, fires), software errors, and human error (example, incorrect maintenance) [93].

The influence degree of the network failure on network performance depends on factors some of which are the failure location, network topology, and the congestion control [93]. When a link or node failure event happens in a routed network, there is unavoidable period of disruption to the delivery of traffic. This period extends until the network re-converges on the new topology. Packets passing through the failed component to their destinations may be dropped or may suffer looping which negatively affects the QoS [94].

The probability of a router complete breakdown is less than the probability of the failure of some of its components. Thus, this chapter discusses link failure only. During the event of link failure, the packet delivery disruption period caused by a conventional routing transition is affected by [94] several factors:

- The time taken by a router to detect the failure and react to it.
- The time taken to inform the other routers in the network about the failure.
- The time taken to re-calculate the forwarding tables. (Few milliseconds for a link state protocol using Dijkstra’s algorithm).
- The forwarding hardware also consumes time to load the revised forwarding tables.

6.2 Research Problems

The link failure is a problem by itself. Dealing with this problem creates specific challenges. Some of those challenges are related to the type of the network that suffers from the link failure.

6.2.1 Link Failure in OSPF/IP Routed Networks

The link failure causes a change in the topology of the network. This change affects the routing tables in all routers. OSPF is a fast converging routing protocol. [3] pp 238 and pp 258. Routing tables over the network reach their fully calculated stable state by recording the change in the network topology in a Link State Advertisement (LSA) and flooding it to all other routers in the routing domain. Upon receiving the updated link state database, routers recalculate their routing tables. Only one re-calculation of each router's routing table achieves the convergence process [3] pp 258.

OSPF uses the Dijkstra algorithm to build and update the routing table [14] pp 444 where the available bandwidth is the main factor that is used to calculate the best path [95] regardless the flows' distribution and their data rates. As the routing operation is completely determined by the weights of the links [96], this raises a problem of congestion due to the reduction of resources resulting from the failure.

6.2.2 Link Failure in MPLS Networks

The link failure event affects the OSPF/MPLS routed network worse than it does with the OSPF/IP routed networks. The time consumed to restore a MPLS tunnel is longer than the time consumed to update routing tables in conventional link state IP routed networks. Under optimum conditions, the alternative tunnel establishment time cannot take less than 2–3 seconds [9]. This outage period is considered a problem in the contemporary networks especially when dealing with sensitive real time applications.

Network restoration represented by fast re-routing gives the solution to this problem however it has several limitations and problems. Next section (**section 6.2.3**) discusses the Fast Reroute (FRR) mechanism and the later section (**section 6.2.4**) illustrates the problems facing this mechanism.

6.2.3 Fast Re-Route (FRR) Theory

Fast Reroute (FRR) is a mechanism used to protect MPLS traffic engineering (TE) LSPs from link and node failures [52]. In the event of a link failure, FRR establishes a procedure that provides rerouting around a failed link. The LSP is routed to the next-hop using a pre-configured backup tunnel. This is called Next-Hop (NHOP) backup tunnels. Fig. 6-1 shows the NHOP backup tunnels in a fictional network.

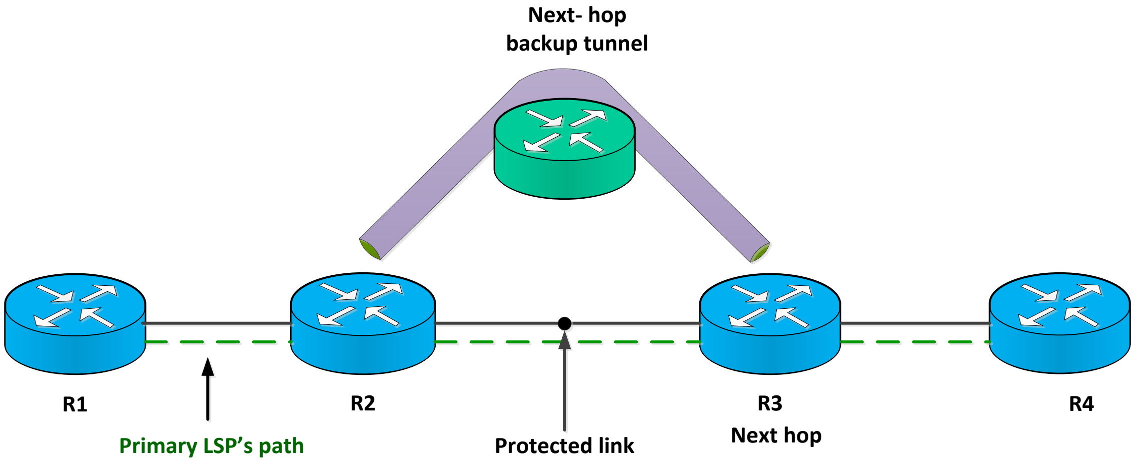


Fig. 6-1 NHOP backup tunnel [52]

The node where the backup LSP path starts is called Point of Local Repair (PLR), while the node where the backup LSP path re-joins the original LSP path is called Merge Point (MP) [97]. In Fig. 6-1, router R2 represents the PLR and router R3 represents the MP.

When the PLR detects a failure on the primary LSP, it must immediately switch packets to the backup LSP instead of carrying on forwarding via the primary LSP. This operation is done within 10s of milliseconds [97]. Subsequently, the PLR signals the outage to the head-end router node (the ingress LSR) to inform it about the failure event, and allow for the signaling of a new optimized LSP that is routed over the remaining TE topology [98]. In Fig. 6-1, router R1 represents the ingress LSR.

The backup tunnel should be designed to avoid congestion, loops and traffic overlap. Traffic overlap means that the packets of the same IP flows pass via the same unidirectional link more than one time [99].

A backup tunnel can protect one LSP or multiple LSPs. Also, a backup tunnel can protect multiple interfaces. This is called many-to-one (N:1) protection which has more advantage than one-to-one (1:1) protection [100].

6.2.4 Problems of Backup Tunnels

The main problems of the backup tunnels can be illustrated as:

- *Bandwidth reservation:*

  Although the backup tunnels do not consume bandwidth when no failure occurs, the network should reserve enough restoration bandwidth to ensure the LSP restoration during the event any failure. Without efficient bandwidth sharing among backup LSPs for different service primary LSPs, the network may reserve much more bandwidth on its links than is necessary [101]. This is considered a waste in network resources.

- *Validity:*

  The backup path may not be available during the failure event or it cannot bypass the failed area.

- *Long term utilization:*

  The backup path is used to forward data of the main tunnel path when failure happens. However, the backup path may not be the most efficient path to compensate the failed link during the whole simulation period.

6.2.5 Bandwidth Sharing among Backup Tunnels

According to [102, 103], the capacity in the backup path can be shared in two ways: *Inter-demand sharing* and *intra-demand sharing*. The *inter-demand sharing* refers to sharing of the backup bandwidths belonging to different demands (different primary LSPs) whose primary paths are linking disjoint. The *intra-demand sharing* means that backup capacity is shared on the link that used to protect the same demand (the same primary LSP).

6.3 Research Objectives

The aim of this chapter is to sooth the destructive effect of link failure on OSPF/IP routed networks and OSPF/MPLS routed networks. In OSPF/IP routed networks, this chapter employs the controller capabilities to deal with congestion resulting from lack of network resources as a consequence of link failure. This chapter discusses using the controller capabilities in OSPF/MPLS routed networks that apply network restoration before the link failure event. Applying those implementations involves the existence of a messaging system between the controller and the routers.

6.4 Controller Behavior Regarding the Link Failure Event

The link failure is an unpredictable event. This chapter uses methodology similar in some aspects to that of SDN to deal with link failure. The contribution of this chapter is represented by involving the controller to soothe the effect of link failure as much as possible. The contribution goes under two different scenarios:

6.4.1 Soothing the Effect of Link Failure in OSPF/IP Routed Network

Involving the controller to provide a new routing table to each router after link failure is not much feasible as the recent advanced routers can update their routing tables in less than one second [94]. Therefore, the controller is only involved with congestion testing after link failure recovery in terms of IP routed packets. It applies FDT and FDA that were presented in **chapter 4** on the new topology. According to the results of FDA and FDT, the controller orders specific routers to re-distribute some flows via alternative paths to avoid the congestion. This solves the problem viewed in **section 6.2.1**.

6.4.2 Soothing the Effect of Link Failure in OSPF/MPLS Routed Network

The controller builds dynamic primary MPLS tunnels as illustrated in **chapter 5** and in this approach it protects them by establishing dynamic backup tunnels. The controller aims to protect all links over the main MPLS path (if possible). The protection operation requires the availability of resources. The problems discussed in **section 6.2.4** (Problems of Backup Tunnels) are solved as follows:

- The controller as central manager calculates the dynamic backup tunnels taking in consideration the bandwidth sharing issue in an efficient way. It uses a specific algorithm to calculate the backup paths.
- The controller establishes two backup tunnels (if available) to protect each link across the path of the primary MPLS tunnel. If one backup tunnel fails or it cannot bypass the failed area, the other compensates it. The routers of the main MPLS path apply a specific scenario to check the validity of the backup tunnels. If the topology of the network does not provide enough resources for local link protection, the network applies a scenario of protecting the current link from a prior hop.
- After the event of link failure, the controller may adopt the path of the backup tunnel as primary path to the new established tunnel or it may choose another path. Therefore, the backup paths are considered temporary.

All those improvements provide our network with a global resource optimization.

6.5 Development Efforts to Minimize the Link Failure Effect

Many studies have been conducted to minimize the effect of link failure in both OSPF/IP networks and OSPF/MPLS networks. Reducing the effect of link failure in MPLS networks concentrates on developing the performance of the FRR.

6.5.1 Minimizing the Effect of Link Failure in OSPF/IP Routed Network

The Preventive Start-time Optimization (PSO) scheme, which identifies a convenient set of OSPF link weights at the start time and has the ability to handle any link failure scenario preventively, is proposed by Kamrul et al in [70]. This PSO is proposed to deal with the weakness of the Start-time Optimization (SO) and the Run-time Optimization (RO) schemes. The set of link weights defined by PSO minimizes the worst-case network congestion ratio for all probable link failure scenarios. The approach illustrated in [104] proposes the Preventive Start-time link-weight Optimization scheme along with link reinforcement considering all possible single-link failure scenarios.

The bi-criteria optimization model, which simultaneously minimizes the congestion of the normal state and that of the failure states, is proposed in [105]. It proposes an approach that uses an artificial objective function which when embedded into a local search algorithm; the function leads the search towards pareto-optimal solutions. The proposed algorithm is used to show whether it is possible to significantly reduce the congestion of the failure states, by allowing a slightly higher congestion in the normal state. These solutions are used for analysing the trade-off between the congestion of the normal state and that of the failure states. If the range of this trade-off is very small, then other means should be considered to upgrade the robustness of the network like increasing the capacity of some crucial links.

All the proposed approaches deal with OSPF links’ weights (costs) to reduce the effect of the congestion. Changing the links’ weights affects the distribution of all flows across the network. The privileges of our approach are that it provides semi-central management by using the controller as a real time device that deals with variable traffic patterns and it changes the distribution of specific flows to deal with the congestion problem. This does not change the links’ weights across the network.

Using the FRR mechanism to protect an OSPF/IP routed network is proposed in [106]. It is called IPFRR. The IPFRR is a protection scheme which computes the backup path prior to the failure event while the Fast Emergency Path scheme (FEP- scheme) is to help during the convergence period by generating a shorter recovery path, which is added as small extension in the Forward Information Base (FIB). The FEP-S utilizes the link state database information updated by the OSPF operation in order to compute the backup path. The OSPF algorithm uses the link weight as *metric* to pre-compute the backup path. We believe that using the IPFRR is not much feasible as the convergence time of the IP/OSPF routed network is very short.

6.5.2 Fast Re-Route (FRR) Mechanism Development Efforts

The OSPF/MPLS networks use the FRR mechanism to sooth the effect of link failure on their performance. There have been several studies for improving the performance of FRR.

Building auto backup tunnels is implemented in [100]. Cisco routers have the ability to dynamically build primary tunnels as well as backup tunnels.

In [107], Bartons et al., propose establishing two protection paths between a node and an egress router in a way that a single link failure would not cause simultaneous loss of connectivity between them. It considers the number of protection paths per link and the length of the protection paths to measure the quality of a protection scheme.

Our approach proposes establishing two backup paths to protect each link across the primary path. Whether their merge point ends at the egress router or not, it chooses the shortest ones to the egress routers. It considers the length of the protection paths as well as the bandwidth factor.

In [102], Kodialam et al propose an online routing algorithm to route both the active (primary) path and the backup path for each link across the active path. This algorithm is applied at the ingress router and concentrates on bandwidth routing for setting up QoS guaranteed paths. It tests three levels of information for routing: No information case, complete information case and partial information case. It adopts the latter one as it is the most feasible one. The amount of bandwidth sharing that can be achieved on the backup path is a function of quantity of information that is known about the routing of demands that are currently active in the network.

The previously proposed approaches of [102, 107] are applied at the routers where there is no controller used and the routers should coordinate with each other to provide efficient bandwidth occupation. Our approach is applied at a controller where there is no need to coordinate among routers which provides easier calculations and better results.

Use of the *p*-cycle scheme to route the bandwidth guaranteed backup tunnels is investigated in [108] by Kang et al. It adapts the *p*-cycle concept to the MPLS layer where it controls bypass tunnels engineered using label stacking under the assumption that all the LSPs in the network have particular bandwidth needs and a backup tunnel should supply guaranteed bandwidth to the LSPs protected by it. The backup paths are pre-computed but not pre-allocated. The *p*-cycle based protection scheme is a technique for the design of mesh restorable networks. The basic idea of *p*-cycle is to build the protection paths by using the concept of fully pre cross-connected linear segments [109]. The *p*-cycle is limited to mesh networks, while our approach is more general as it deals with mesh networks as well as other topologies.

Using a centralized manner to pre-calculate the protection tunnels and reserved bandwidth is proposed in [110] where Klopfenstein et al. use SNMP to collect link load data. The proposed algorithm ensures the lowest congestion possible in the network in any possible failure case. This path computation is performed at the network management platform. When the protection paths are computed, the routers are configured accordingly. In our approach, the controller calculates and installs both of the primary and the backup tunnels dynamically by using specific messaging system to communicate with routers.

In [96], Mereu. et al. implement hybrid IGP/MPLS path restoration procedure. In order to avoid congestion scenarios due to link failures, it proposes two restoration schemes: the link restoration scheme, where the traffic is routed through a back-up path that is found to protect the failed link and the path restoration scheme, where if a failure occurs along a path, the traffic is routed in a completely new path that starts from the origin source node of the commodity and terminates in the destination node of the commodity. In our approach, the controller is responsible for calculating the two backup tunnels (if available) for each link lies on the path of the primary tunnel. It allocates specific bandwidth to serve the primary tunnels as well as it allocates extra specific bandwidth to serve the backup tunnels. There is no bandwidth sharing between the primary tunnels and the backup tunnels and there is no possible congestion that follows any link failure. Using the backup tunnels is temporary until establishing new primary tunnels which compensate the old ones.

Protecting a network against multiple link failures was proposed in [111] by Sinha et al. They describe several designs of networks that add a small number of edges to an existing topology. Another advantage of our approach is the scenario is followed by the routers to check the validity of backup tunnels periodically. This helps in dealing with multiple link failures. If a failure occurs on a link that lies on a path of backup tunnel or the bandwidth of some links within the backup path have been occupied by another backup tunnel that serves a different primary tunnel suffers from link failure apart from this area. This backup tunnel will be removed and the PLR either only adopts the second backup tunnel (if there is any) or informs the controller which takes the responsibility of calculating a new backup tunnel path.

In [112], A. Jarry proposes two algorithms to compute the shortest guaranteed primary MPLS paths with their backup towards a single destination. The first algorithm is used in the directed graphs while the other is used in the undirected graphs. When computing the primary and the backup paths, those algorithms consider some of the quality of service constraints represented by the cost function that measures link latencies. We believe that those algorithms do not deal with real time applied load. Even though the computed primary and backup tunnels matches the shortest available paths, they are not necessary to be the best paths of all time. Sometimes, flows should be diverted through other paths to provide better bandwidth occupation. Our approach proposes the establishment of dynamic primary and backup paths where choosing of paths depends on both the topology and the applied traffic pattern to achieve better possible distribution.

We believe that our approach represents the first attempt to use a separate controller provided with knowledge and ability to affect the network in calculating and installing the MPLS primary and backup tunnels taking the bandwidth sharing matter into consideration. Our approach deals with all issues discussed in [96, 100, 107, 108, 110, 111]. In addition to that, involving the controller in calculating and installing dynamic primary MPLS tunnels and their dynamic backup tunnels represents a separation between the control plane and the data plane.

6.6 Decreasing the Effect of Link Failure by Using the Controller

The procedure of this chapter is a continuation to the previous chapters. Inet-OMNeT++ project is the modelling tool. The Inet-OMNeT++ project gives neither explanation nor example about link failure procedure in OSPF network. The development of OSPF network under a link failure event involves achieving successful link failure followed by network recovery under standard conditions. Then, any development in network performance can be added later. Thus, the methodology consists of three main steps:

- Procedure of achieving successful link failure in an OSPF network.
- Procedure of soothing the link failure effect in OSPF/IP routed networks.
- Procedure of soothing the link failure effect in OSPF/MPLS networks.

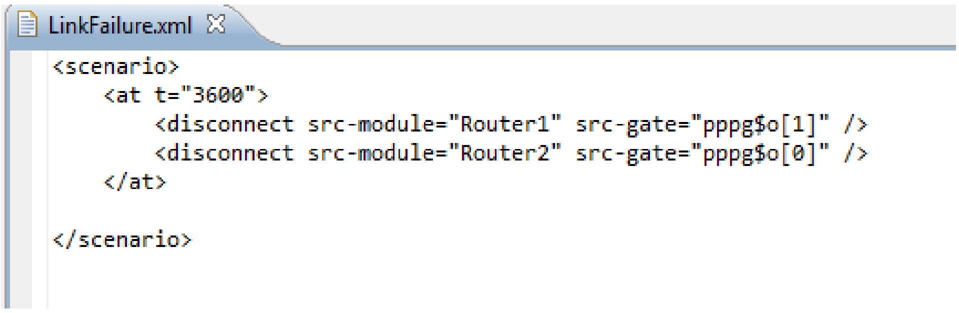
Those procedures are applied under condition that neither congestion has resulted from high data rates nor flows have been diverted before the link failure event.

6.7 Procedure of Achieving Link Failure Event Followed by Network Re-convergence in OSPF Network of Inet-OMNeT++ Project

The first step is to develop the *OSPF Routing* model of the router so that it can handle the link failure. The link failure procedure in OSPF network of Inet-OMNeT++ project has been achieved according to the following procedure:

6.7.1 Main Procedure to Achieve Successful Link Failure

- Create a XML file inside the project and give the created file a name like: *LinkFailure.xml*.
- Inside *LinkFailure.xml* file, add the link failure attributes, which are represented by the time of the link failure and the link that should be stopped from working. Fig. 6-2 below shows the contents of the XML file which indicates that at time = 3600 seconds (one houer after the simulation commences), the link between router 1 and router 2 goes down.



```
<scenario>
    <at t="3600">
        <disconnect src-module="Router1" src-gate="pppg$o[1]" />
        <disconnect src-module="Router2" src-gate="pppg$o[0]" />
    </at>

</scenario>
```

Fig. 6-2 Contents of XML file

- Inside the *omnetpp.ini* file, add the instruction that invokes the created XML file during simulation.

  **.scenarioManager.script = xmldoc("*LinkFailure.xml*")

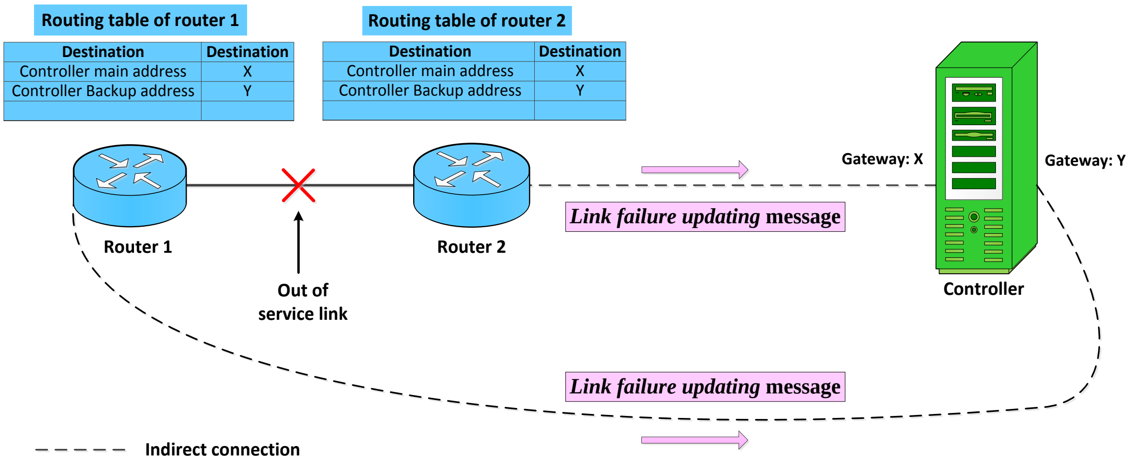
  The instruction above is added within the project parameters.
- During IP packets forwarding process, the *Developed IP* model (part of the *Network layer* model) of each router examines the link (Channel) forwarding availability before forwarding any packet via that link. In MPLS packets' forwarding process, the *Developed MPLS* model of the router performs the same testing procedure on the forwarding link.
- Link failure removes the link from existence (Channel == Null).
- When the router detects the link failure, it indicates the name and the number of the failed links then it activates the convergence scenario at the *Developed OSPF Routing* model.
- The interface state is altered to *DOWN*.
- All timers (*the Hello Timer*, *the Wait Timer and the Acknowledgement Timer*) of the *Developed OSPF Routing* model that related to the failed interface are cleared [113].
- The adjacent neighbour of the failed link is removed from the neighbour table and the neighbour's states are put to *DOWN*.
- This activates the generation of new *OSPF Link State Advertisement (LSA)* messages, which updates the routing tables of all routers according to the new topology.

6.7.2 Extra Steps in Dealing with Link Failure

After detecting the link failure and executing the network convergence procedure, the routers that were connected to the failed link perform the following procedure:

- If our network is an OSPF/IP routed network, it converges very fast. The routers of the failed link immediately commence forwarding the packets of the flows, which were forwarded through the failed link according to the new calculated routing tables.
- If our network forwards packets under the MPLS technique (an OSPF/MPLS network), the routers of the failed link will activate the backup tunnels immediately when detecting the link failure. They commence forwarding the packets of the flows, which were forwarded through the failed link to the directly connected links of the backup paths.

- Both routers connected to the failed link inform the controller about their failed link by sending *Link failure updating* messages to the primary address and all the secondary addresses of the controller. This operation is done to guarantee that the controller is informed about the failed link as fast as possible. The operation of informing the controller about the link failure is illustrated in Fig. 6-3 below.


Routing table of router 1

Destination	Destination
Controller main address	X
Controller Backup address	Y

Routing table of router 2

Destination	Destination
Controller main address	X
Controller Backup address	Y

Fig. 6-3 The link failure updating messages sent to the controller

The *Link failure updating* message notifies the controller of the place and the address of the failed link. Its structure was shown in Fig. 3-16 of **chapter 3**.

- Upon receiving any *Link failure updating* message, the controller updates its database according to the received information. It activates the *Link failure handler* model, which invokes the stored paths, eliminates those passes through the failed link and re-stores the other paths. The new topology of the network inside the controller is replica to the real new one after the link failure.
- After updating the routing tables to the new topology, each router sends its routing table to the controller through generated *Routing table* message. The controller updates its database according to the received *Routing table* messages. The structure of the *Routing table* message was shown in Fig. 3-9 of **chapter 3**.
- The routers rebuild their routing tables according to the new topology so that they can convey the *Control* messages issued by the controller to their destinations. The controller updates its routing table to be compatible with the new topology as well.
- This link failure scenario was applied on the network of 9 routers of Fig. 6-4 and on the network of 13 routers (UK amp) of Fig. 6-10. It showed successful results.

6.8 Soothing the Link Failure Effect in OSPF/IP routed Networks

After updating its database, the controller applies FDT to examine flows' distribution over the new topology according to the updated routing tables. If there is any congestion, the controller applies its FDA to solve the problem as usual.

Example, suppose the network of Fig. 6-4, where flows are distributed normally without congestion. The controller has recorded all flows passing through the network as it has the full knowledge about the network. Suddenly, the link connecting router 2 with router 4 goes down. Router 2 and router 4 inform the controller about the link failure by sending a *Link failure updating* message, as shown in Fig. 6-4 as well.

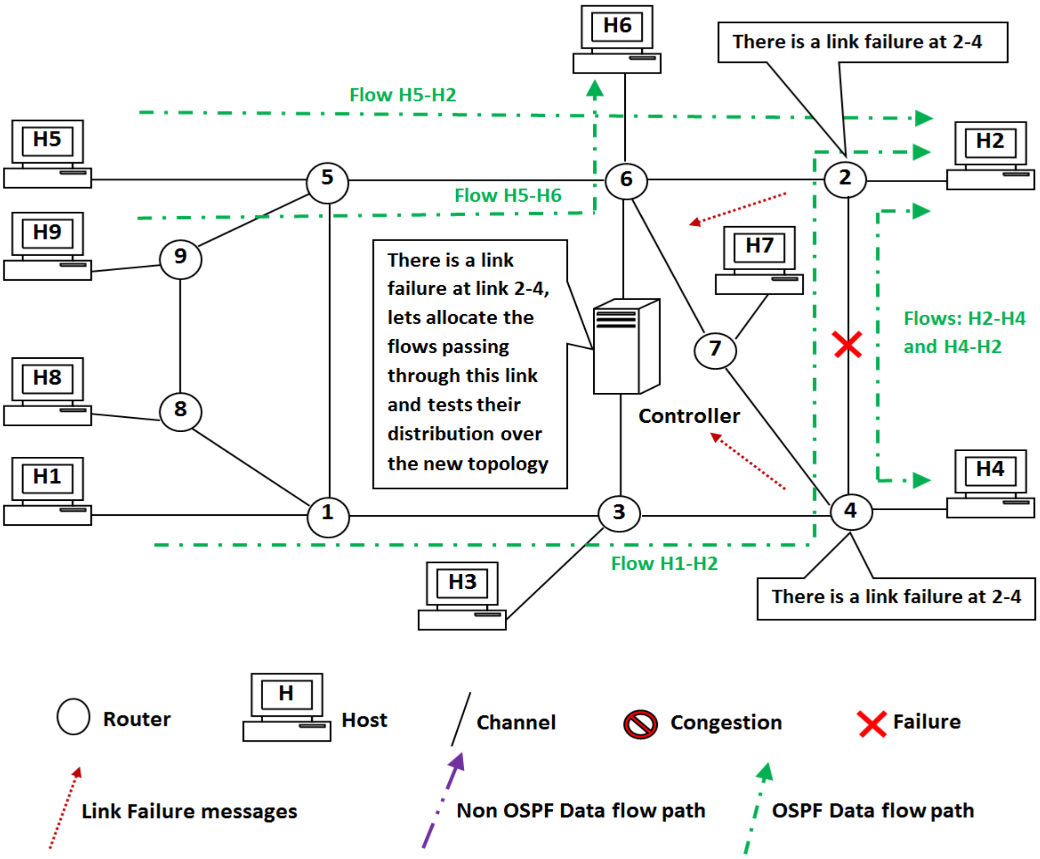

H6
There is a link failure at 2-4
Flow H5-H2
H5
H2
5
6
2
Flow H5-H6
H9
9
H7
There is a link failure at link 2-4, lets allocate the flows passing through this link and tests their distribution over the new topology
Flows: H2-H4 and H4-H2
H8
7
8
Controller
H1
H4
1
3
4
Flow H1-H2
H3
There is a link failure at 2-4
Router
H
Host
Channel
Congestion
Failure
Link Failure messages
Non OSPF Data flow path
OSPF Data flow path

Fig. 6-4 Link failure at link 2-4 of the 9 routers network

Upon receiving the first arrived *Link failure updating* message, the controller updates the recorded topology of its database to the new topology. It removes all the calculated paths that pass through link 2-4 and allocates all flows pass through the link as well. The controller performs all those operations inside its *Link failure handler* model.

The allocated flows passing through link 2-4 are:

- Flow H1-H2.
- Flow H2-H4.
- Flow H4-H2.

Beside the *Link failure updating* message that is sent to the controller, the routers at the place of failure (router 2 and router 4) update their routing tables according to the new topology as fast as possible and commence sending *OSPF Link State Update Advertisement (LSA)* messages to their neighbours to update their routing tables according to the new topology as well.

After updating the routing tables of all the routers, each router creates a *Routing table* message, inserts its new routing table inside it and sends it to the controller. When receiving the *Routing table* messages from all the routers of the network, the controller updates its stored OSPF paths through applying the procedure of extracting the OSPF paths of Fig. 3-13 of **chapter 3** before. Then, it applies its FDT to test the distribution of the flows that were passing through the failed link and finds the results of Table VI-I.

TABLE VI-I
TESTS OF FLOWS DISTRIBUTION AFTER LINK FAILURE

Flow	New distribution Path	FDT results
H1-H2	1-5-6-2	Not OK – There is congestion
H2-H4	2-6-7-4	OK – There is no congestion
H4-H2	4-7-6-2	OK – There is no congestion

FDT finds that the distribution of the flows in the new topology according to the updated OSPF routing table causes a congestion problem at link 5-6. This situation is similar to the network as it is without controller shown in Fig. 6-5.

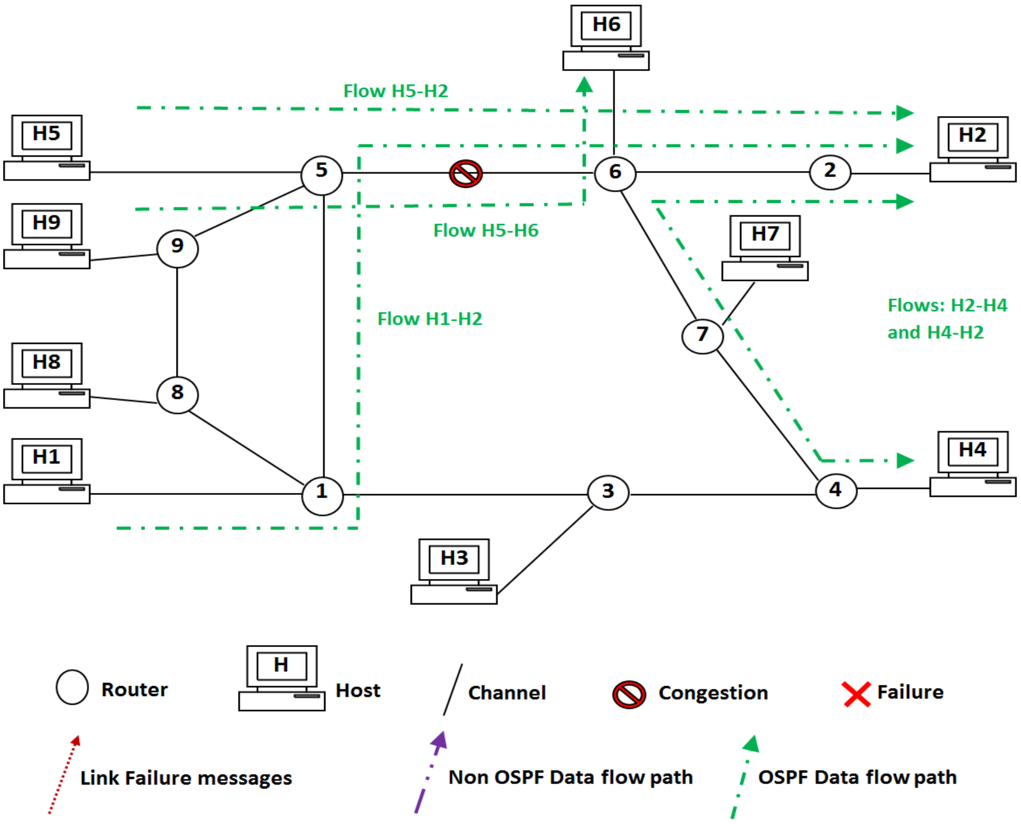


Fig. 6-5 Congestion over network of 9 routers (without controller) after link 2-4 fails

The controller allocates the flows that are passing through the congested link/links (flows: H1-H2, H5-H2 and H5-H6). The controller applies FDA upon the flows at the area of congestion (link 5-6). FDA finds that the only flow that can be diverted to a less long path is flow H1-H2. The controller diverts flow H1-H2 from its main path that passes through routers 1-5-6-2 to a detour passes through routers 1-3-4-7-6-2. Flow H1-H2 traverses the new path to its destination as shown in Fig. 6-6.

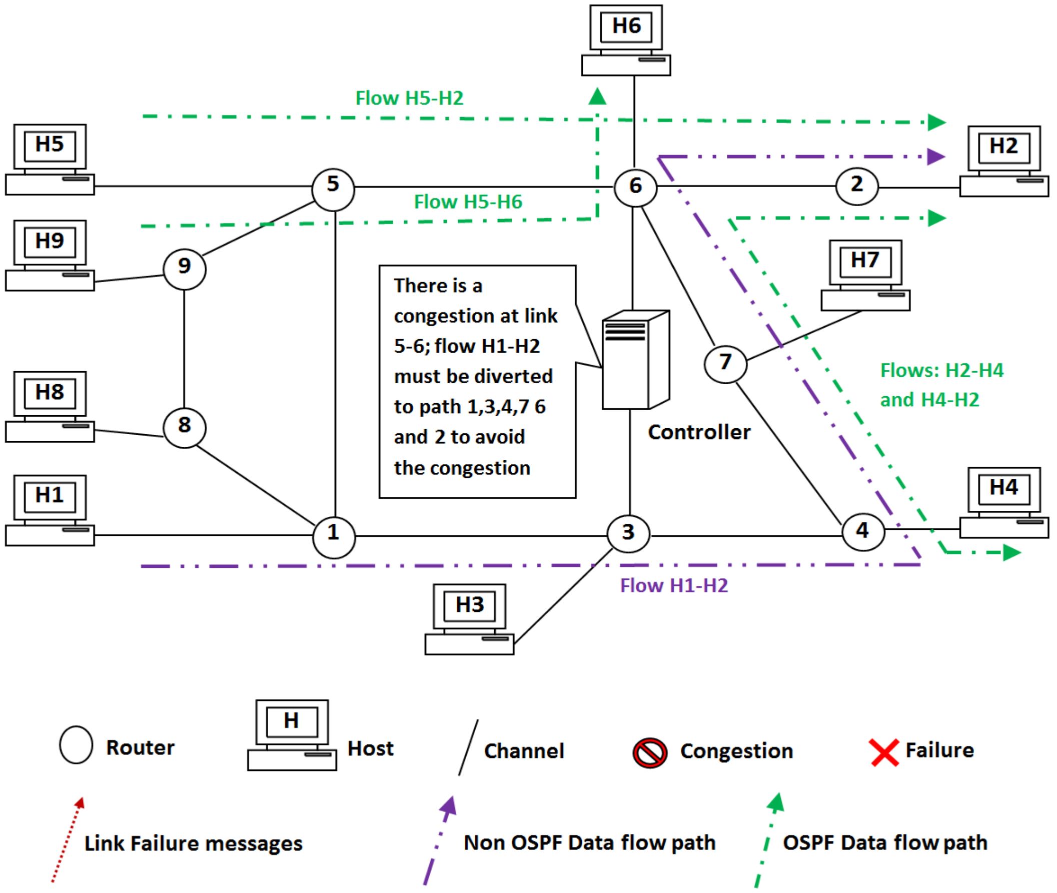


Fig. 6-6 Flows re-distribution to avoid congestion

The results obtained are recorded for the same link failure problem in identical networks one with controller and the other is without controller for total simulation time of 20 minutes and link failure occurs at time of 15 minutes. The average throughput recorded is as shown in Fig. 6-7 which shows only the last 11 minutes of the simulation.

The differences between the plot of network with controller (in blue) and the network without controller (in red) represents the packets dropped because of the congestion occurred after the link failure. The controller saved the packets of flow H1-H2 by diverting it to an alternative path as well as the packets of flows H5-H2 and H5-H6. The latency of flows H5-H2 and H1-H2 are shown in Figs. 6-8 and 6-9 respectively where both figures show that using the controller achieved less latency.

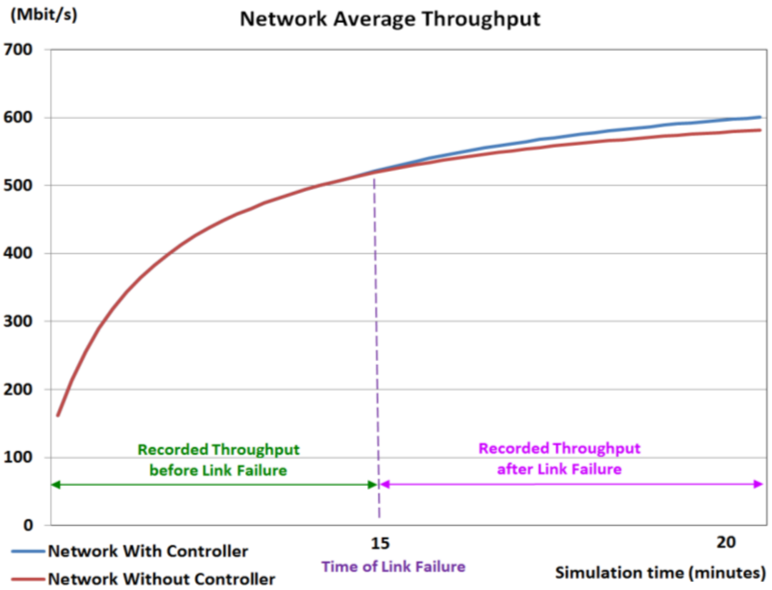


Fig. 6-7 Network average throughput recorded before and after link failure

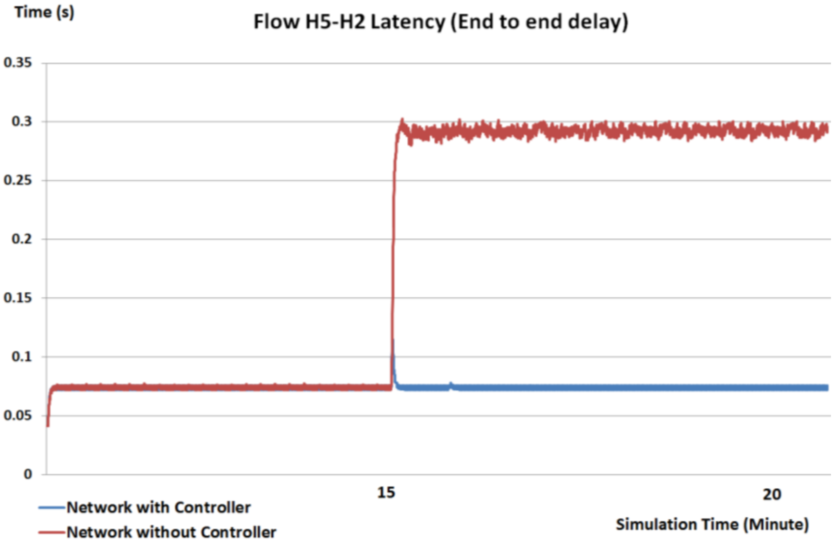


Fig. 6-8 Flow H5–H2 latency (with and without controller)

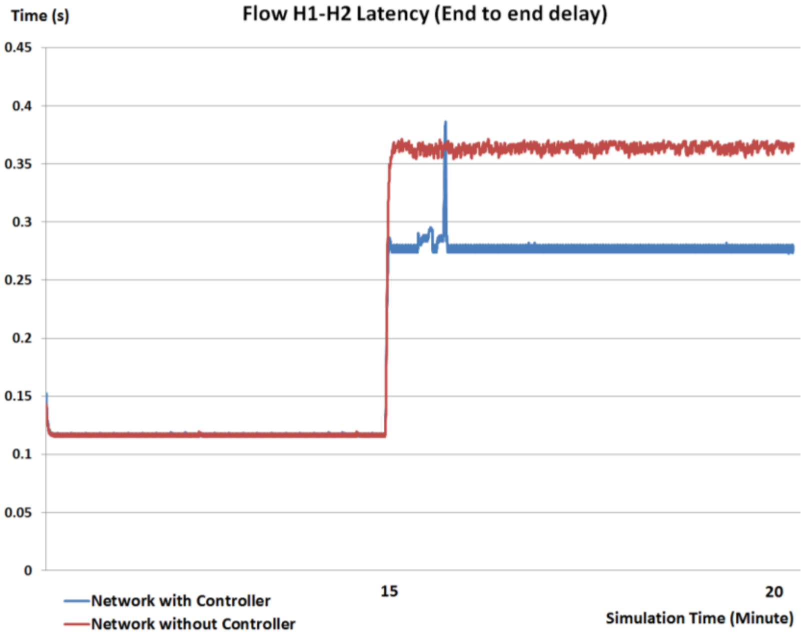


Fig. 6-9 Flow H1–H2 latency (with and without controller)

6.9 Soothing Link Failure Effect in OSPF/MPLS Networks

The controller provides MPLS backup tunnels to the previously established primary MPLS tunnels. The controller prepares the backup tunnels within a short time after establishing each primary MPLS tunnel by using a specific algorithm. It installs the backup tunnels inside each router over the primary MPLS path as well as over the other backup routers within long time before the link failure event (make before break).

By allocating and establishing the backup tunnels, the controller tries to provide full protection to the primary MPLS path. The backup tunnels are ready to be used when a link failure event takes place. The title: ***Fast Re-route*** by adopting the ***SDN*** notion **(*FRR-SDN*)** will be given to the procedure for calculating the backup path and establishing them into the involved routers. The ***FRR-SDN*** consists of three steps:

- Backup Paths Calculation Algorithm (BPCA).
- Efficient bandwidth allocation mechanism.
- The installation of the backup tunnels.

Routers respond to the commands ordered by the controller related to backup tunnels. They establish the backup tunnels and keep them in an inanimate status. The controller calculates the paths of the backup tunnels and allocates bandwidth for them in its database without bandwidth reservation on the routers. After installing the backup tunnels into the involved routers, it is the responsibility of the routers to maintain them and to check their validity from time to time. This operation is performed by routers according to specific scenario. All the messages that are used to establish the backup tunnels or to check the validity of their paths are forwarded across the network by using the IP forwarding mechanism and according to the OSPF routing tables of the routers.

To avoid congestion, the controller may change the path of some of the primary MPLS tunnels. In such situations, it also changes the paths of their backup tunnels. The routers of the extinct primary tunnel remove the backup tunnels related to it. The controller establishes new backup tunnels capable of serving the new primary MPLS path. Therefore, the backup tunnels are dynamic as well as the primary tunnel they protect.

At the event of link failure, the routers involved with the event activate the backup paths to avoid service outage. The MPLS forwarded flows, which were passing through the failed link, commence using the backup paths until establishing new alternative primary tunnels.

6.10 Features of the Dynamic Backup Tunnels

The backup paths calculated by the controller have the following features:

- The protection of any link at the primary MPLS path depends on the link location within the topology. Some links are locally protected with two backup paths; some others are protected with one backup path while the remains are not locally protected with any backup path as the controller could not find any backup path to protect them (lack of resources). The controller tries to include the protection of the non-locally protected links within the protection of their previous neighbour links via the primary path by applying a specific scenario.
- Some of the backup tunnels can be ***full detours***; some others are ***partial detours*** while the remains represent ***one link protectors***. The ***full detour*** is the alternative path that begins at the head-end router and ends at the tail-end router without passing through any other routers or links that belongs to the primary path. It protects all the links of the primary path (end-to-end protection). The ***partial detour*** is the alternative path that begins from any router within the primary path except the tail-end and the pen-ultimate routers and it protects more than one link over the primary path.
- The paths of the primary tunnels and the paths of the backup tunnels share many links but without bandwidth sharing. The backup tunnels share bandwidth with each other and this sharing provides more bandwidth saving.
- The routers of the primary tunnel regularly check the validity and the residual bandwidth of the backup paths so that the router of the primary path can choose which backup path to use at the event of link failure. The residual bandwidth regarding the MPLS technology is defined as the difference between the link capacity and the amount of bandwidth already utilized by the primary tunnels and their backup tunnels traversing the link [102].
- The controller provides each backup tunnel with a global specific identification number. The rerouted packets carry this label number when they pass via the backup tunnel.
- The controller removes from its database all the backup tunnels related to a primary tunnel when the primary tunnel is deleted. The related routers also remove those backup tunnels as well.

6.11 Fast Re-route by Adopting the SDN Notion (FRR-SDN)

The controller is acquainted about the network topology including the bandwidths of all links and all possible paths from each router to all other routers. It obtains the data it needs from its database which was set as illustrated in **section 3.12.1** of **chapter 3** which related to controller's knowledge. It also uses its routing table to contact the involved routers via the nearest gateway.

6.11.1 Backup Paths Calculation Algorithm (BPCA)

The controller applies the following mechanism at its *Fast reroute* model by using the stored paths in its database to calculate the backup paths:

- Define a network with specific topology. The network consists of number of nodes (routers) and number of links.
- Define $\boldsymbol{A}$ and $\boldsymbol{B}$ as two nodes that belong to the network.
- Define a flow of specific data rate commencing from node $\boldsymbol{A}$ and terminating at node $\boldsymbol{B}$ passing through some other nodes and links over the established primary MPLS path denoted by $\boldsymbol{P_{A-B\,MPLS}}$. The primary MPLS tunnel path $\boldsymbol{P_{A-B\,MPLS}}$ consists of several nodes and links.
- Define $\boldsymbol{L_{Pl}}$ as a link which the controller wants to protect. The protected link $\boldsymbol{L_{Pl}}$ can be any link that lies over the primary MPLS path $\boldsymbol{P_{A-B\,MPLS}}$.
- The router that precedes the protected link and is directly connected to it is called the point of local repair router. It is denoted as $\boldsymbol{plr}$. Generally, the point of local repair can be any router over the MPLS path except the tail-end router. It is the place where the backup tunnel begins.
- The backup tunnel merges with the primary tunnel at any router that follows the protected link but does not necessarily have a direct connection to it and it is called the merge point router. The merge point router is denoted as $\boldsymbol{mp}$. It is the router where the backup tunnel terminates. Generally, a merge point router can be any router over the MPLS path except the head-end router. The routers that precede the point of local repair router along the primary path cannot be merge points. All the routers and the links of the backup tunnel path should be apart from the protected primary MPLS tunnel path except the point of local repair and the merge point.
- From its database, the controller identifies all paths start from a $\boldsymbol{plr}$ and ends at any following merge point $\boldsymbol{mp}$ that lies in sequence after the protected link. It avoids the protected link and the loop point routers.

- The modified MPLS path between $\boldsymbol{A}$ and $\boldsymbol{B}$ that is used during link failure event is: $\boldsymbol{P}_{(\boldsymbol{A}-\boldsymbol{plr}\,)} + \boldsymbol{P}_{(\boldsymbol{plr}-\boldsymbol{mp})} + \boldsymbol{P}_{(\boldsymbol{mp}-\boldsymbol{B})}$
- The controller calculates all possible paths that can protect a primary MPLS tunnel. It starts the adoption operation from the shortest upward.
- If there is only one backup tunnel to protect this link $\boldsymbol{L}_{\boldsymbol{pl}}$, The controller adopts this backup path. If there are several backup tunnels to protect this link $\boldsymbol{L}_{\boldsymbol{pl}}$, the controller adopts the first and the second shortest paths to the tail-end router. If the controller could not find any backup tunnel to protect this link $\boldsymbol{L}_{\boldsymbol{pl}}$, the controller does not adopt any specific backup tunnel start from this directly connected router and in this case, tries to protect this link by including the previous link protection and protecting both links with one bypass backup tunnel that starts from the previous router.

6.11.2 Bandwidth Allocation Scenario

The bandwidth requirement for a MPLS tunnel is the most important characteristic [37] pp 284. In **chapter 5**, we removed the *RSVP-TE PATH* message and the establishment of the MPLS tunnel is performed by using the *RSVP-TE RESV* message only. In the establishment of the primary backup tunnels, the bandwidth is reserved along the links of the routers. The controller establishes the primary MPLS tunnels and records the reserved bandwidth of each tunnel in its database. The database map of the controller is real time image of the network and all its attributes including flows’ distribution.

In this chapter, the controller is responsible of establishing the backup tunnels. The controller does not want the routers to reserve any bandwidth for the backup tunnels on their links. However, it adds the allocated bandwidth to its database. The controller may use part of the allocated bandwidth that serves a specific backup tunnel to serve another backup tunnel. The controller applies the following mechanism at its *Fast Re-Route* model to achieve *intra-demand* bandwidth sharing and the *inter-demand* bandwidth sharing among the paths of the backup tunnels.

i. Intra-demand bandwidth sharing

In the case of link that is segment of different backup tunnels that serve the same primary tunnel, the controller reserves a bandwidth value at the link which equals to the bandwidth of the primary tunnel only once.

- Define a network with specific topology. The network consists of number of nodes (routers) and number of links.

- Define $\boldsymbol{A}$, ***B, C, D, G*** and $\boldsymbol{H}$ as nodes that belong to the network.
- Define primary MPLS tunnel path $\boldsymbol{P_{A-B\,MPLS}}$ consists of number of nodes and number of links. The flow that is travelling from node $\boldsymbol{A}$ to node $\boldsymbol{B}$ is denoted by $\boldsymbol{F_{A-B}}$ and it has data rate of $\boldsymbol{BW_{A-B}}$. Some links over the primary MPLS tunnel path $\boldsymbol{P_{A-B\,MPLS}}$ are protected with the backup tunnel $\boldsymbol{P_{(plr_k - mp_q),\ A-B}}$ and some other links are protected with the backup tunnel $\boldsymbol{P_{(plr_s - mp_t),\ A-B}}$.
- Define $\boldsymbol{L_{G-H}}$ as a link that hast total available bandwidth of $\boldsymbol{BW_{Available\ G-H}}$. It does not belong to the MPLS tunnel path $\boldsymbol{P_{A-B\,MPLS}}$. Both of the backup tunnels that protect the primary tunnel $\boldsymbol{P_{A-B\,MPLS}}$are passing through link $\boldsymbol{L_{G-H}}$. This makes $\boldsymbol{L_{G-H}}$ as common link between $\boldsymbol{P_{(plr_k - mp_q),\ A-B}}$ and $\boldsymbol{P_{(plr_s - mp_t),\ A-B}}$.
- The controller reserves a bandwidth value equal to the data rate of flow $\boldsymbol{F_{A-B}}$ only once at the link $\boldsymbol{L_{G-H}}$ regardless of the number of the backup tunnels passing through it as long as they serve the same primary tunnel:

  $$\boldsymbol{BW_{Reserved\ G-H}} = \boldsymbol{BW_{A-B}}$$

  $$\boldsymbol{BW_{Residual\ G-H}} = \boldsymbol{BW_{Available\ G-H}} - \boldsymbol{BW_{A-B}}$$

Our approach applies this method of bandwidth sharing. However, we did not include its results in the results plots that are viewed later in this chapter.

ii. Inter-demand bandwidth sharing

In the case of link that is segment of different backup tunnels that serve multiple primary tunnels of different flows, the controller reserves a bandwidth value at the link which equals to the highest recorded flow data rate among the flows.

- Define the network illustrated at the previous section.
- In addition to the previously mentioned primary MPLS tunnel path $\boldsymbol{P_{A-B\,MPLS}}$, define another primary MPLS tunnel path $\boldsymbol{P_{C-D\,MPLS}}$ that consists of number of nodes and number of links. The flow that is travelling from node $\boldsymbol{C}$ to node $\boldsymbol{D}$ is denoted by $\boldsymbol{F_{C-D}}$ and it has data rate of $\boldsymbol{BW_{C-D}}$. Some links over the primary MPLS tunnel path $\boldsymbol{P_{C-D\,MPLS}}$ are protected with the backup tunnel $\boldsymbol{P_{(plr_e - mp_f),\ C-D}}$.
- $\boldsymbol{L_{G-H}}$ is a common link between $\boldsymbol{P_{(plr_k - mp_q),\ A-B}}$ and $\boldsymbol{P_{(plr_e - mp_f),\ C-D}}$. This means that link $\boldsymbol{L_{G-H}}$ reserves two backup tunnels that protect two different primary tunnels of different paths: $\boldsymbol{P_{A-B\,MPLS}}$ and $\boldsymbol{P_{C-D\,MPLS}}$.
- In this case the controller reserves a bandwidth value at the common link that

equals to the data rate of one primary tunnel and chooses the highest among them: $\boldsymbol{BW_{Reserved\ G-H}} = \boldsymbol{Max}\left(\boldsymbol{BW_{A-B,}}\ \boldsymbol{BW_{C-D}}\right)$

- If $\boldsymbol{BW_{C-D}}$ is greater than $\boldsymbol{BW_{A-B}}$

  Then, $\boldsymbol{BW_{Reserved\ G-H}}$ equals to $\boldsymbol{BW_{C-D}}$

  And $\boldsymbol{BW_{Residual\ G-H}} = \boldsymbol{BW_{Available\ G-H}} - \boldsymbol{BW_{C-D}}$

The reserved bandwidth can serve any of the flows $\boldsymbol{F_{A-B}}$ or $\boldsymbol{F_{C-D}}$ if their primary MPLS tunnel paths suffer from link failure at routers $\boldsymbol{plr_k}$ or $\boldsymbol{plr_e}$ respectively.

Example, suppose utilizing the network of 13 routers that is shown in Fig. 6-10. The controller establishes a primary MPLS tunnel that connects ***4*** with ***11***. The path of the primary tunnel ***4*** → ***11*** passes through routers: ***4 – 1 – 6 – 9 – 11***.

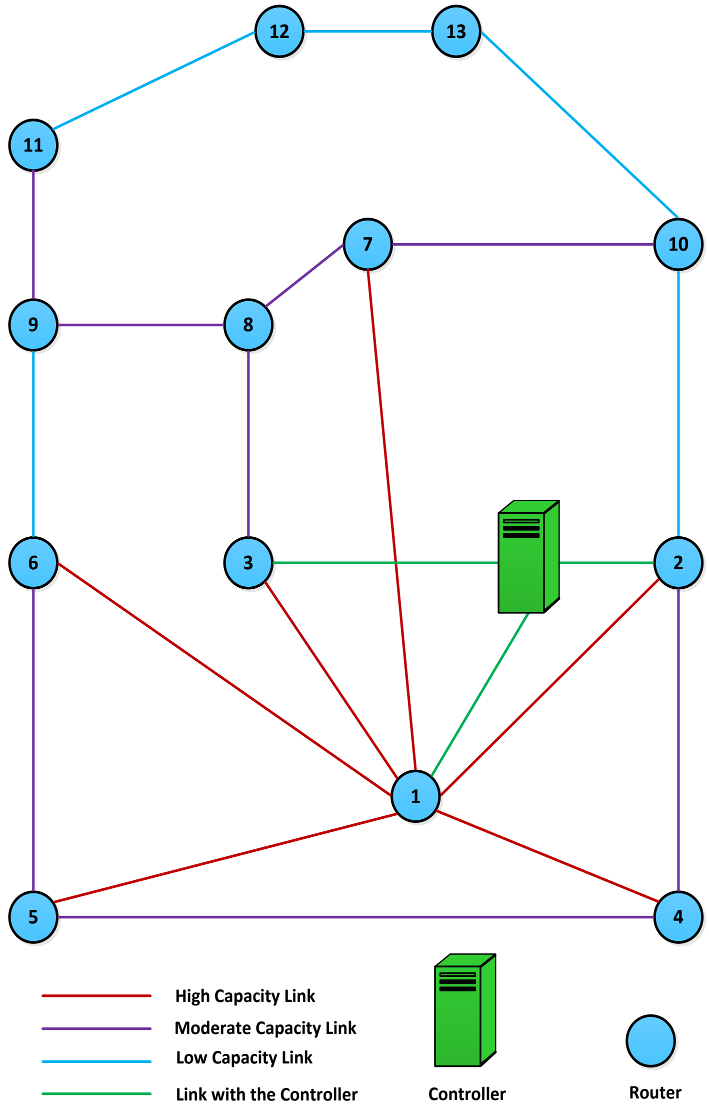


Fig. 6-10 Network of 13 routers

After establishing the primary tunnel ***4*** → ***11***, the controller according to its BPCA calculates the following paths of the backup tunnels that locally protect the links of the primary tunnel ***4*** → ***11***:

Link: ***4 – 1***

- *4 – 5 – 6.*
- *4 – 2 – 1.*

Link: ***1 - 6***

- *1 – 3 – 8 - 9.*
- *1 – 5 – 6.*

Link: ***6 - 9***

- *None.*

Link: ***9 - 11***

- *9 – 8 – 7 – 10 – 13 – 12 - 11.*

The bandwidth allocation and the inter-demand bandwidth sharing plot among backup tunnels that protect the primary tunnel ***4* → *11*** is zero as it is the only tunnel that exists on the network. It is shown in Fig. 6-11. However, there is an intra-demand bandwidth sharing among the backup tunnels that protects the primary tunnel ***4* → *11*** but we did not view it in bandwidth sharing plot. The only considered bandwidth sharing in all plots is the inter-demand bandwidth sharing.

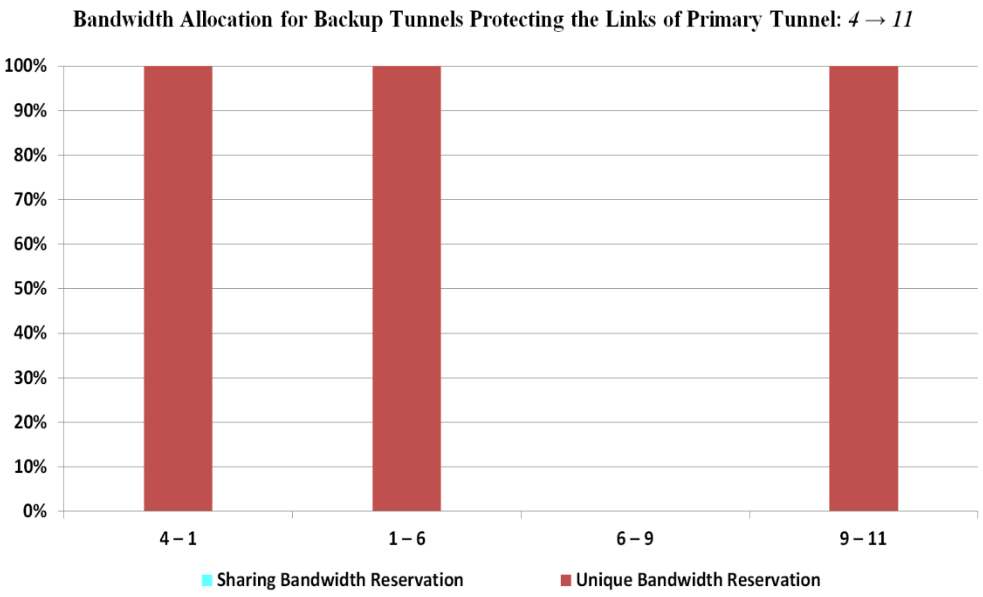


Fig. 6-11 Inter-demand bandwidth sharing of backup tunnels that protects single existing primary tunnel: *4 → 11*

Some links like (***6 – 9)*** cannot be locally protected as providing protection involves the availability of resources. The controller tries to include the protection of such links within the protection of the previous links. The backup tunnel: *1 – 3 – 8 – 9* can be considered as a protection to both ***1 – 6*** and ***6 - 9*** links. However, link ***6 – 9*** is not indicated as protected link in Fig. 6-11 as it is not locally protected.

In addition to the existing primary MPLS tunnel ***4* → *11***, the controller adds another primary MPLS tunnel to the network. The new MPLS tunnel connects ***1*** with ***12*** and passes through nodes: ***1 - 2 – 10 - 13 - 12.*** Then, the controller applies its BPCA which calculates the following backup paths to protect the links of the primary tunnel ***1* → *12***:

Link: ***1 – 2***

- *1 – 6 – 9 – 11 - 12.*
- *1 – 7 – 10.*

Link: ***2 - 10***

- *2 - 4 - 5 – 6 - 9 - 11 - 12.*

Link: ***10 - 13***

- *10 - 7 – 8 – 9 - 11 - 12.*

Link: ***13 - 12***

- *None.*

The backup tunnels that protect each link of the primary tunnel ***4* → *11*** share bandwidth with the backup tunnels that protect primary tunnel ***1* → *12***. The bandwidth allocation and the inter-demand bandwidth sharing plot of backup tunnels that protect the primary tunnels ***4* → *11*** is as shown in Fig. 6-12., while the bandwidth allocation and the inter-demand bandwidth sharing plot of backup tunnels that protect the primary tunnel ***1* → *12*** is as shown in Fig. 6-13. Both of the primary tunnels have the same bandwidth values.

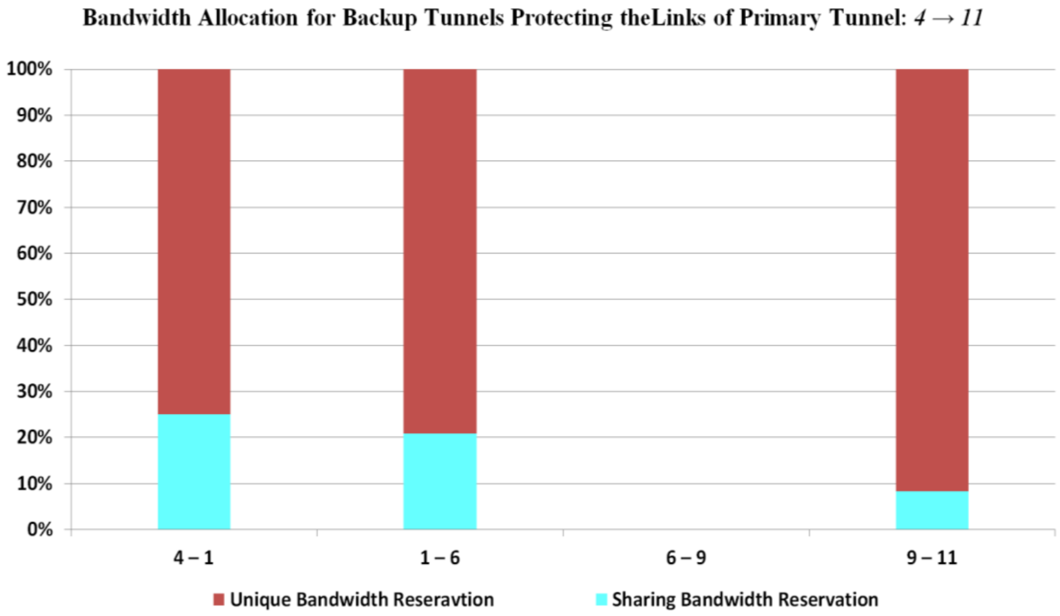


Fig. 6-12 Bandwidth allocation for backup tunnels that protects primary tunnel *4* → *11* sharing with the backup tunnels that protect primary tunnel *1* → *12*

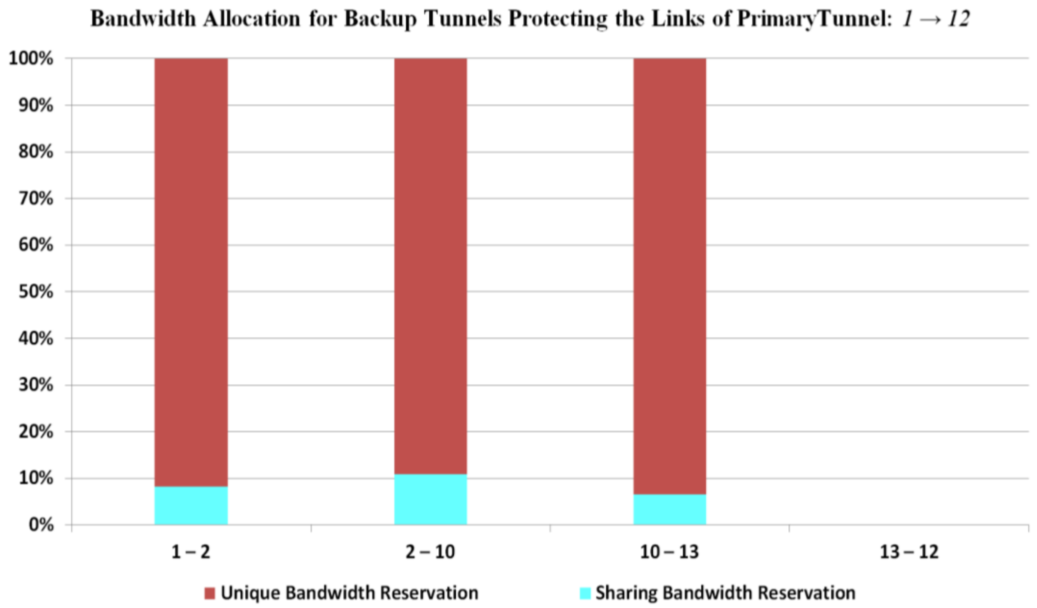


Fig. 6-13 Bandwidth allocation for backup tunnels that protects primary tunnel *1 → 12* sharing with the backup tunnels that protect primary tunnel *4 → 11*

For the Figures that show results, the red part of the column represents the real bandwidth value allocated to protect the mentioned link while the blue part refers to the values of the bandwidths allocated to protect the links of the other primary tunnels, which can be used to protect this link as well. Suppose the bandwidth of the primary tunnel ***4 → 11*** is 1 Mbps. The link ***4 – 1*** is protected via two different backup paths. The average bandwidth allocated at the backup links to protect link ***4 – 1*** should be 1 Mbps as in Fig. 6-11 while, it is 0.75 Mbps in Fig. 6-12. The reason behind this reduction in bandwidth allocation is the bandwidth sharing with the links that protect the primary tunnel ***1 → 12***.

Later, the controller adds extra four primary tunnels to the network: ***5 → 10, 2 → 8, 6 → 13*** and ***3 → 11*** and calculates the backup tunnels to protect them by applying its BPCA. The paths of all currently existing primary tunnels are as shown in Table VI-II:

TABLE VI-II
PATHS OF SIX PRIMARY TUNNELS' IN 13 ROUTERS NETWORK

Primary Tunnel Name	Primary Tunnel Path
4 → 11	*4 – 1 – 6 – 9 – 11*
5 → 10	*5 – 1 – 2 - 10*
1 → 12	*1 - 2 – 10 - 13 - 12*
2 → 8	*2 – 1 – 3 - 8*
6 → 13	*6 – 1 – 2 – 10 - 13*
3 → 11	*3 – 8 –9 - 11*

All the primary tunnels of Table VI-II have the same bandwidth values. The bandwidth allocation and the inter-demand bandwidth sharing plots of backup tunnels that protect each of the primary tunnels ***4 → 11, 1 → 12, 3 → 11*** and ***6 → 13*** with all other backup tunnels are shown in Figs. 6-14, 6-15, 6-16 and 6-17 respectively.

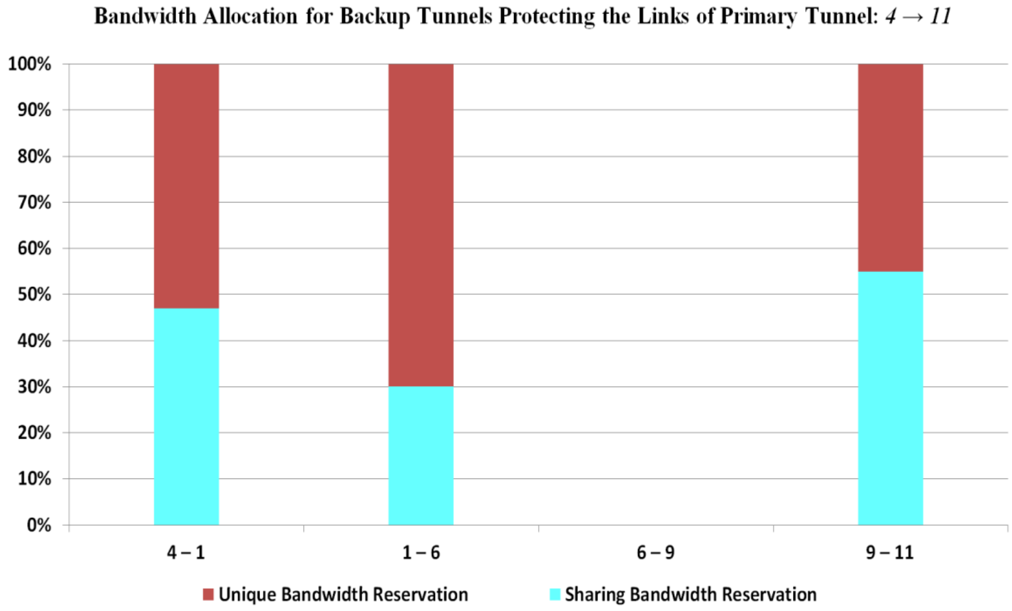


Fig. 6-14 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel: *4 → 11*

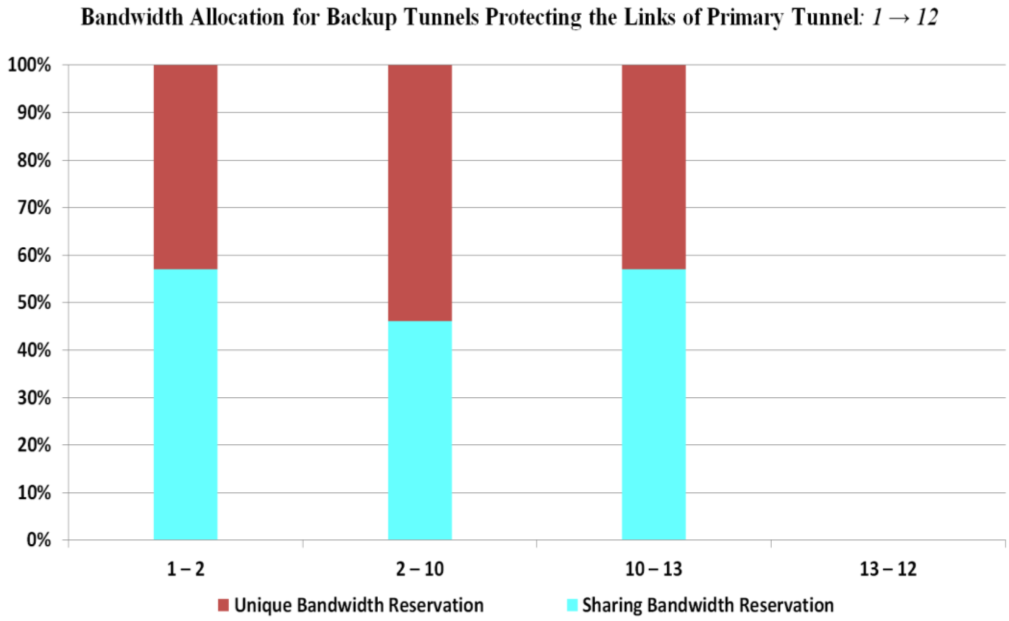


Fig. 6-15 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel: *1 → 12*

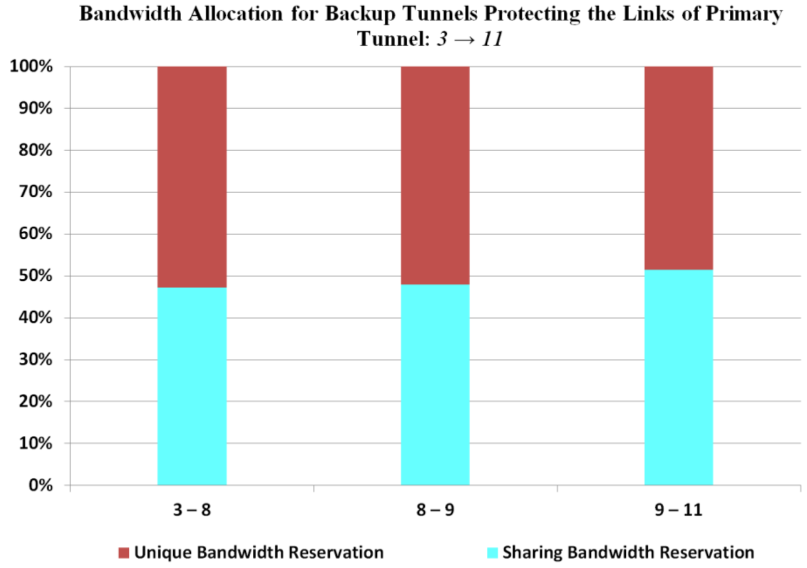


Fig. 6-16 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel: *3 → 11*

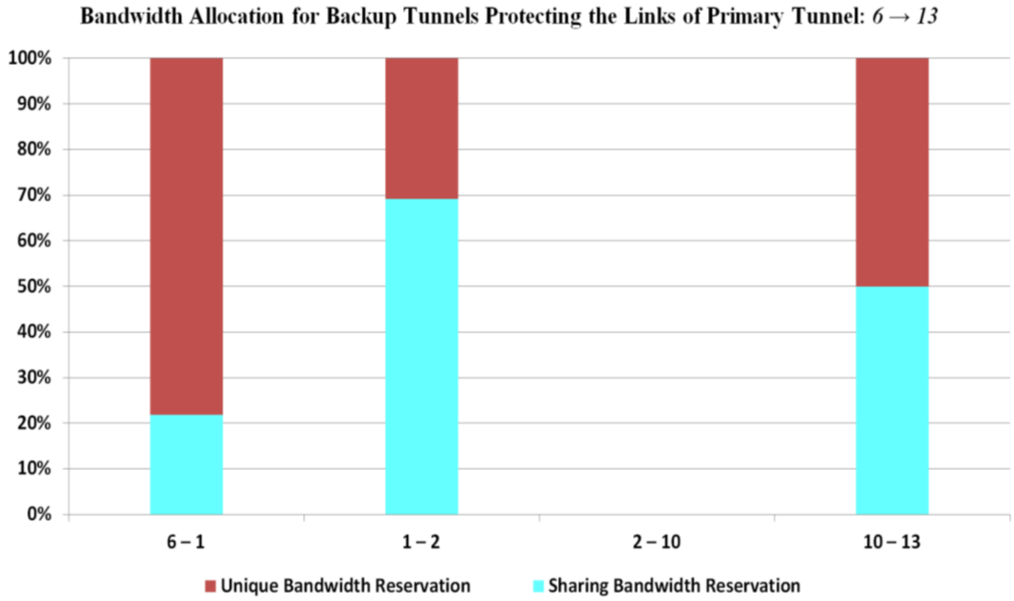


Fig. 6-17 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel: 6 → 13

The same procedure has been repeated by establishing the same primary tunnels but with different bandwidth values and as shown in table VI-III:

TABLE VI-III
ESTABLISHED TUNNELS OF DIFFERENT VALUES OF BANDWIDTH

Tunnel Name	Bandwidth (Mbps)
4 → 11	*2.64*
5 → 10	*2.64*
1 → 12	*5.44*
2 → 8	*5.44*
6 → 13	*9.44*
3 → 11	*9.44*

Figs. 6-18, 6-19 and 6-20 represent the bandwidth allocation and sharing plots of the backup tunnels that protect each of the following primary tunnels of bandwidths shown in table VI-III: ***4* → *11***, ***1* → *12*** and ***6* → *13***.

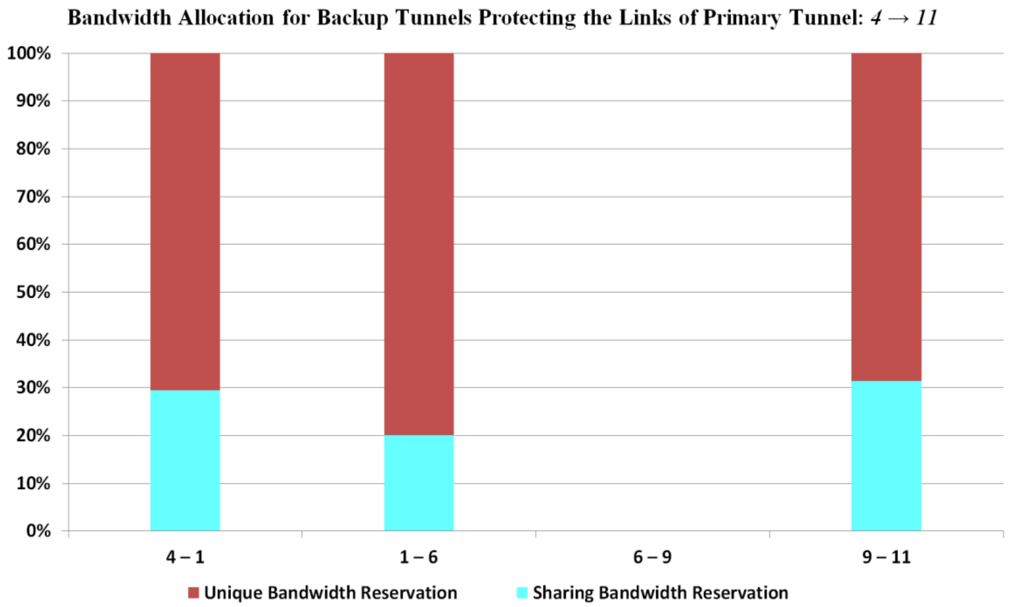


Fig. 6-18 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel: *4 → 11*

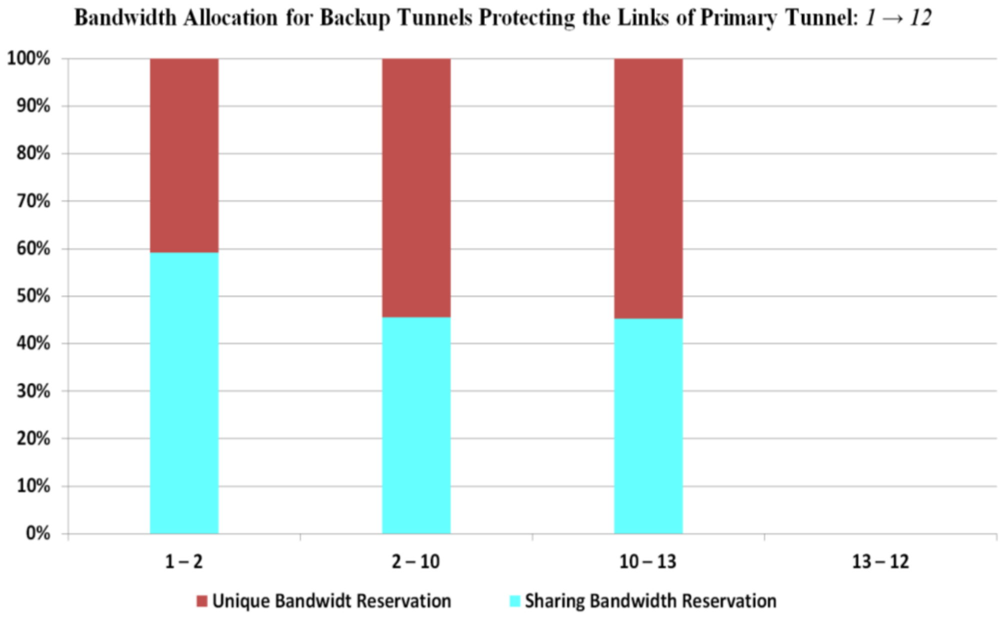


Fig. 6-19 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel: *1 → 12*

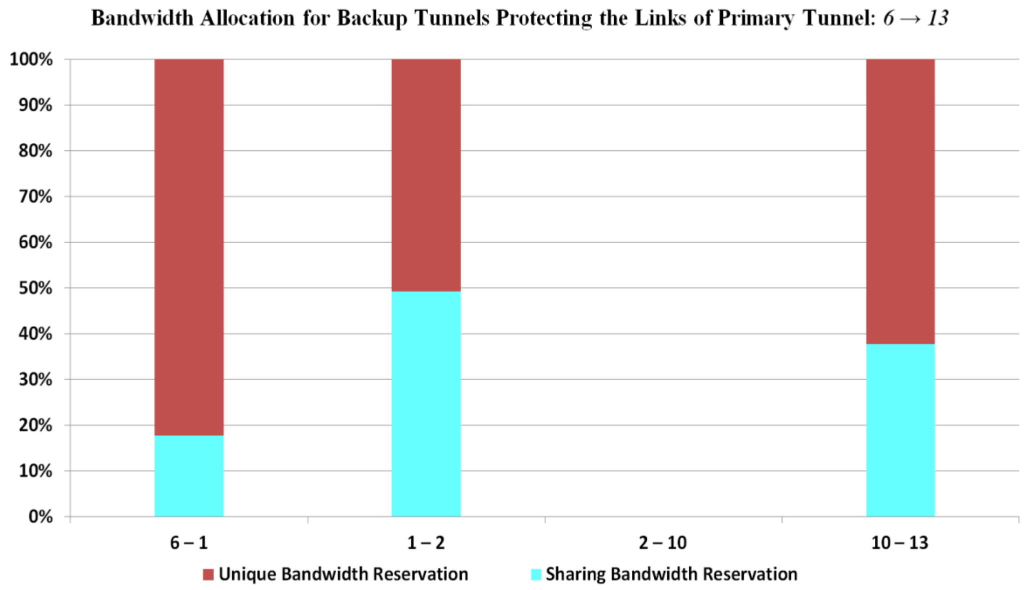


Fig. 6-20 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel: *6 → 13*

By comparing the bandwidth sharing of primary tunnels of same bandwidth represented in Figs. 6-14, 6-15, and 6-17 with the bandwidth sharing of primary tunnels of different bandwidths represented in Figs. 6-18, 6-19 and 6-20, it is obvious that the ones of the same bandwidth values achieve higher inter-demand bandwidth sharing.

6.11.3 Backup Tunnels Installation

After bandwidth allocation, the controller assigns specific identification numbers to each backup tunnel as it did before with the primary tunnels. Informing the routers to create backup tunnels without bandwidth reservation involves developing new kind of messages and creates specific containers inside the *Developed MPLS* models and *Developed LIB table* models of the routers to store the backup tunnels and use them when necessary. The controller informs each merge point router over the main MPLS path about the backup path that ends at it via *Controller backup tunnel creation* message which structure is shown in Fig. 6-21.

Protected (Primary) Tunnel Identifications and Attributes: Tunnel Number, Tunnel BW, Tunnel ID	**Backup Tunnel Identifications:** Tunnel Number, Tunnel ID	**Backup Tunnel Path:** $\boldsymbol{N_1\ N_2\ N_3\ N_4\ \ldots\ldots\ldots N_k}$

Fig. 6-21 Structure of the controller backup tunnel creation message

***N_1*:** The address of the Point of Local Repair (PLR) router.
***N_k*:** The address of the Merger Point (MP) router.
Both ***N_1*** and ***N_k*** are routers that lie on the primary MPLS tunnel path.
N_2*, *N_3*, *N_4*, ……. *N_{k-1} represent routers that lie apart from the primary MPLS tunnel path.

The *Controller backup tunnel creation* message is created at the *Developed RSVP* model of the controller. It gets its information from the *Fast Re-Route* model of the controller. Then, it is encapsulated within IPv4 datagram, transmitted across the network and received at the *Developed RSVP* model of the merge point router. When receiving the *Controller backup tunnel creation* message, the merge point router extracts the identifications and the attributes of the backup tunnel, which are represented by the backup path, backup tunnel number and the number of the primary (protected) MPLS tunnel. It stores the data of the backup tunnel in a container at its *Developed MPLS* model and to be used when necessary.

The merge point router contacts the penultimate backup tunnel router by creating a *Router backup tunnel creation* message, which has the same frame of the *Controller backup tunnel creation* message. The penultimate router extracts and stores the backup tunnel identifications at a container in its *Developed MPLS* model as well.

This operation is repeated by each router along the backup path until reaching the point of local repair (PLR) router where the establishment of the backup tunnel completes. Unlike the other routers of the backup path, the PLR stores the information of the backup tunnel in passive condition at specific container inside its *Developed LIB* table model. When detecting local link failure via the primary MPLS path, the PLR swaps the label of the packets with that of the backup tunnel and forwards to the next router of the backup path (*N_2*).

6.12 Backup Paths Inspection Scenario

Checking the validity of the backup tunnels and the residual bandwidth across them is the responsibility of the routers along both the primary and the backup paths. The involved routers perform the backup paths inspection scenario regularly to verify the validity of the previously established backup tunnels in terms of survival and the residual bandwidth availability.

The backup paths inspection scenario is according to the following steps:

- Periodically, each PLR creates a *Backup scout* message that contains the protected tunnel's ID, flow *source-destination* pair and the backup path. It sends the *Backup scout* message to the MP router. The *Backup scout* message is created at the *Operation* model of the router. The structure of the *Backup scout* message is as shown in Fig. 6-22 below.

Backup Tunnel Identifications: Tunnel Number, Tunnel ID	**Backup Tunnel Path:** $\boldsymbol{N_1\ N_2\ N_3\ N_4\ \ldots\ldots\ldots N_k}$

Fig. 6-22 Structure of the Backup scout message

***N_1*:** The address of the Point of Local Repair (PLR) router.

***N_k*:** The address of the Merger Point (MP) router.

Both ***N_1*** and ***N_k*** are routers that lie on the primary MPLS tunnel path.

N_2*, *N_3*, *N_4*, ……. *N_{k-1} represent routers that lie apart from the primary MPLS tunnel path.

- Upon receiving the *Backup scout* message, the MP extracts the information related to the backup tunnel that the PLR enquires about. It replies by creating an *Inform backup path status* message and sending it back to the PLR. The *Inform backup path status* message passes inversely through the backup path and collects the path's information via each router it passes through. This message is created at the *Operation* model and its structure is as shown in Fig. 6-23 below.

Backup Tunnel Attributes: Tunnel Number, Tunnel ID	**The least Residual Bandwidth recorded via the Backup path**	**Backup Tunnel Path:** $\boldsymbol{N_1\ N_2\ N_3\ N_4\ \ldots\ldots\ldots N_k}$

Fig. 6-23 Structure of the Inform backup path status message

***N_1*:** The address of the Point of Local Repair (PLR) router.

***N_k*:** The address of the Merger Point (MP) router.

Both ***N_1*** and ***N_k*** are routers that lie on the primary MPLS tunnel path.

N_2*, *N_3*, *N_4*, ……. *N_{k-1} represent routers that lie apart from the primary MPLS tunnel path.

The least residual bandwidth recorded via the Backup path: The least residual bandwidth across the path of the currently tested backup tunnel.

- The *Inform backup path status* message moves inside the network according the OSPF/IP routing table.
- Each router via the backup path compares the residual bandwidth on the link to the next router via the backup path with that obtained from the received *Inform backup path status* message. It chooses the least value obtained from the comparing operation. It also deletes the received *Inform backup path status* message and creates another *Inform backup path status* message, fills **the least residual bandwidth recorded via the Backup path** field of the message with the least residual bandwidth value obtained from the comparing operation and sends it to the previous router across the backup path.
- If there is a link failure via the backup path, the router that is directly connected to the failed link drops the *Inform backup path status* message. The router performs this procedure at its *Developed IP* model.
- This operation is repeated until reaching the PLR router, which upon receiving the *Inform backup path status* message guarantees the existence of the backup tunnel and knows the least residual bandwidth along the backup path.
- If the PLR did not receive the *Inform backup path status* message within specific period of time after sending the *Backup scout* message, this means that either there is a link failure via this backup path or it is not currently suitable to be used for local restoration.
- In the case of not receiving the *Inform backup path status* message or the least residual bandwidth of the backup path obtained from the received *Inform backup path status* message is small and critical value, the PLR adopts the second backup path as first choice of forwarding in the case of link failure and requests the controller to find another backup tunnel (if available).
- If there is only one backup tunnel and it is not currently available, the PLR uses *prior hop protection scenario* as will be explained in **section 6.13.3** of this chapter. The backup paths’ inspection scenario is as shown in Fig. 6-24.
- This procedure checks the errors of the system as well.

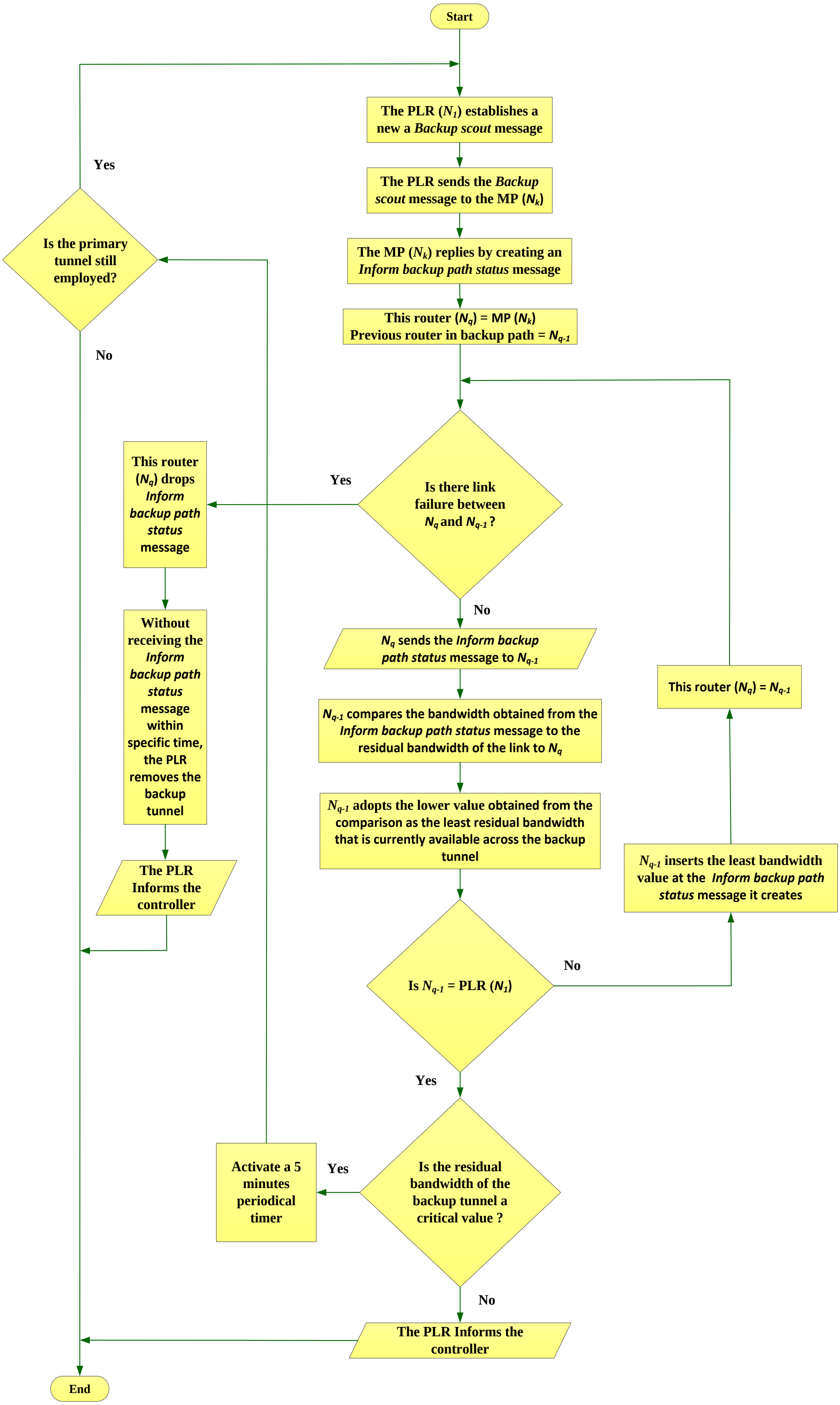

Start
The PLR (N_1) establishes a new a *Backup scout* message
The PLR sends the *Backup scout* message to the MP (N_k)
The MP (N_k) replies by creating an *Inform backup path status* message
This router (N_q) = MP (N_k)
Previous router in backup path = N_{q-1}
Is there link failure between N_q and N_{q-1} ?
Yes
This router (N_q) drops *Inform backup path status* message
Without receiving the *Inform backup path status* message within specific time, the PLR removes the backup tunnel
The PLR Informs the controller
No
N_q sends the *Inform backup path status* message to N_{q-1}
N_{q-1} compares the bandwidth obtained from the *Inform backup path status* message to the residual bandwidth of the link to N_q
N_{q-1} adopts the lower value obtained from the comparison as the least residual bandwidth that is currently available across the backup tunnel
Is N_{q-1} = PLR (N_1)
No
N_{q-1} inserts the least bandwidth value at the *Inform backup path status* message it creates
This router (N_q) = N_{q-1}
Yes
Is the residual bandwidth of the backup tunnel a critical value ?
Yes
Activate a 5 minutes periodical timer
Is the primary tunnel still employed?
Yes
No
No
The PLR Informs the controller
End

Fig. 6-24 Periodical backup paths inspection scenario

6.13 Network Behavior during the Link Failure Event

As mentioned before, a backup tunnel consists of a point of local repair (PLR), one or more intermediate routers and a merge point (MP). The PLR router at the event of link failure swaps the label of the MPLS packets of a flow that was passing through the failed link with that of the backup tunnel. It invokes the backup tunnel number from its *Developed LIB table* model.

The PLR forwards the packets to the output interface that leads to the next router along the backup path instead of the primary path. The intermediate routers along the backup tunnel continue forwarding the MPLS packets according to the backup tunnel number without swapping with different number. Finally, the merge point swaps the backup tunnel number of the MPLS packets of the flow with the primary MPLS tunnel number and continues forwarding according to the primary MPLS tunnel number and across the primary tunnel path.

At the event of link failure, there is no need to involve the higher layers to activate the backup paths. The *Developed MPLS* model (layer 2.5) handles the situation. After updating the routing tables of the routers according to the new topology and informing the controller about the new situation, the controller commences the establishment operation of a new primary tunnel as mentioned in **chapter 5** before. The path of the new tunnel avoids the failed links. The MPLS packets of the diverted flow commence passing through the path of this new established tunnel.

The behavior of the router at the PLR depends on the previously stored backup tunnels and the real time updates related to the validity and the residual bandwidth along those backup tunnels. The following cases represent the PLR behavior:

- There is only one previously established backup path.
- There are two previously established backup paths.
- There is none previously established backup path.

6.13.1 There is only One Previously Established Backup Path.

Unless there is a link failure on the path of the backup tunnel, the point of local repair forwards the traffic of the disconnected MPLS path to the pre-established backup path until establishing the new primary tunnel.

Example, suppose the 13 routers network of Fig. 6-10 where the link that connects router *9* with router *11* goes down suddenly. Router *9* identifies the paths of the primary tunnels passing through that link. The path of tunnel ***4* → *11*** passes through routers: ***4* –**

1 – 6 – 9 – 11. It passes through the failed link: ***9 – 11*** and it has one backup path that passes through: *9 – 8 – 7 – 10 – 13 – 12 - 11*.

Router *9* is the point of local repair. It activates the backup tunnel by considering router *8* as the next router of forwarding the packets of flow ***4 – 11*** instead of router *11*. It swaps the label of the primary tunnel on the MPLS packets with the label of the backup tunnel during the forwarding operation. Router *8* receives the MPLS packets of flow ***4 - 11*** and identifies them from the labels they have.

It forwards them to the next router of the backup path, *7*, with the same label. The operation is repeated until reaching the merge point, router *11*. Router *11* is also the tail-end router. It removes the label from the packets and carries on forwarding them to their last destination (the directly connected host **11**).

After updating the routing tables of the routers according to the new topology, the controller establishes a new primary tunnel ***4* → *11***. The new tunnel avoids the failed link (*9 - 11*) by passing through routers: ***4 – 2 - 10 – 13 – 12 – 11***.

When the tunnel establishment operation finishes, the network depends the new tunnel as the primary path of forwarding the packets of flow **4 – 11**. The new tunnel differs from the old one in terms of path and the identification numbers. The latency of flow **4 – 11** during the simulation of *9 – 11* link failure is as shown in Fig. 6-25. Later, the controller removes the old primary tunnel.

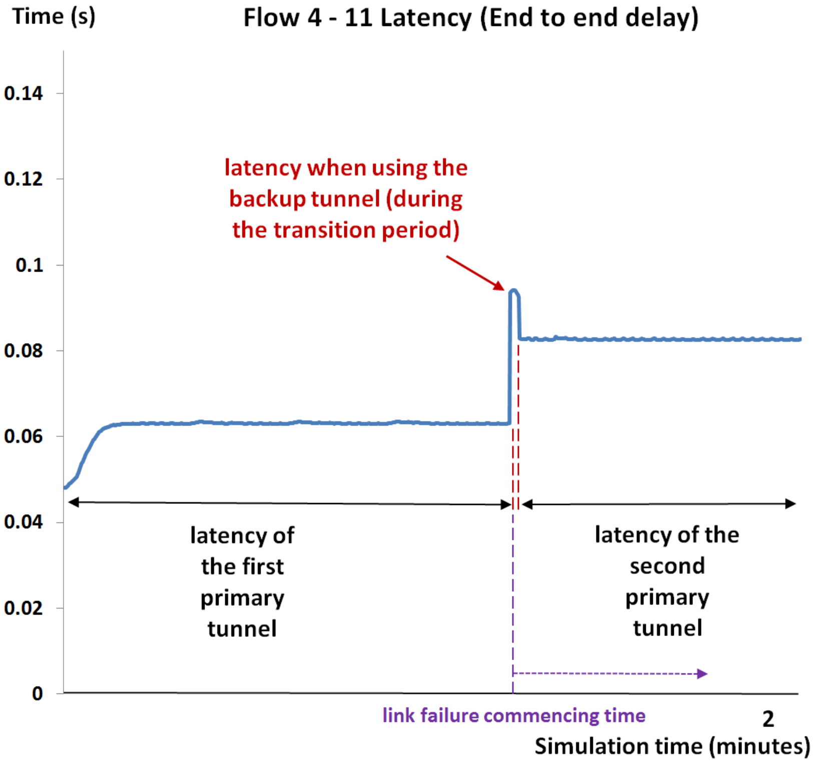


Fig. 6-25 Flow 4 - 11 latency during the simulation of link failure between routers 9 – 11

6.13.2 There are Two Previously Established Backup Paths

The point of local repair depends on the final value of the residual bandwidth on the backup paths obtained from the last received *Backup path status* message. If the residual bandwidth of the first backup path is not enough to handle the flow of the primary path or the *Backup path status* message has not been received, the point of local repair router adopts the second backup path through which to forward the flow at the event of link failure. If none of the backup paths can handle the flow, the point of local repair asks the controller to establish a third backup tunnel whose path should avoid the existing ones.

Example, suppose the 13 routers network of Fig. 6-10 where there is previously established primary tunnel ***4*** → ***11***. This tunnel passes through routers: ***4 – 1 – 6 – 9 – 11***. The link that connecting router ***1*** with router ***6*** goes down suddenly. This link is protected by two backup tunnels. The path of the first backup tunnel is: *1 – 5 – 6* and the path of the second one is: *1 – 3 – 8 – 9*.

Within a short time before the link failure event takes place, router ***1*** sent two *Backup scout messages* to check the validity of both backup tunnels. Router ***1*** received only one *Backup Path Status message* that indicates the validity of the backup tunnel: *1 – 5 – 6* in terms of existence and residual bandwidth. Router ***1*** did not receive any *Backup Path Status message* that indicates the validity of the backup tunnel: *1 – 3 – 8 – 9* because of a problem somewhere at that backup tunnel. Router ***1*** adopts the backup tunnel: *1 – 5 – 6* as a unique backup tunnel.

At the event of link failure between router ***1*** and router ***6***, router ***1*** immediately diverts flow **4 – 11** through the backup path *1 – 5 – 6*. It swaps the label of the primary path with the label of the backup path. When the packets of the flow reach the merger point (Router ***6***), it returns the original label of the primary tunnel and carries on forwarding the packets of the flow as usual.

The controller establishes a new primary MPLS tunnel for flow **4 - 11**. The path of the new primary MPLS tunnel passes through nodes: ***4 – 5 – 6 – 9 – 11***. The latency of flow **4 - 11** during the simulation of *1 – 6* link failure is as shown in Fig. 6-26. Later, the controller removes the old primary tunnel.

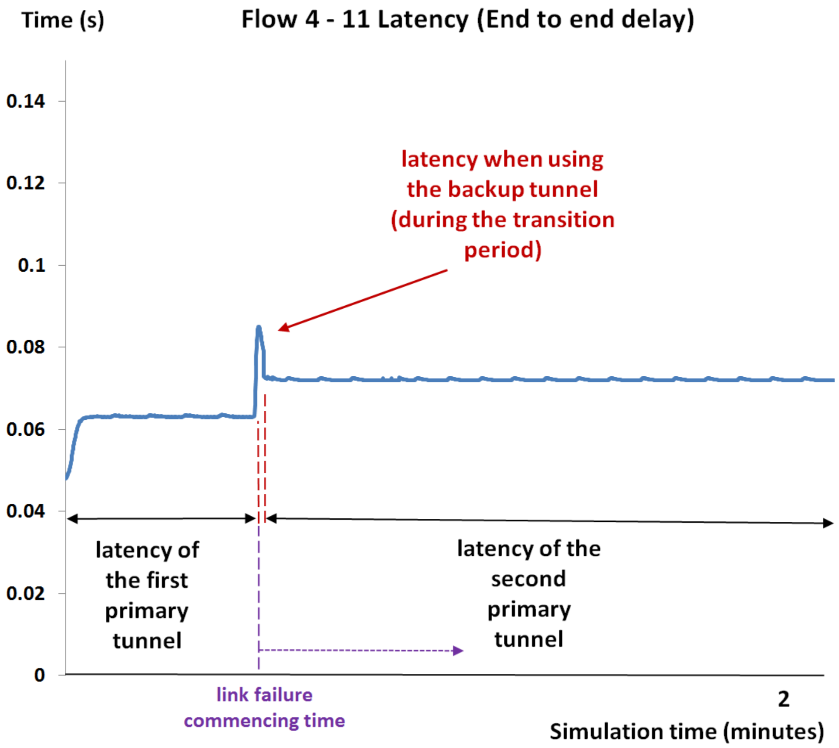


Fig. 6-26 Flow 4 – 11 latency during the simulation of link failure between routers 1 – 6

6.13.3 There is None Previously Established Backup Path (Prior Hop Protection)

The controller trends to protect all the links of all primary tunnels. It tries to find two backup tunnels or at least one backup tunnel to protect each link. The controller is also constrained by the availability of resources represented by the network topology. Some links cannot be protected as there are no available backup paths to protect them. Those links are denoted as non-locally protected links. The controller includes the protection of the non-locally protected links within the protection of their prior protected links. The router that is considered as a PLR uses a backup tunnel of *partial detour* path (or *full detour* path) that skips the non-locally protected link.

During the event of link failure of non-locally protected link, the router that is directly connected to the failed link immediately sends a *Link failure of none backup* message to the previous router. The *Link failure of none backup* message contains the addresses of the routers which should not be merge points, the address of the failed link and the identifications of the tunnels passing through the failed link and do not have any backup paths. The *Link failure of none backup* message is generated at the *Operation* model of the router and its structure is as shown in Fig. 6-27.

List of routers to be avoided from being merge points	**Failed Link Address**	**Identification numbers of tunnels that do not have backup paths**

Fig. 6-27 Structure of the Link failure of no available backup paths message

The router that precedes the router where link fails forwards the *MPLS Data* packets of the disconnected tunnel to the backup path that can skip the upcoming router of the failed link (if there is any). The router that precedes the router where link fails (in MPLS tunnel path sequence) is considered as the point of local repair.

If the previous router does not have any backup tunnel of path that bypasses the failed link, it adds its address to the **List of routers to be avoided from being merge points** field of the *Link failure of none backup* message and forwards it as well as the *MPLS Data* packets of the flow to its prior router. This operation is repeated until finding the router of the backup path that bypasses the failed link. Generally, any router along the primary MPLS path can be used as PLR if its backup paths can bypass the failed link.

Example, suppose the 13 routers network of Fig. 6-10 where the link that connecting routers ***6*** and ***9*** goes down suddenly. Router ***6*** identifies the paths of tunnels passing through that link. The path of tunnel ***4* → *11*** passes through that failed link. Unfortunately, there is no backup tunnel to locally protect this tunnel at router ***6***. Thus, router ***6*** immediately informs the previous router, router ***1***, through a *Link failure of none backup* message that it suffers from a link failure on the path of the tunnel ***4* → *11*** and it has no backup path to protect this tunnel. It inserts its address into the *List of routers to be avoided from being merge points* of the message.

Router ***1*** receives the *Link failure of none backup* message from router ***6***. Router ***1*** has two previously established backup paths that locally protect the primary tunnel ***4* → *11***. The first backup path is: *1 – 5 – 6* and the second is: *1 – 3 – 8 – 9*. Router ***1*** compares the routers of the paths of its backup tunnels with the **List of routers to be avoided from being merge points** obtained from the received *Link failure of none backup* message (router ***6***).

The path of the first backup path protects link ***1* – *6*** but it does not protect link ***6* – *9*** as it ends at router ***6***. Therefore, using it will not solve the problem. The second backup tunnel ends at router ***9***. It protects both links: ***1* – *6*** and ***6* – *9***. Router ***1*** adopts it as a temporary backup tunnel until establishing the new primary tunnel path.

Router ***1*** swaps the label of the primary tunnel on the MPLS packets with the label of the backup tunnel during the forwarding operation. The intermediate routers of the backup tunnel (***3*** and ***8***) continue forwarding the packets of the flow according to the backup tunnel number. Router ***9*** as a merge point swaps the label of the backup tunnel on the packets with the label of the primary tunnel during the forwarding operation.

After updating the routing tables to match the new topology and informing the controller all the information it needs, the controller establishes a new primary MPLS tunnel which in this case, takes the place of the current temporary used path. The path of the new established tunnel passes through routers: ***4 – 1 – 3 – 8 – 9 – 11.*** It is identical to the previous primary tunnel plus the used backup tunnel used to deal with ***6 – 9*** link failure. The latency of flow **4 - 11** during the simulation of *6 - 9* link failure is as shown in Fig. 6-28. Later, the controller removes the old primary tunnel.

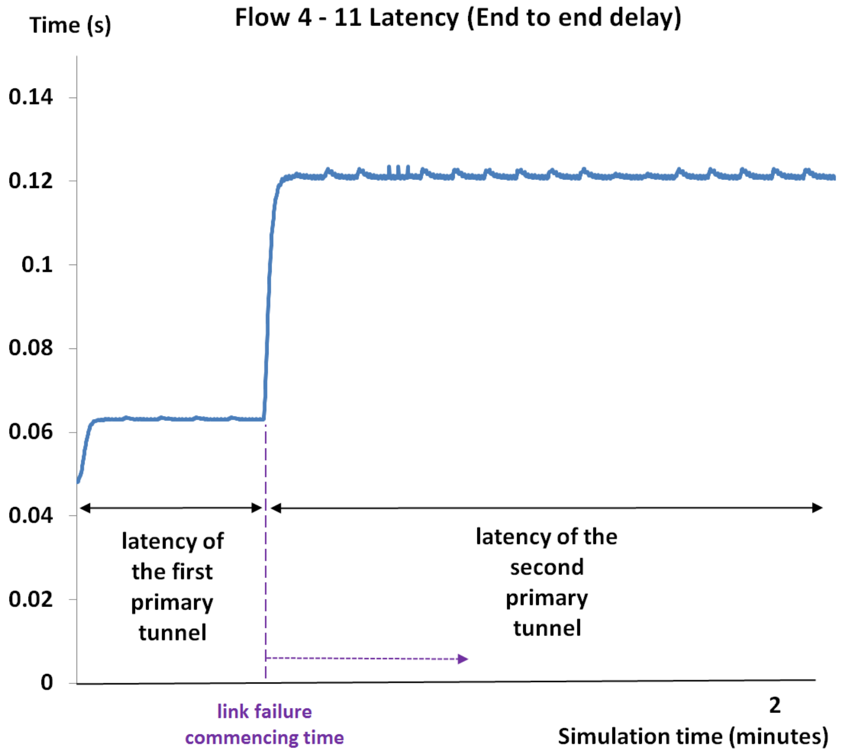


Fig. 6-28 Flow 4 – 11 latency during the simulation of link failure between routers 6 – 9

6.14 Removing the Old Primary Tunnel and Its Backup Tunnels

After establishing the new primary MPLS tunnel, routers commence using it instead of the old MPLS tunnel that suffered from the link failure. The controller removes the old MPLS tunnel by using the MPLS tunnel removal procedure explained in **section 5.11.3** of **chapter 5**.

After removing the primary MPLS tunnel, there are neither *Backup scout* messages nor *Inform backup path status* messages issued regarding the backup tunnels that protect the removed primary tunnel. Without receiving those messages within specific time, the intermediate routers of the backup paths delete the identifications and the attributes of the backup paths. Later, the controller can use those identifications when establishing new primary or backup tunnels.

6.15 Bandwidth Allocation in the Network of 18 Routers

In this network of 18 routers of Fig. 6-29, the controller establishes several primary MPLS tunnels and their backup tunnels. It shows the efficiency in bandwidth allocation of the backup tunnels. Each router is connected to a host that has the same number.

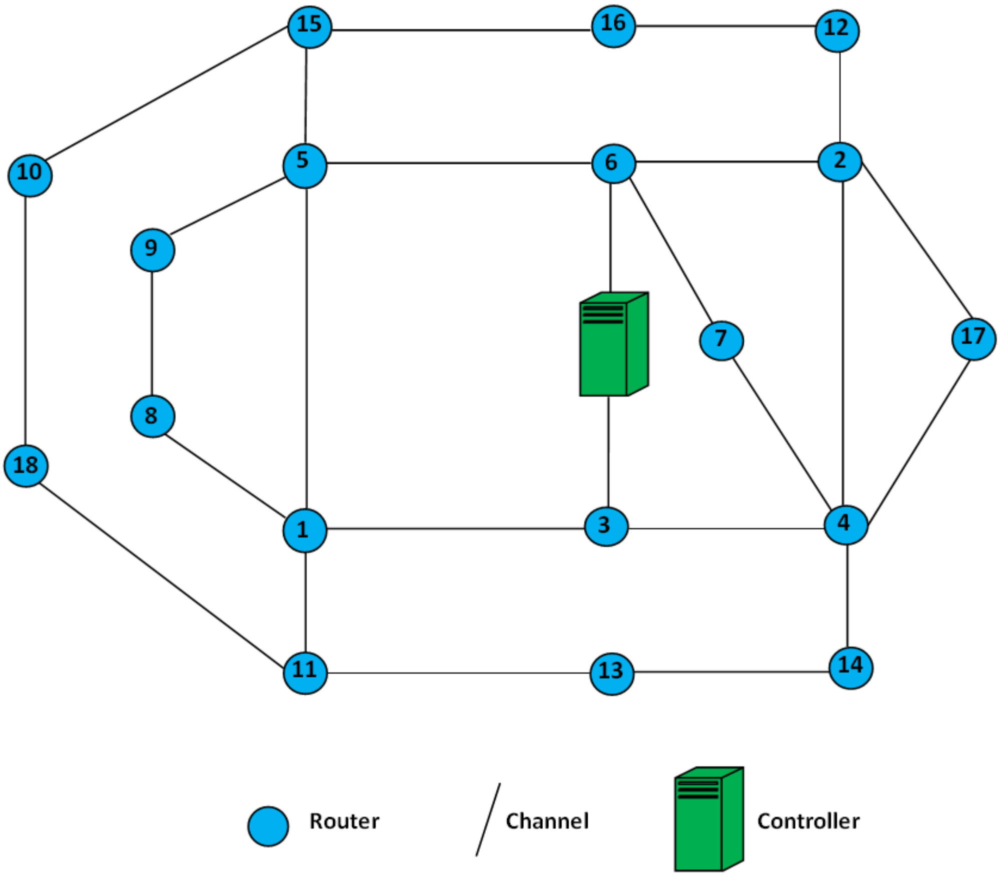


Fig. 6-29 Topology of 18 routers network

The controller establishes the primary MPLS tunnels when they are due. After a while, it uses its Backup Paths Calculation Algorithm (BPCA) to calculate the paths of their backup tunnels. Neither high load nor link failure has been applied on this network. The primary MPLS tunnels that are protected are:

- ***11-13-14.***
- ***1-3-4-2.***
- ***11-1-5-15.***
- ***9-5-6-2-12.***
- ***2-4-3-1-11-18.***
- ***10-15-5-6-2-17.***

All flows using those tunnels have the same data rate value of 3 Mbps. Table VI-IV shows the paths of those primary tunnels under normal load and the paths of their backup tunnels calculated by the controller.

TABLE VI-IV

THE PRIMARY TUNNELS AND THEIR BACKUP MPLS TUNNELS OF 18 ROUTERS NETWORK

Primary MPLS Tunnel Path (Routers)	Links of the primary tunnel	Number of Calculated Backup Tunnels	Paths of Backup MPLS Tunnels (Routers)	Type of the Backup Tunnel
11-13-14	*11-13*	*2*	*11-1-3-4-14*	*Full detour*
			11-18-10-15-5-1-3-4-14	*Full detour*
	13-14	*0*	*None*	*Previous node protection*
1-3-4-2	*1-3*	*2*	*1-5-6-2*	*Full detour*
			1-11-13-14-4	*Partial detour*
	3-4	*0*	*None*	*Previous node protection*
	4-2	*2*	*4-17-2*	*One link protection*
			4-7-6-2	*One link protection*
11-1-5-15	*11-1*	*2*	*11-18-10-15*	*Full detour*
			11-13-14-4-2-12-16-15	*Full detour*
	1-5	*2*	*1-8-9-5*	*One link protection*
			1-3-4-2-12-16-15	*Partial detour*
	5-15	*1*	*5-6-2-12-16-15*	*One link protection*
9-5-6-2-12	*9-5*	*1*	*9-8-1-3-4-2*	*Partial detour*
	5-6	*2*	*5-15-16-12*	*Partial detour*
			5-1-3-4-2-12	*Partial detour*
	6-2	*0*	*None*	*Previous node protection*
	2-12	*1*	*2-4-14-13-11-18-10-15-16-12*	*One link protection*
2-4-3-1-11-18	*2-4*	*2*	*2-5-6-1*	*Partial detour*
			2-12-16-15-10-18	*Full detour*
	4-3	*2*	*4-14-13-11*	*Partial detour*
			4-7-6-5-1	*Partial detour*
	3-1	*0*	*None*	*Previous node protection*
	1-11	*2*	*1-5-15-10-18*	*Partial detour*
			1-8-9-5-15-10-18	*Partial detour*
	11-18	*0*	*None*	*Previous node protection*
10-15-5-6-2-17	*10-15*	*1*	*10-18-11-13-14-4-17*	*Full detour*
	15-5	*1*	*15-16-12-2*	*Partial detour*
	5-6	*2*	*5-1-3-4-17*	*Partial detour*
			5-8-9-1-3-4-17	*Partial detour*
	6-2	*1*	*6-7-4-17*	*Partial detour*
	2-17	*2*	*2-4-17*	*One link protection*
			2-6-7-4-17	*One link protection*

From Table VI-IV, some paths of the backup tunnels are very long. Forwarding flows through such paths during the link failure may cause packets to drop before reaching their destinations. Therefore, the number of hops of the backup path should be less than the Time to live (TTL) of the flows' *MPLS Data* packets. In the case of backup path of hops number bigger than the TTL value, the controller should not adopt that backup path and activates the *Prior Hop Protection* procedure of **section 6.13.3** instead.

Figs. 6-30, 6-31, 6-32, 6-33 and 6-34 represent the allocation and the inter-demand bandwidth sharing between the backup tunnels that protect each of the following primary tunnels shown in table VI-IV: ***1* → *2*, *11* → *15*, *9* → *12***, ***2* → *18*** and ***10* → *17*.**

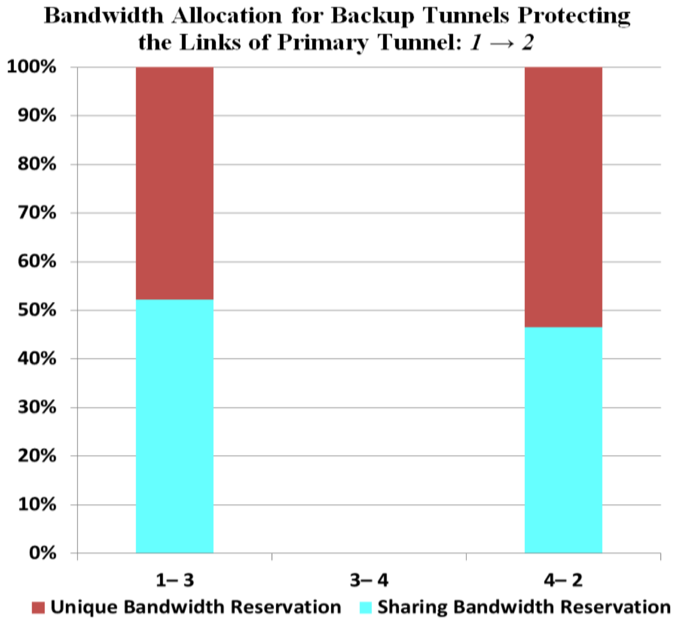


Fig. 6-30 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel *1* → *2*

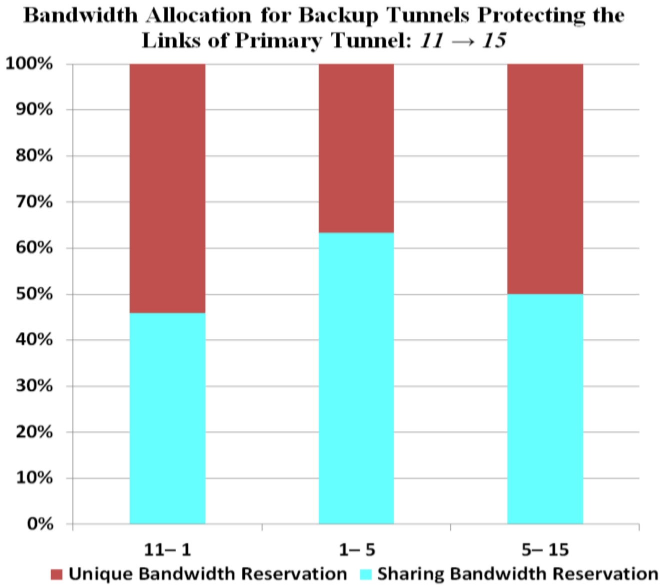


Fig. 6-31 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel *11* → *15*

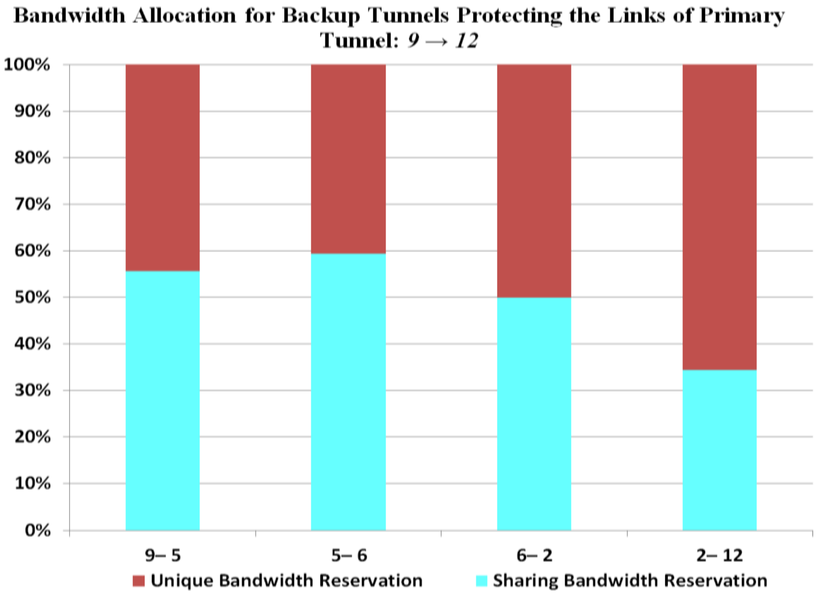


Fig. 6-32 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel *9 → 12*

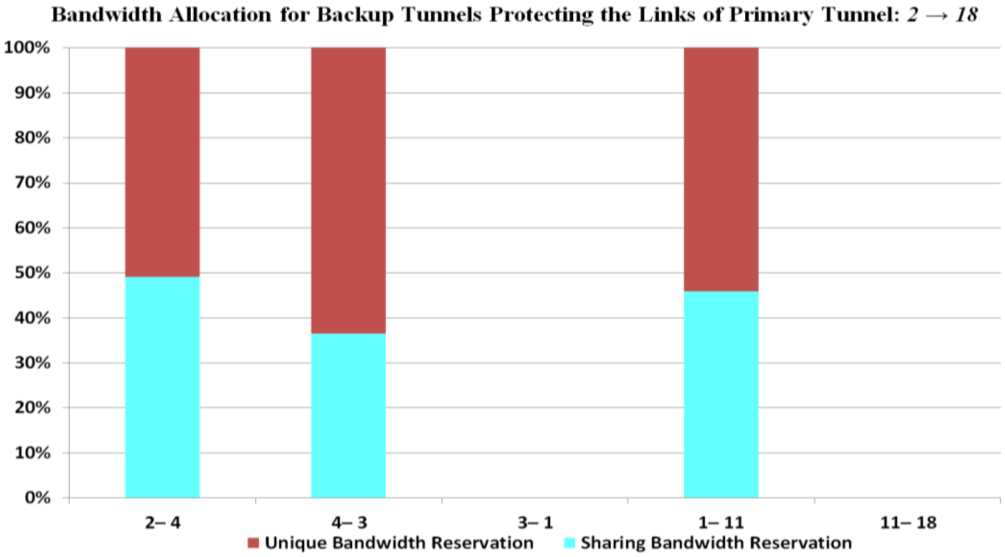


Fig. 6-33 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel *2 → 18*

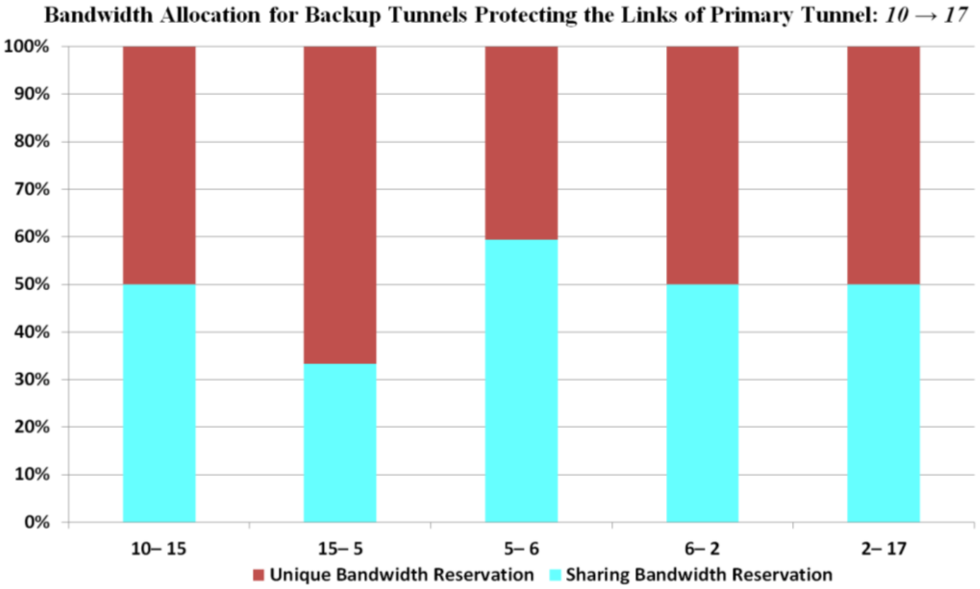


Fig. 6-34 Bandwidth allocation and sharing of backup tunnels protecting primary tunnel *10 → 17*

6.16 Conclusions Regarding Improving the FRR Mechanism

The basic idea of this chapter is using the controller as central manager that calculates the dynamic backup tunnels taking into consideration the bandwidth sharing issue in an efficient way. This decreases the complexity of the operation and provides higher level of protection. The controller provides the network with a global resources optimization by higher bandwidth saving. The more the bandwidth sharing among different backup tunnels results in the higher the bandwidth saving and the better the usage of the network resources.

The bandwidth, which allocated at a link to serve several backup tunnels that protect multiple links belong to same primary tunnel, is reserved only once and it is equal to the bandwidth of the primary tunnel. The controller that is involved in allocating the identification numbers to both the primary and the backup tunnels provides the tunnels with global identification numbers.

The calculation of the backup tunnels is affected by the paths of the previously established primary tunnels as well as the availability of the resources (network topology, number of routers and links available bandwidth). However, increasing the number of the primary tunnels increases inter-demand bandwidth sharing among their backup tunnels. There is no bandwidth sharing between the primary tunnels and the backup tunnels.

Some of the backup tunnels can be ***full detours*** that protect all the links of a MPLS primary tunnel; some others are ***partial detours*** that protect more than one link while the rest represent one link protectors. When a link failure event occurs, there is no possibility of congestion occurring at the backup tunnels. The calculated backup tunnels are also loop free.

Generally, there is also higher inter-demand bandwidth sharing among the backup paths if the protected primary tunnels are more convergent in their bandwidth values.

Chapter 7

Improvement of Performance of the EIGRP Algorithm by Adopting the SDN Notion

Introduction to Chapter 7

This chapter describes the theory of the EIGRP including the problems facing this technology and our proposed solution. Then, it presents our simple design to build a simple EIGRP network. Later, it illustrates the methodology of our approach to improve the performance of the EIGRP network and finally, it shows the results obtained from our proposed approach.

7.1 EIGRP Theory

The EIGRP (Enhanced Interior Gateway Routing Protocol) is an enhanced version of the Interior Gateway Routing Protocol IGRP [3] pp 232. The EIGRP develops the capabilities of link-state protocols into distance vector protocols [18]. The Diffusing Update Algorithm (DUAL) is the convergence algorithm integrated into the EIGRP [114]. It enables EIGRP routers to find whether a path advertised by a neighbour is loop-free or looped. It also permits a router running EIGRP to find alternative paths without waiting on updates from other routers [18].

The EIGRP router does not send periodic updates. Instead, it sends partial updates only when there is change in the *metric* of a route [18].

7.1.1 EIGRP-Based Algorithm

The multi-path discovery process depends on the EIGRP-based algorithm that is designed and embedded inside each router. Each router must have at least one path to every destination. The EIGRP-based algorithm depends on both the available bandwidth and the delay according to the following expression of the default behavior:

$$Metric = \left(\frac{10^7}{Bandwidth} + Delay\right) * 256 \qquad (2)$$

In (2), *Bandwidth* is the least bandwidth (measured in kilobits/ second) of all the outgoing links to the destination, while *Delay* represents the sum of the delays (measured in 10s of microseconds) configured on the interfaces, links, and hops on the path to the destination [115].

7.1.2 EIGRP Concepts

The EIGRP-based algorithm has been used because it can find several paths to the destinations. The route that has the least *metric* value is called the *successor*, and is considered to be the best route to the destination (the *successor* has the highest bandwidth and the lowest delay). The neighbour routers that have an advertised *metric* less than the *metric* of the current routing table are called *feasible successors* [32]. Theoretically, for every destination there are only one *successor* and up to six *feasible successors* [14] pp 420.

A router running EIGRP stores all its neighbours' routing tables so that it can quickly adapt to alternate routes. If no appropriate route exists, EIGRP queries its neighbours to discover an alternate route. These queries propagate until an alternate route is found. To provide superior routing performance, EIGRP employs four key technologies that combine to differentiate it from other routing technologies: neighbour discovery/recovery, Reliable Transport Protocol (RTP), DUAL finite-state machine, and protocol-dependent modules. Reliable Transport Protocol (RTP) is responsible of guaranteed, ordered delivery of EIGRP packets to all neighbours [18].

7.2 Research Problems

As far as the traditional routing protocol network is concerned, two kinds of congestions occur namely: the local congestion and the remote congestion [5]. The EIGRP supports unequal cost path load balancing among the feasible successor paths depending on the *metric* value of the path [33]. This load balance operation has been applied without referring to the remote congestion. Furthermore, it may not be necessary to split the flow if it has a very low data rate and the *successor* path fulfills the forwarding requirements. If a flow is split somewhere and recombined at another place, the sequence order of packets will change. Thus, our algorithm avoids flow split as much as possible and takes in consideration the remote congestion problem. In addition to that, load balance in EIGRP network cannot be applied over a path if the path is not a *feasible successor* [33]. Our proposed algorithm enables using any path to avoid congestion whether the path is a *feasible successor* or not.

7.3 Research Objectives

The aim of this chapter is to enhance the performance of a traditional Enhanced Interior Gateway Routing Protocol (EIGRP) algorithm by adding some features of the Software Defined Networking (SDN) in a modified approach to improve some network's performance metrics like link congestion and packet loss. The improvement of those metrics leads to an improvement in the total throughput of the network as well as QoS.

This approach results in a network that contains an intelligent real time dynamic supervisory controller, which is capable of detecting the location of a congestion that may take place before or at the brink of its occurrence and dealing with selected flows on selected routers across the network in a way that prevents the congestion by using a smart heuristic congestion avoidance and routing algorithm thereby reducing the generation of *Control* messages as much as possible.

7.4 Research Objectives

The aim of this chapter is to enhance the performance of a traditional Enhanced Interior Gateway Routing Protocol (EIGRP) algorithm by adding some features of the Software Defined Networking (SDN) in a modified approach to improve some network's performance metrics like link congestion and packet loss. The improvement of those metrics leads to an improvement in the total throughput of the network as well as QoS.

This approach results in a network that contains an intelligent real time dynamic supervisory controller, which is capable of detecting the location of a congestion that may take place before or at the brink of its occurrence and dealing with selected flows on selected routers across the network in a way that prevents the congestion by using a smart heuristic congestion avoidance and routing algorithm thereby reducing the generation of Control messages as much as possible. This chapter represents a supplementary work. There is no full contribution in it regarding this research as its basic idea is similar in many aspects to that discussed in **chapter 4**. The differences are in methodology and in precedence. However, it can be considered as the first attempt to improve of the performance of the EIGRP in terms of congestion control by adopting the SDN notion.

7.5 Related Work Regarding EIGRP Technology

The development efforts regarding the EIGRP are very few and very limited. Most of publications regarding the EIGRP concentrate on comparing its performance with the other existing routing protocols [77, 116, 117]. Very few real development efforts are published. Here, the most notable developments published on the *IEEE explore* web site are reviewed.

In [118], Zhao et al introduce some modification to the Diffusing Update Algorithm (DUAL) where the amount of distributed computation will be reduced and time spent on convergence is expected to be cut down. The modified DUAL will be loop-free and can converge faster than the original one.

The calculation of *metric* for the EIGRP protocol, which takes into consideration the information security risks of transit traffic, is proposed in [119]. It is proposed to evaluate the risk of information security using the methodology of the NIST CVSS (National Institute of Standards and Technology - Common Vulnerability Scoring System) standard and the theory of information system's survivability. It performs prioritization and evaluation of traffic confidentiality, as well as mechanisms to assess the state of the network to enable dynamic consideration of score, which is based on the calculation of the route information security risk.

Our approach neither modifies the DUAL nor evaluates the risk of information security. It improves flows distribution by using the controller’s capabilities.

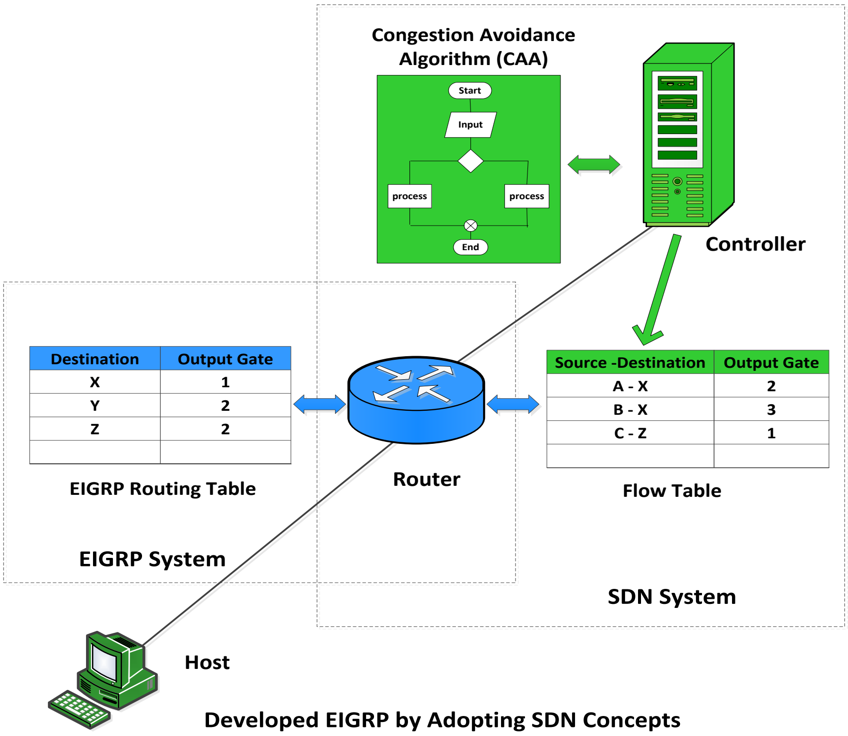
As the idea of this chapter is similar to the idea proposed in **chapter 4**, *OSPF and QoS Development Efforts,* fulfils the requirements of this section as well.

7.6 Network Model Design, Architecture and Features

This section presents the architecture of the designed network. The network has been designed by using OMNeT++ simulator, which depends on both C++ programming language and Network Description (NED) language.

7.6.1 Network Model Design

Our network architecture utilizes traditional network EIGRP routing protocol which is implemented then developed by adopting the notion of SDN network controller. Fig. 7-1 represents this adoption.



Destination	Output Gate
X	1
Y	2
Z	2

Source -Destination	Output Gate
A - X	2
B - X	3
C - Z	1

Fig. 7-1 Network basic design - Developed EIGRP by adopting SDN notion

The network hardware components are:

- Controller.
- Routers.
- Hosts.
- Channels (links).

Our network represents a *flat routing system*; the routers are peer of all others as well as the hosts.

7.6.2 Network Features

- None of the models of the Inet-OMNeT++ project is used in the design.
- The devices are identified by natural numbers instead of the IP addresses. There is only one address that represents each device without using network addresses.

7.6.3 Network Architecture

The OMNeT++ simulator of version 4.3 does not support the EIGRP in any of its projects. However, it provides the capabilities to design any network system. Improving the EIGRP involves designing the basic EIGRP algorithm, then adding the improvement and later comparing the performance of the improved design with that of the basic design. The design of the EIGRP system depends on its attributes published at Cisco Systems Incorporated web sites. The devices of our EIGRP network are constructed to work in one layer instead of the seven layers of the OSI model. The architectures of the devices are as below where Figs. 7-2 and 7-3 show the architectures the controller and the router respectively. Each model performs a specific task.

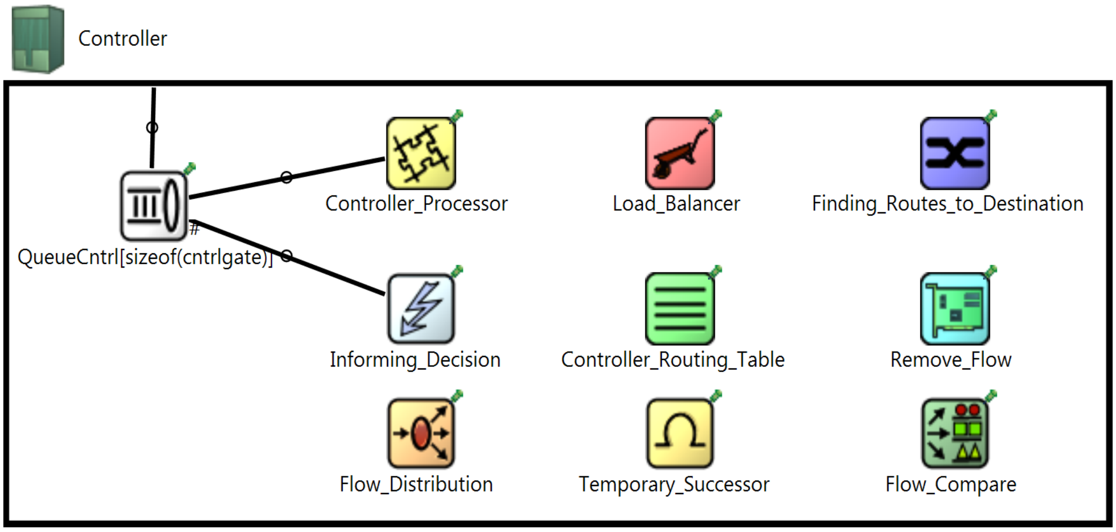


Fig. 7-2 Controller's architecture of one layer EIGRP network

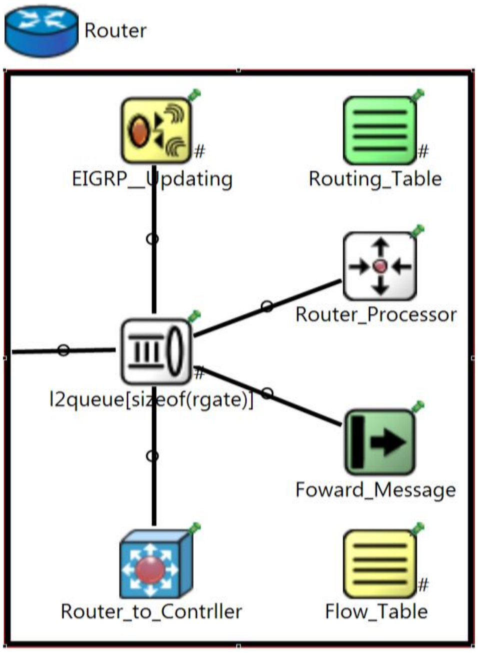


Fig. 7-3 Router's architecture of one layer EIGRP network

7.7 Network Topologies

Our project examines two networks of different topologies. In the first one there are 16 routers and 6 hosts, while in the second there are 10 routers and 5 hosts, as shown in Figs. 7-4 and 7-5 respectively. All devices are connected via channels.

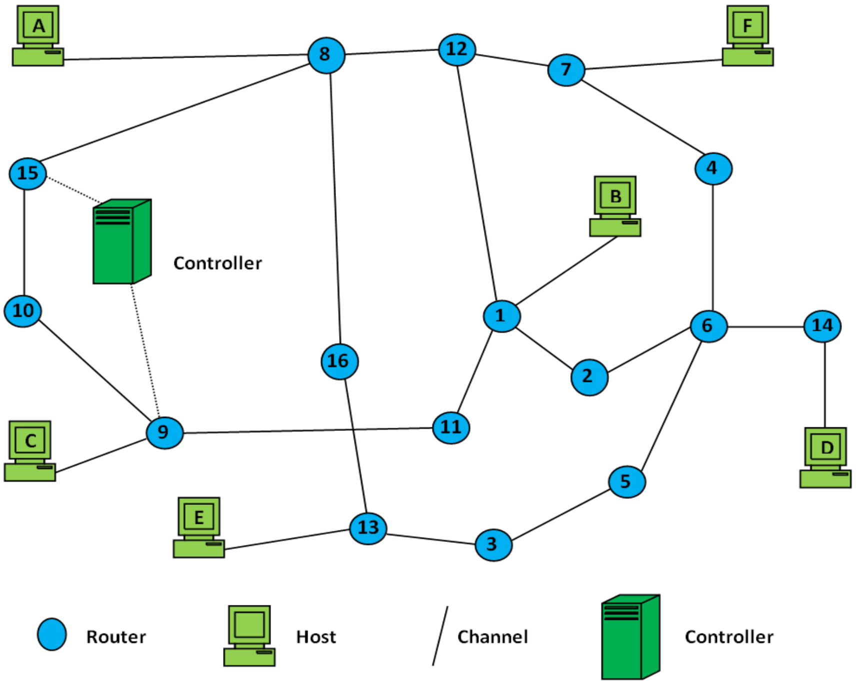


Fig. 7-4 EIGRP network 1 of 16 routers and 6 hosts

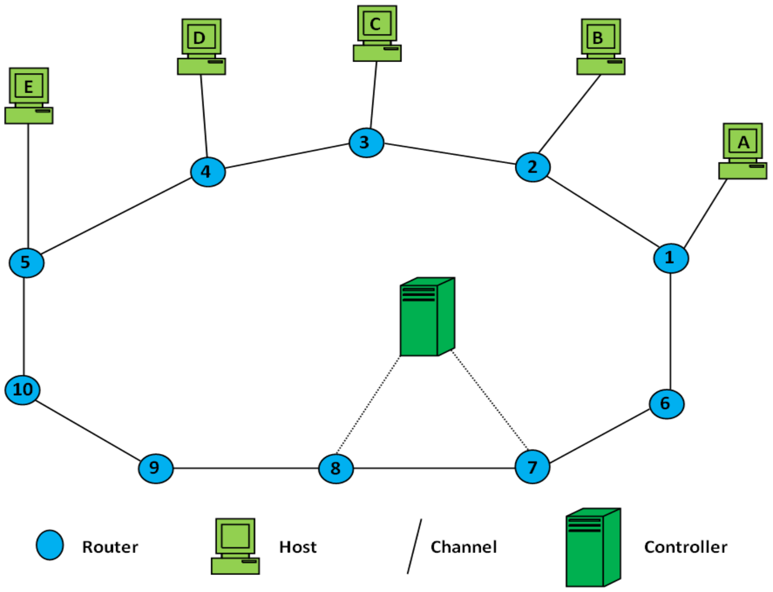


Fig. 7-5 EIGRP network 2 of 10 routers and 5 hosts

Channels are the links that connect the devices with each other. The channels represent general design of the OMNeT++ data rate that contains bandwidth and delay. The used channels do not represent any of the standard connections like (Ethernet or serial). The topologies are still fixed during simulation as there is no link failure event occurs.

7.8 Network Messages and Packets

The messages and packets do not represent any specific type (UDP, TCP….). There are different purpose messages and packets travel across the network:

- *EIGRP standard* messages which include (*Hello, Acknowledgment, Update, Query, and Reply*) [114]. Those messages are used to build the EIGRP routing table inside the router. As our design represents the very simple model of EIGRP, the last two messages were not used in it.
- *Information* messages issued by routers and directed to the controller. The information messages carry the network information like router address, routing tables of each router and its links bandwidth.
- *Flow status updating* messages issued by head-end routers and directed to the controller. They carry flow's current information like flow source, flow destination and flow data rate.
- *Control* messages (including *Approval* messages) issued by the controller and directed to specific routers.
- *Regular updating* messages issued by all routers every several minutes and forwarded to the controller.
- *Acknowledge* messages issued by controller and routers to confirm the arrival of other messages.

  All above mentioned messages are sent to do a specific job.
- The *Data* packets are issued only by hosts and directed to other hosts. *Data* packets represent the flows. They are sent according to specific periodic timing as they have a specific length as well.

The design of this project involves a semi ideal environment. Using the OMNeT++ simulation enables achieving this situation. Only the *Data* packets are designed to have a specific bit length while the other messages do not have any length (zero bits length). Therefore, the recorded throughput considers the *Data* packets only.

7.9 The EIGRP Router

The router in our architecture has two tables, a *traditional EIGRP routing table,* which is built by the router itself through information acquired from other routers and a *flow table* that is received through *Control* messages sent by the controller. The flow table is temporary and has higher priority than the EIGRP routing table. Some flows are forwarded according to the EIGRP routing table whereas other flows are forwarded according to the flow table or both tables. The router architecture is shown in Fig. 7-3. Each router has one address and several interfaces and gates.

7.9.1 Procedure of Construction of EIGRP Routing Table

- At the beginning of the simulation, all routers commence exchanging *EIGRP Hello* messages with their neighbours.
- Each router creates *Hello* messages, inserts its address into them and sends them to all its directly connected neighbours.
- When a host receives the *Hello* message, it creates *HelloReply* message, inserts its address and identity into it and sends it to the router.
- Router receives *Hello* messages from its neighbour routers and *HelloReply* messages from its neighbour hosts (if it is connected to any). From the received *Hello* and *HelloReply* messages, router calculates the channel delay, gets and records types and addresses of all directly connected neighbours in the *neighbour table*. A router has the ability to estimate the bandwidth of its directly connected channels.
- Routers calculate the *metric* to each neighbour according to the bandwidth and the delay of each directly connected channel by using equation (2). A primary routing table of the directly connected devices is constructed.
- Each router inserts the information of its primary routing table represented by minimum bandwidth and total delay to each discovered destination into an *EIGRP update* messages and sends those messages to its neighbour routers.
- Neighbour routers receive the *EIGRP update* messages and extract the routing information from them. According to the extracted routing information, each router uses equation (2) to calculate the *metric* to each available destination and updates its routing table.
- The router creates new *EIGRP update* messages, inserts specific information of its routing table into each of them and sends each message to a specific neighbour router.

- The inserted information represents the bandwidth, the delay and the path of nodes to the destination.

  Path of nodes: represents the *successor* path that flow traverses until reaching the destination. The first address in the path represents the current router address and the last address represents the address of the router that is directly connected to the destination.
- If the destination obtained from the received *EIGRP update* message does not exist in the current routing table, the router updates the routing table by adding it with its *metric* value.
- If the destination obtained from the received *EIGRP update* message already existing in the routing table, the router calculates the *metric* value from the information received within received *EIGRP update* message. Then it compares the *metric* value calculated from received *EIGRP update* message with the existing one that belongs to the same destination existing in its routing table. The router adopts the one with less *metric* value with its path and ignores the one of higher *metric* value.
- The router does not send updates to the direction they came from. This is called *split horizon* [14] pp 382.
- Routing loops problem may occur. Routing loops are prevented by examining the discovered path of routers. A router address must be mentioned only once within the path of routers. Repeated address of the router means that there is loop problem. *Split horizon* is also used to prevent loops [115].
- The creation of *EIGRP update* messages continues as well as *metric* comparison and routing table updating until all destinations are identified in all routers through the less *metric* values. This *metric* value is called the *feasible distance* [115]. The routers then stop creating new *EIGRP update* messages as the routing table creating operation is completed.
- Now, routers send their routing tables to their neighbours to identify the *feasible successors*. The neighbour routers that have an advertised *metric* to a destination less than the *metric* of the current router are recorded as *feasible successors*. This *metric* along a path to a destination is called the *reported distance* [115].
- Inside router, some destinations have only *successor* path while other destinations have one or two *feasible successors* beside the *successor* path.
- The routers deal with the controller during routing table construction operation as a normal host.

7.9.2 Procedure Verification

In this section, there is an example that confirms the validity of our method to calculate the *successor* and the *feasible successor* paths to a specific destination. The example considers the calculations of paths inside router 1. Fig. 7-6 shows our experimental EIGRP network of 16 routers and 6 hosts with some of its links' attributes. The *Switching Delay* of the routers is designed to be *zero* seconds. Only the *Propagation Delay* of the links is considered. The delays of all links are the same and equal to 200µs regardless the bandwidth of the link.

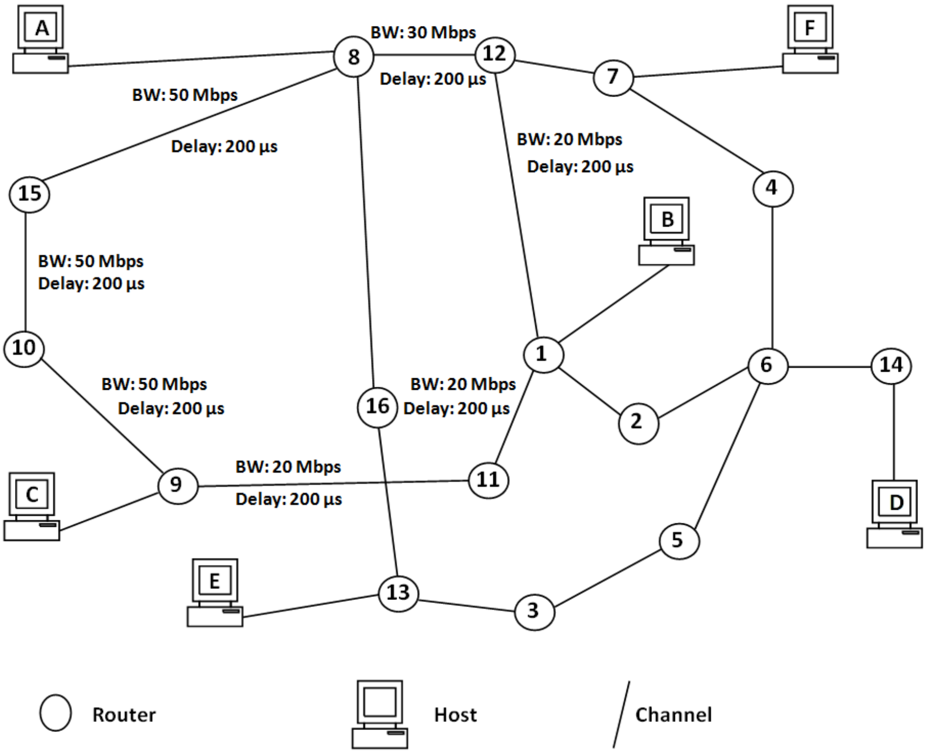


Fig. 7-6 EIGRP network of 16 routers and 6 hosts without controller

From the received *EIGRP updating* messages, router 1 calculates the *metric* to destination Host C which is directly connected to router 9 through several paths. Router 1 applies equation (2) to perform its calculations. As delay represented in equation (2) in tens of microseconds and bandwidth represented in kilobits, the link delay should be divided by 10 and bandwidth should be divided by 1000.

1. Through routers 11 and 9.

   Total delay = (Delay of link 1-11) + (Delay of link 11-9) = 200 + 200 = 400.

   Path bandwidth = Minimum (Bandwidth of link 1-11, Bandwidth of link 11-9)

   Bandwidth of link 1-11 = Bandwidth of link 11-9 = 20 Mbps.

   Minimum bandwidth over the path is 20 Mbps.

   Scaled bandwidth = Path bandwidth ÷ 1000 = 20 000 000 ÷ 1000 = 20 000

   Scaled delay = Total delay ÷ 10 = 400 ÷ 10 = 40

Router 1 applies equation (2), *metric* = (10 000 000/20 000 + 40) * 256 = 138240.

2. Through routers 12, 8, 15, 10 and 9. Minimum bandwidth over the path is 20 Mbps and total delay = 200 + 200 + 200 + 200 + 200 = 1000

   Scaled bandwidth = 20 000 000 ÷ 1000 = 20 000

   Scaled delay = 1000 ÷ 10 = 100

   Router 1 applies equation (2), *metric* = (10 000 000/20 000 + 100) * 256 = 153600

3. Through routers 2, 6, 4, 7, 12 …… and 9.

   Router 1 applies equation (2), *metric* = very high value. (Ignored)

Router 1 adopts the path 1 - 11 - 9 as a *successor* path because it has the least *metric* value. In the same way, router 12 calculates *metric* to destination Host C through several paths.

1. Through routers 8, 15, 10 and 9. Minimum bandwidth over the path is 30 Mbps and total delay = 200 + 200 + 200 + 200 = 800.

   Scaled bandwidth = 30 000 000 ÷ 1000 = 30 000

   Scaled delay = 800 ÷ 10 = 80

   Router 12 applies equation (2), *metric* = (10 000 000/30 000 + 80) * 256 = 105813

2. Through routers 1, 11 and 9. Minimum bandwidth over the path is 20 Mbps and total delay = 200 + 200 + 200 = 600.

   Scaled bandwidth = 20 000 000 ÷ 1000 = 20 000

   Scaled delay = 600 ÷ 10 = 60

   Router 12 applies equation (2), *metric* = (10 000 000/20 000 + 60) * 256 = 143360

3. Through routers 7, 4, 6, 5…… and 9.

   Router 1 applies equation (2), *metric* = very high value. (Ignored)

Router 12 adopts the path 12 - 8 - 15 - 10 - 9 as a *successor* path because it has the least *metric* value. The constructed routing table inside router 1 related to destination host C is shown in Table VII-I.

TABLE VII-I
ROUTING TABLE OF ROUTER 1 REGARDING HOST C

Destination	Path Type	Next Router	Metric	Path of nodes
Host C	Successor	11	138240	1 – 11 – 9
Host C	Feasible successor 1	12	153600	1 – 12 – 8 – 15 - 10 – 9

The routing table inside router 12 related to destination host C is shown in Table VII-II.

TABLE VII-II
ROUTING TABLE OF ROUTER 12 REGARDING HOST C

Destination	Path Type	Next Router	Metric	Path of nodes
Host C	Successor	8	105813	12 – 8 – 15 – 10 – 9

In addition to the *successor* path, router 1 has one *feasible successor* path to host C across router 12 while router 12 does not have any *feasible successor* path to host C.

All routers use the same method to calculate the *metric* values to all destinations. The routing table is represented in a C++ map container inside the router. It is invoked to forward all messages as well as the *Data* packets.

7.9.3 Extra Features in Design

Some extra features are added to our design to compensate the difference from the real design of EIGRP:

- The routing table is built gradually through exchanging *EIGRP updating* messages among the routers as there is no *topology table* used.
- The designed *EIGRP updating* messages contain minimum bandwidth, total delay and path of routers to every destination they advertise.

7.9.4 The Queue of the Router

The queue is connected to the output gates of the router; the capacity of the queue is designed to have the same value for all the routers in the network. To make the connection between the routers and the controller faster and more reliable, scheduling mechanism of the queue is adjusted so that the sending priority of *Flow status updating* messages and *Control* messages is higher than the sending priority of *Data* packets.

7.9.5 The Flow Counter

If the decision of the controller for a specific router is to forward a flow over more than one path according to a specific ratio, then the router will establish a flow counter to apply this distribution ratio among the gates.

7.10 The EIGRP Controller

The controller is the device that has the responsibility to monitor and manage the traffic across the network and to deal with any congestion problem. It can be directly connected only to some routers. For the routers that are not directly connected to the controller, all the exchanged messages use the EIGRP routing table to find their way from routers to the controller and vice versa. The controller is identified with an address. Each router contacts the controller through the path of less *metric* value.

7.10.1 The Controller Behavior

The controller receives all the information related to the network, such as topology, channels' bandwidth and EIGRP routing tables of all routers through the *Information* messages sent by the routers. According to this information, the controller constructs the network topology and it knows the behaviour of the routers (the forwarding decision related to *Data* packets) when they deal with the applied flows. It can be said that the mathematical and comparison operations inside the controller represent a replica of the network and the decision is the most complicated operation of the controller.

The controller receives the *Flow status updating* messages from the head-end routers and analyses them according to the behavior of the network against the flow to see if there is any possible congestion. When the controller recognizes congestion on one link or more, it uses its Congestion Avoidance Algorithm (CAA) to solve the problem then intrudes to change the behaviour of selected routers by changing their forwarding decisions related to selected flows to prevent the congestion. These flows will be distributed or re-distributed temporarily according to the controller's decision, which is represented in the router by the received flow table. The amount of the data rate sent on each specific router's gate is an accurate value defined by the controller. The intervention of the controller in our architecture is very limited with very few *Control* messages required to be sent.

7.10.2 The Congestion Avoidance Algorithm (CAA)

When the controller intrudes to solve a congestion problem, it applies the proposed algorithms. The procedure of the Congestion Avoidance Algorithm (CAA) is shown in Fig. 7-7.

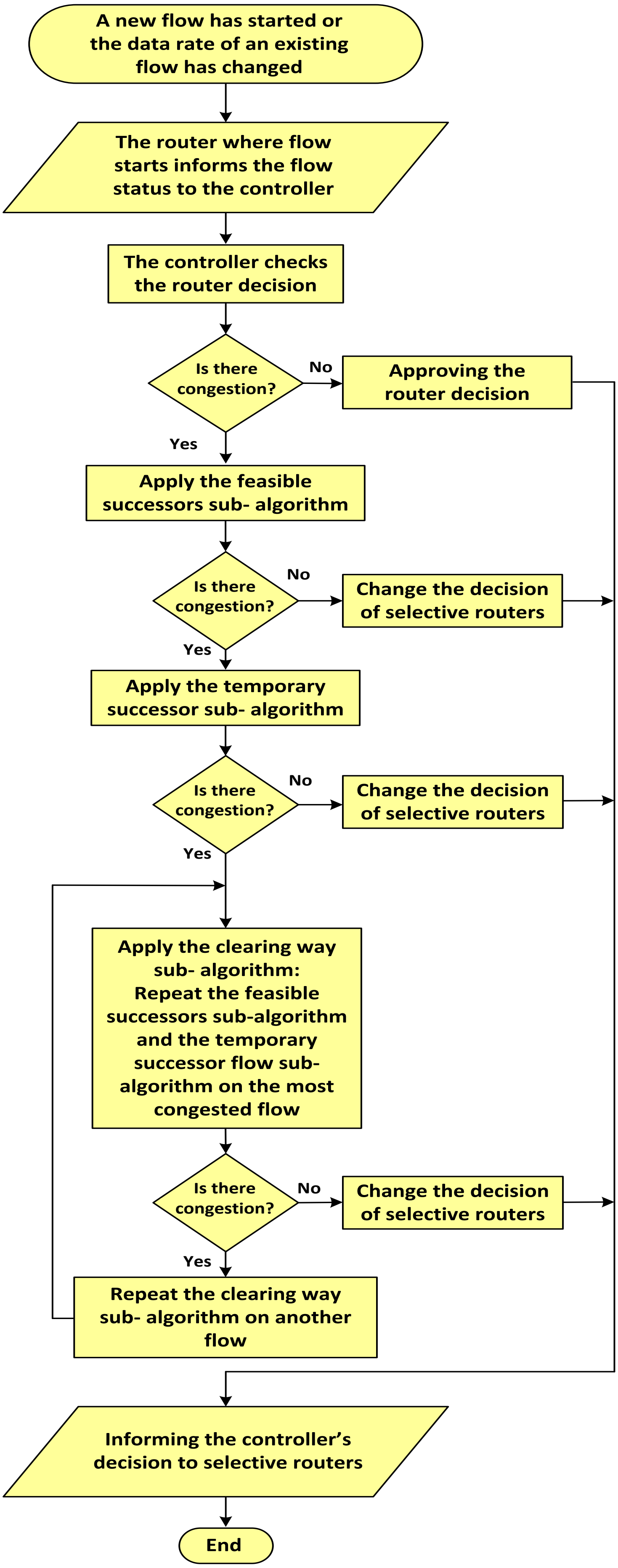

A new flow has started or the data rate of an existing flow has changed
The router where flow starts informs the flow status to the controller
The controller checks the router decision
Is there congestion?
No
Approving the router decision
Yes
Apply the feasible successors sub- algorithm
Is there congestion?
No
Change the decision of selective routers
Yes
Apply the temporary successor sub- algorithm
Is there congestion?
No
Change the decision of selective routers
Yes
Apply the clearing way sub- algorithm: Repeat the feasible successors sub-algorithm and the temporary successor flow sub- algorithm on the most congested flow
Is there congestion?
No
Change the decision of selective routers
Yes
Repeat the clearing way sub- algorithm on another flow
Informing the controller's decision to selective routers
End

Fig. 7-7 The Congestion Avoidance Algorithm (CAA) inside the controller

The Congestion Avoidance Algorithm (CAA) consists of three sub-algorithms (processes). Firstly*, the feasible successors flow distribution sub-algorithm,* then *the temporary successor sub-algorithm* and finally *the clearing way sub-algorithm.* These sub-algorithms are applied consecutively to deal with a specific flow and only on a high traffic situation, where the probability of congestion is high as well. If the current sub-algorithm solves the congestion problem, then there is no necessity to apply the next sub-algorithm. If the first two processes of the algorithm applied on a certain flow cannot solve the congestion problem, the third process will repeat the first two processes on another flow and so on. The congestion occurs when the summation of flows' rates applied on a link is bigger than the capacity of the link.

The controller applies the CAA inside the following models of the controller:

- *Finding Routes to Destination* model.
- *Flow Distribution* model.
- *Controller Processor* model.
- *Temporary Successor* model.
- *Load Balancer* model.

7.11 The Host

Each host is identified with one address and has one interface that is connected to a router. After constructing the routing tables inside routers, hosts commence exchanging *Data* packets among each other. The sending rate of *Data* packets is constant while the size (length) of the *Data* packets is changeable with time. The hosts are provided with counters to increase the sizes of the *Data* packets they send. The data rate of the flows is ascending with time until reaching specific value. Then, they begin to descend.

7.12 System Performance

For every new flow (or obvious change in the data rate of an existing flow), the following events take place:

The head-end router, the router where the flow enters the routing domain, normally deals with the flow, i.e., it forwards the flow by default according to the EIGRP routing table through the appropriate gate and, at the same time, It informs the controller by sending a *Flow updating* message that contains the flow’s information.

When the controller receives the *Flow updating* message from the router, it extracts the flow's information (source, destination and data rate) then operates an internal mathematical operation to model the network behaviour on dealing with the flow (the flow forwarding operation across the routers on the *successor* path) and notices if there is a congestion that may occur at any link inside the network due to this flow. If there is no possible congestion to occur at any link over the path where the flow is supposed to pass, the controller sends the head-end router an *Approval control* message. The head-end router and the other routers over the path continue forwarding flow packets according to their EIGRP routing table.

Example, in network 1 of Fig. 7-4, host B commences sending a low data rate flow to host C, so that the flow B-C will pass through the *successor* path across routers 1, 11 and 9. The head-end router (router 1) informs the controller about the flow status by sending *Flow status updating* messages. The controller internally tests the network behaviour against the flow; the result will be no congestion at any link if the flow is forwarded via the *successor* path. According to this result, the controller simply sends an *Approval control* message to router 1. Routers 1, 11 and 9 continue forwarding the flow B-C according to their EIGRP routing tables and over the *successor* path. The flow B-C is transmitted successfully from its sender (host B) to its final destination (host C) as shown in Fig. 7-8.

If the controller notices congestion that may occur at any link/links because of this flow, it applies its CAA to deal with the network flows. The first step of the CAA is to utilize the *feasible successor flow distribution sub-algorithm*.

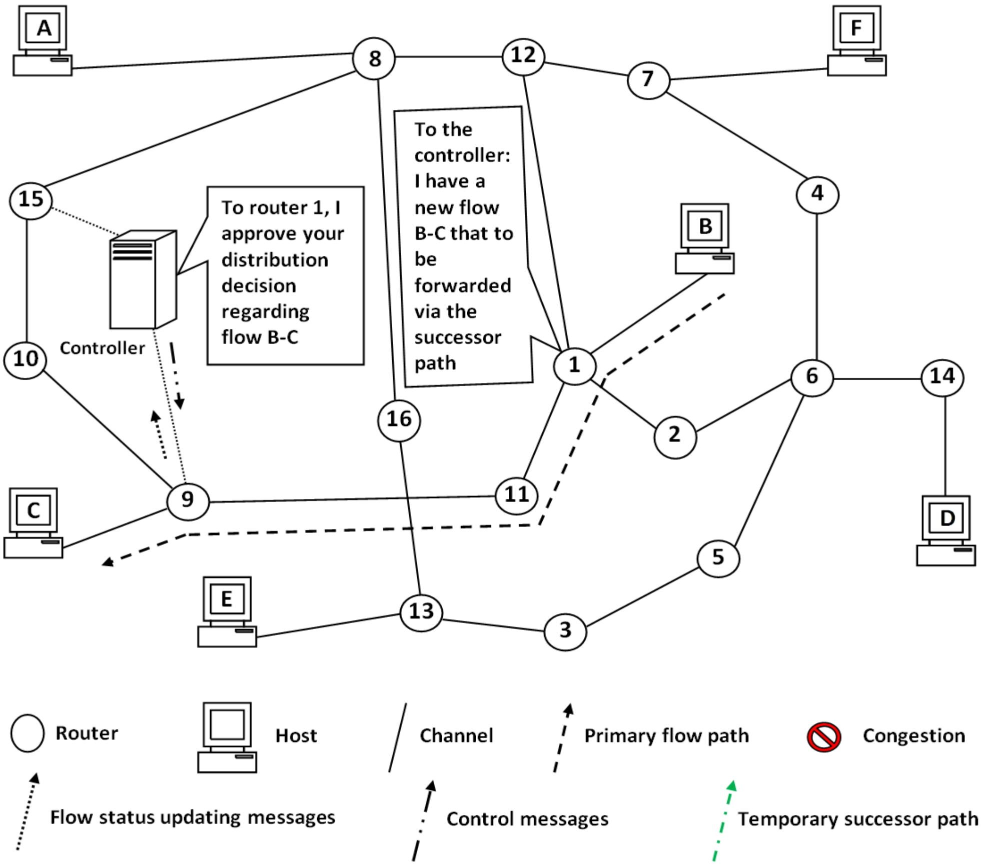

A
F
8
12
7
To the controller: I have a new flow B-C that to be forwarded via the successor path
15
To router 1, I approve your distribution decision regarding flow B-C
4
B
Controller
10
1
6
14
16
2
11
9
C
5
D
E
13
3
Router
Host
Channel
Primary flow path
Congestion
Flow status updating messages
Control messages
Temporary successor path

Fig. 7-8 Normal traffic distribution with no congestion in network 1

7.12.1 The Feasible Successors Flow Distribution Sub-Algorithm

1. Allocate the links, where the congestion occurs or is supposed to occur, mark them with a *no entry symbol* represented by 🛇 and estimate the amount of the excessive data rate that causes the congestion.
2. Test the links through all the *feasible successor* paths and record the residual bandwidth on each *feasible successor* path.
3. If the residual bandwidth of the *first feasible successor* path is greater than or equal to the extra amount of flow data rate (that causes the congestion), then forward the extra flow data rate through this path.
4. If the residual bandwidth of the *first feasible successor* path is not enough, then repeat step 3 on the s*econd feasible successor* path and so on until finding the *feasible successor* path that can handle the extra data rate.
5. If there is no *feasible successor* available, the controller will not apply this step and it will immediately move on to the next step of the CAA (the *temporary successor sub-algorithm*).

Example, suppose that the data rate of the flow B-C of Fig. 7-8 has increased significantly. Router 1 informs the controller about the current situation of flow B-C by sending *Flow status updating* message. The controller begins an internal testing operation to the flow current forwarding path, which results in a congestion at link 1-11 (similar to the network's real status) as shown in Fig. 7-9.

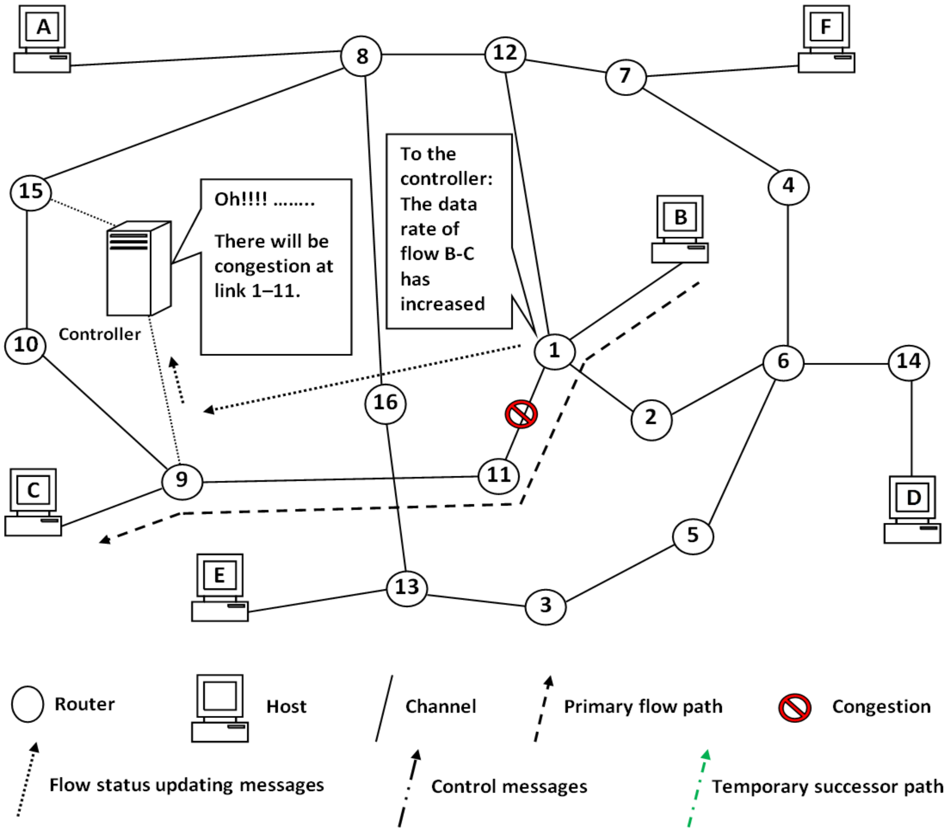


Fig. 7-9 Distinguishing of congestion that commences at link 1-11

To solve the congestion problem, the controller activates its congestion avoidance algorithm. The first step of the CAA is searching among the available *feasible successor* paths to find another path to forward the excessive flow data rate that causes the congestion. The controller decides to divert the extra flow data rate through the first *feasible successor* which passes via routers 1-12-8-15-10- 9.

The controller informs router 1 to divide flow B-C between the *successor* and the *first feasible successor* paths through its *Control* message. The flow B-C will be implemented through both the *successor* and the *first feasible successor* paths in *specific ratio* imposed by the controller so that no congestion is going to take place at any link. The flow B-C is distributed as shown in Fig. 7-10.

The load balance in our network depends on the decision of the controller, which has full instantaneous observation on the network and therefore it is probable to completely divert the flow through the *feasible successor* instead of the *successor* if there is a remote congestion on the *successor* path and the congested link is fully occupied by other flows.

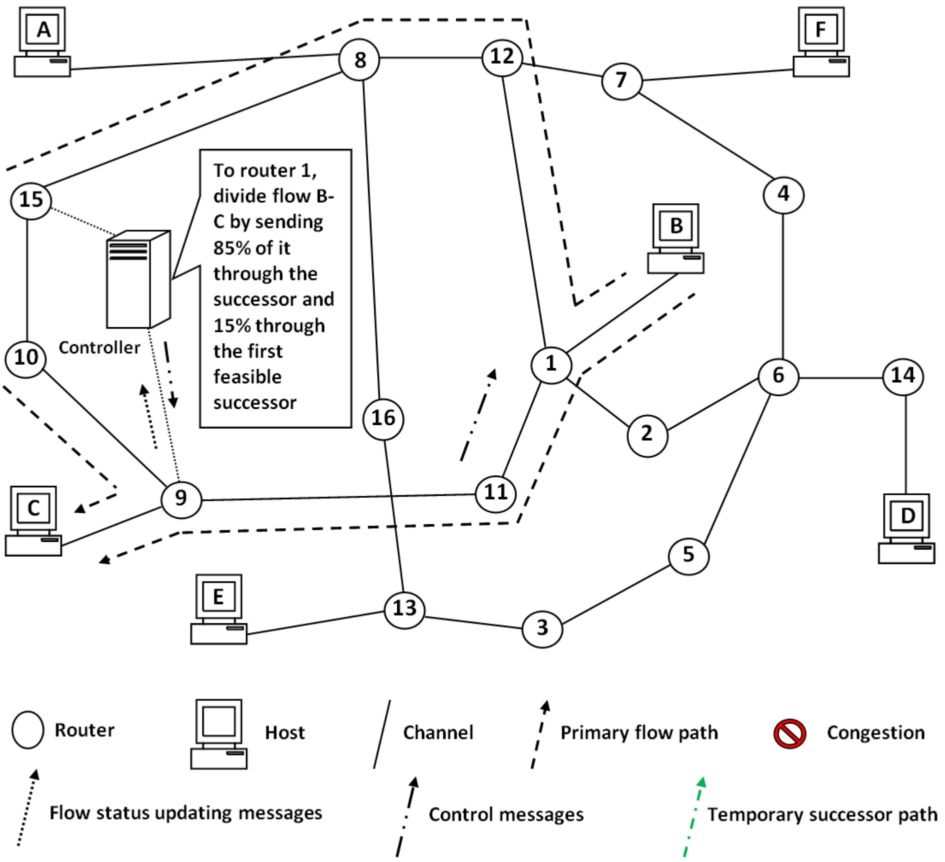

A
F
8
12
7
15
4
B
10
Controller
1
6
14
2
16
11
9
C
D
5
E
13
3
To router 1, divide flow B-C by sending 85% of it through the successor and 15% through the first feasible successor
Router
Host
Channel
Primary flow path
Congestion
Flow status updating messages
Control messages
Temporary successor path

Fig. 7-10 Applying the feasible successors flow distribution sub-algorithm in network 1 successfully

If the *feasible successors flow distribution sub-algorithm* does not solve the problem, the controller will move on to the next step of the CAA (the *temporary successor sub-algorithm*).

7.12.2 The Temporary Successor Sub-Algorithm

1. Allocate the links where the congestion is supposed to begin, mark them with a *no entry symbol* 🚫 , name them *congested links*, and then estimate the highest amount of the excessive data rate caused by the flow that causes the congestion.
2. Examine all the other links in the network. For those which have residual bandwidth of a value greater than or equal to the amount of the excessive flow data rate, name them *available links*. For the others where the spare bandwidth is smaller than the amount of the excessive flow data rate, name them *forbidden links*.
3. From the head-end router, begin a marching operation across the available links (through connections of the routers), until reaching the flow destination.
4. If the marching operation reaches a *congested link* or a *forbidden link*, then avoid that link and divert the marching to the next link of the router. If all the links connected to the router (where the marching operation has now arrived) are either

congested or *forbidden links*, then the router will be named *forbidden router*. The marching operation will start again through another path avoiding the forbidden routers until reaching the flow destination.

5. The marching operation should avoid loops.
6. The discovered path across the available links is called *temporary successor*.
7. The controller sends *Control* messages to inform each router within the temporary successor to create the temporary flow table and forward the extra flow data rate through the appropriate gate.
8. The flow may be forwarded according to both the EIGRP routing table and the flow table, or it may be forwarded according to the flow table only depending on the decision of the controller. This operation is similar to the Dijkstra algorithm and is performed inside the controller before the decisions are sent to inform to the involved routers.

Example, suppose that in the network 1 depicted in Fig. 7-10, the data rate of flow B-C has increased extremely and both the *successor* (1-11-9) and the *first feasible successor* (1-12-8-15-10-9) paths cannot handle the high data rate of flow B-C that caused congestion on links 1-11 and 12-8 as shown in Fig. 7-11.

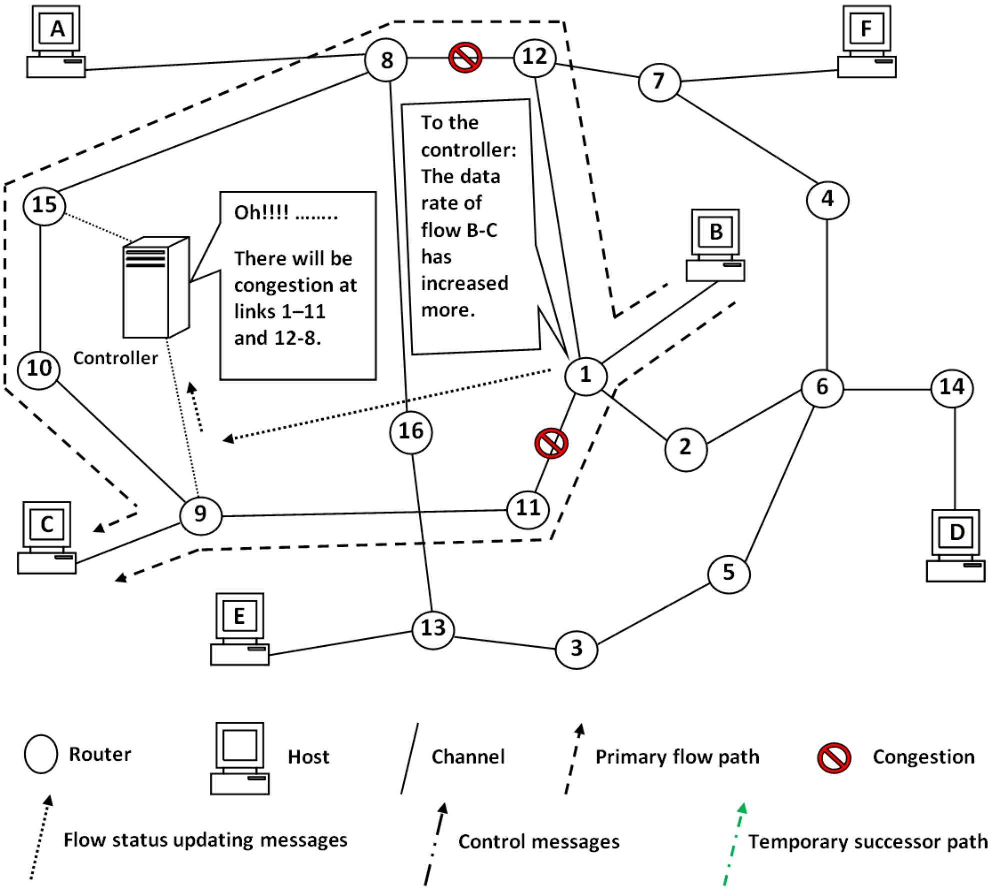
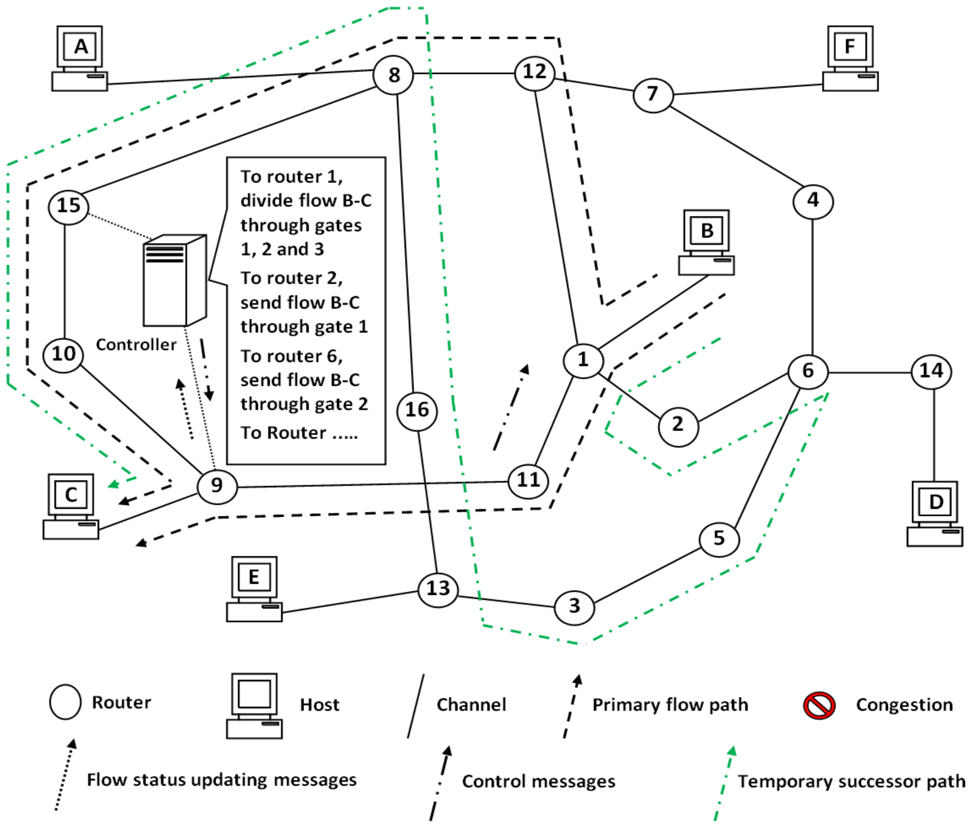

A
F
8
12
7
To the controller: The data rate of flow B-C has increased more.
4
B
15
Oh!!!! ....
There will be congestion at links 1–11 and 12-8.
Controller
10
1
6
14
16
2
C
9
11
D
5
E
13
3
Router
Host
Channel
Primary flow path
Congestion
Flow status updating messages
Control messages
Temporary successor path

Fig. 7-11 Distinguishing of congestion that commences at links 1-11 and 12-8

The controller applies the *temporary successor sub-algorithm* which finds the *temporary successor* path that passes through routers 1-2-6-5-3-13-16-8-15-10-9. The extra flow data rate will be forwarded through this path. The flow B-C will be divided among the *successor*, the *first feasible successor* and the *temporary successor* paths in a distribution ratio that does not cause congestion at any link through any path. The controller informs the involved routers to apply its decision through *Control* messages.

Unlike the *successor* and the *feasible successor* paths, the *temporary successor* path is dynamic and changeable, and depends on the instantaneous residual bandwidth of links. In router 1, the EIGRP routing table is represented by the *successor* and the *feasible successor* forwarding gates while the flow table is represented by the *temporary successor* forwarding gate as shown before in Fig. 7-1. Flow B-C is now forwarded as shown in Fig. 7-12.

It is obvious that to the head-end router (router 1), the congestion at link 1-11 represents a local congestion while the congestion at link 12-8 represents a remote congestion.



To router 1, divide flow B-C through gates 1, 2 and 3

To router 2, send flow B-C through gate 1

To router 6, send flow B-C through gate 2

To Router .....

Fig. 7-12 Applying the temporary successor sub-algorithm in network 1 successfully

If the controller fails to find an alternative path for the excessive amount of flow data rate or the *temporary successor* path is very long and not rational, then the controller starts dealing with the other flows that pass through the congested links, by applying the *clearing way sub-algorithm* described below.

7.12.3 The Clearing Way Sub-Algorithm

1. Allocate the links where the congestion is about to occur, mark them with a *no entry symbol* 🚫 , and then allocate all the flows that pass via the congested links.
2. Find a flow that is mostly repeated on those congested links.
3. Extract the amount of the flow data rate that is equal to the excessive data rate that causes the congestion.
4. Repeat using the *feasible successors flow distribution sub-algorithm* and the *temporary successor sub-algorithm* to find an alternative path to this selected flow.
5. If there is no alternative path to the flow that is mostly repeated on the congested links, then repeat the same steps on the secondly most repeated flow on the congested links and so on until terminating the congestion or reducing its effects as much as possible. Some flows will be forwarded according to the EIGRP routing tables, some other flows will be forwarded according to the flow tables and finally some others will be forwarded according to both tables.

Example, suppose that in the network 2 depicted in Fig. 7-5, host A sends a flow to host E through the *successor* EIGRP path 1-2-3-4-5, host B sends another flow to host C through the *successor* path 2-3 and host C sends a third flow to host D through the *successor* path 3-4. Both flows B-C and C-D increase their data rate until they occupy all the available bandwidth of links 2-3 and 3-4 respectively as shown in Fig. 7-13.

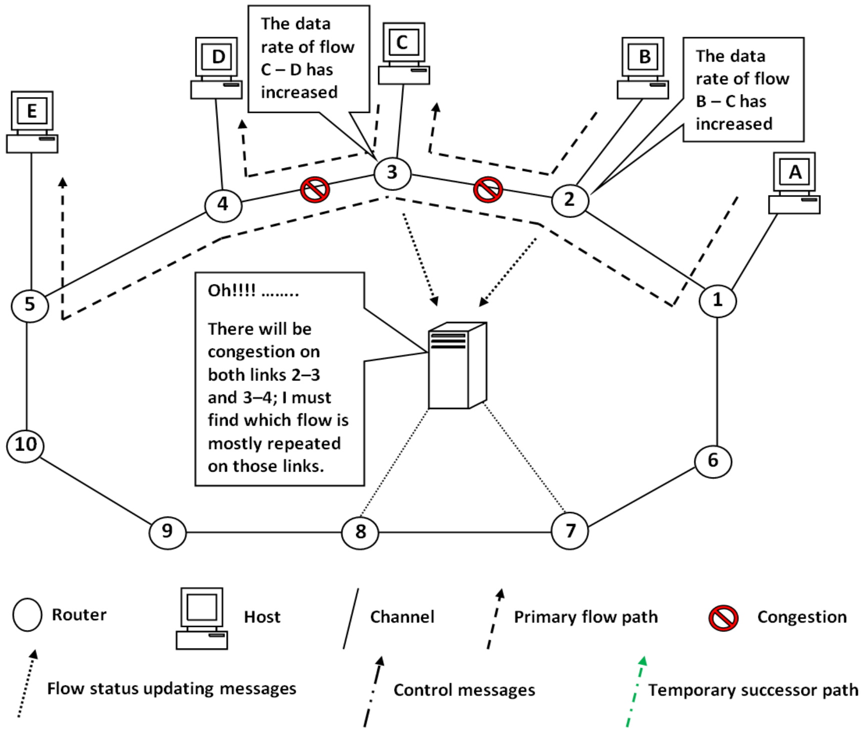

D
The data rate of flow C – D has increased
C
B
The data rate of flow B – C has increased
A
E
3
4
2
1
5
Oh!!!! ....
There will be congestion on both links 2–3 and 3–4; I must find which flow is mostly repeated on those links.
10
6
9
8
7
Router
Host
Channel
Primary flow path
Congestion
Flow status updating messages
Control messages
Temporary successor path

Fig. 7-13 Traffic distribution with congestion at two successive links in network 2

The controller tests the distribution of the flows and finds congestion at two successive links: 2-3 and 3-4. The controller identifies the flow that passing through the congested links and considers it as the most congested flow. The most congested flow now is flow A-E because it is the flow mostly repeated on the congested links.

The controller applies the feasible successors flow distribution and the *temporary successor sub-algorithms* to find an alternative path to flow A-E. The new discovered path passes through hops 1-6-7-8-9-10-5. Flow A-E is fully forwarded over the new bypass path as shown in Fig. 7-14.

The new path represents a *temporary successor* path. It can be noticed that even when a data rate increase occurred in flows B-C and C-D, the path that changed is the path of flow A-E.

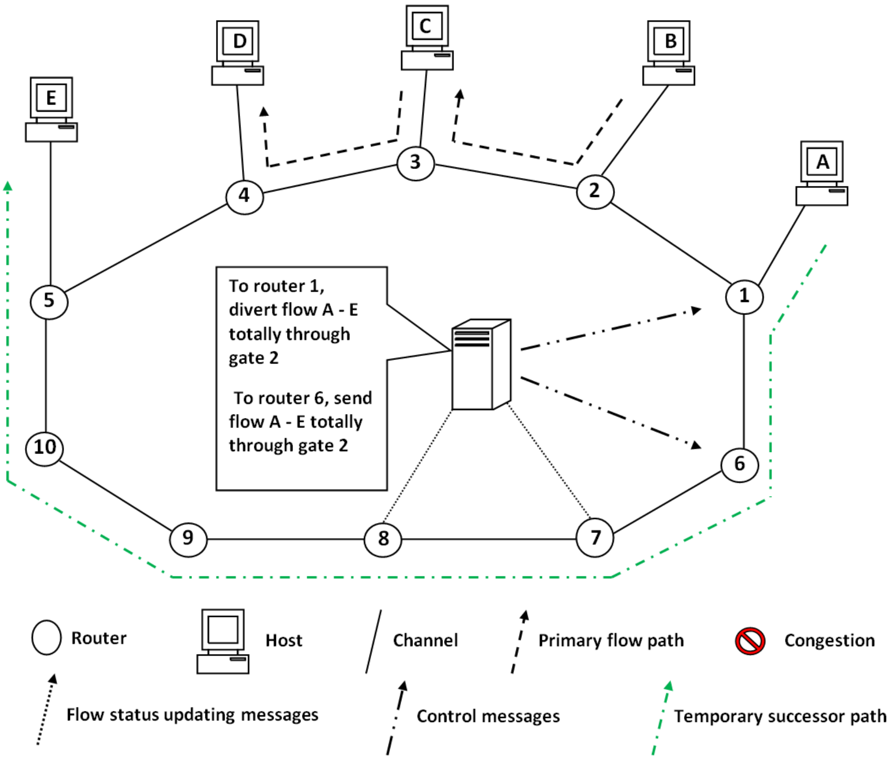

D
C
B
E
A
3
4
2
5
1
To router 1, divert flow A - E totally through gate 2
To router 6, send flow A - E totally through gate 2
10
6
9
8
7
Router
Host
Channel
Primary flow path
Congestion
Flow status updating messages
Control messages
Temporary successor path

Fig. 7-14 Applying the clearing way sub-algorithm in network 2 successfully

During the simulation, the routers send regular *Updating* messages every several minutes to the controller; those *Updating* messages inform the controller of the current status of each router, which helps to detect if there is any error regarding flows' distribution and therefore to correct the situation.

7.13 Results of Chapter 7

Both networks 1 and 2, shown respectively in Figs. 7-4 and 7-5, have been tested under two simulations. In the first simulation there is no controller (simple EIGRP), while in the second one there is a controller that achieves the mentioned steps of the CAA (enhanced EIGRP by SDN). The resulted throughput of networks 1 and 2 for both simulations (with and without controller) are shown respectively in Figs. 7-15 and 7-16.

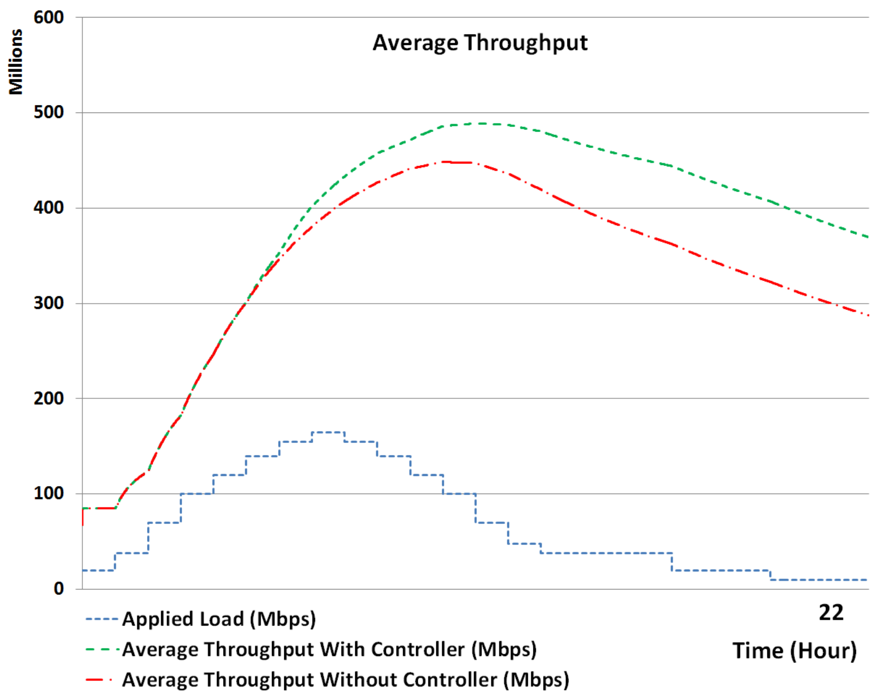


Fig. 7-15 Throughput comparison between two simulations of network 1 (with and without controller) regarding the same applied load

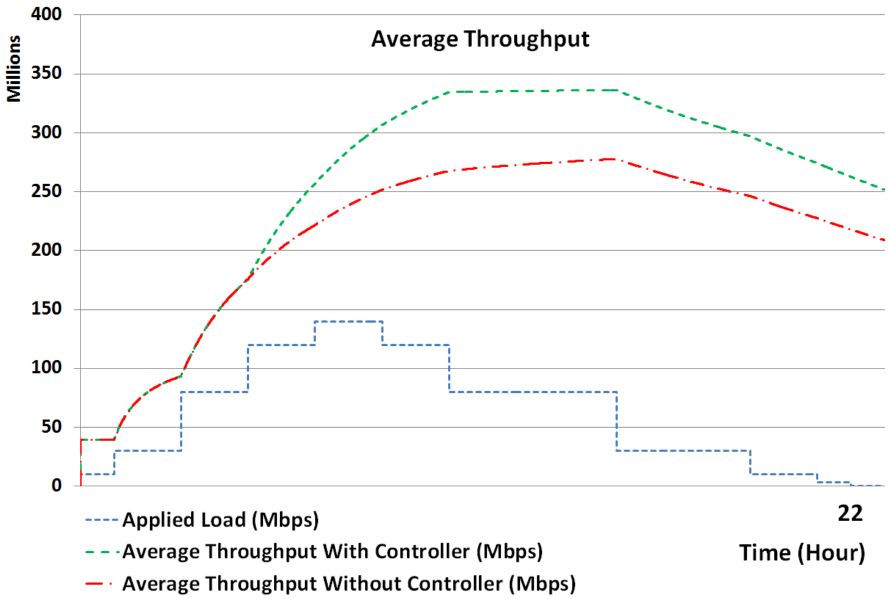


Fig. 7-16 Throughput comparison between two simulations of network 2 (with and without controller) regarding the same applied load

For Figs. 7-15 and 7-16., it seems that for low traffic, the throughput values are almost the same either by using a controller or not. By increasing the applied traffic in both simulations in both networks, the controller CAA starts working inside the controller and according to its behavior; some flows will be diverted or divided over paths to avoid congestion. Therefore, the throughput curve will obviously increase, while in the case where there is no controller, the throughput curve will slightly increase. The higher the throughput the lesser the packets drop. As the applied load increases, the controller effect becomes more obvious. After traffic returns to its normal rate, both curves start to become parallel. This means that the dealing with traffic is now similar for both simulations per network.

The number of *Control* messages issued by the controller is limited. To re-distribute a flow according to the *feasible successors flow distribution sub-algorithm*, the controller contacts the head-end router only. The controller only contacts router 1 to divide flow B-C between the *successor* and the *first feasible successor* paths as shown in Fig. 7-10.

To re-distribute a flow according to the *temporary successor sub-algorithm*, the controller contacts the routers which their EIGRP routing table differs from the decision of the controller to build a flow table inside them. Also it may contact the head-end router or any router on the EIGRP path to split the flow. In our example, the controller contacts routers 1, 2, 6, 5 and 3 to re-distribute flow B-C among the *successor*, the *feasible successor* and the *temporary successor* paths as shown in Fig. 7-12.

7.14 Summary of Chapter 7

The development of the EIGRP routing protocol algorithm by using the concepts of the SDN provides a network with better load balance, less packet drop and higher throughput. Using a controller supplies full monitoring and semi central management to the network. The performance improvement induced by the controller depends on the topology and the available resources, such as the number of routers and the links' available and residual bandwidth. Under real environment, the limited number of the *Control* messages issued by the controller saves bandwidth and gives the routers more autonomy. Under low traffic conditions, the network behaves as a traditional EIGRP network. The controller works on higher applied load to avoid congestion. Under very high traffic conditions, the applied algorithm inside the controller fails to protect all packets from being dropped. However, a network that employs the proposed controller still performs better than a network that has no controller support. The success degree of CAA or any of its sub-algorithms depends on the topology, the available resources and the currently applied load value and pattern.

Chapter 8

Conclusions and Further Work

Introduction to Chapter 8

This chapter discusses the conclusions obtained from our research, the obstacles that may prevent from applying it, the encouragements of applying it on real world and the proposed future work to develop it.

8.1 General Conclusions

Different conclusions were obtained from the different ideas implemented in this research. Our approach improves the performance of the conventional computer networks by designing them to adopt some of the SDN features. Therefore, our network model can be considered as a hybrid system. It adds the good aspects of the SDN system to the conventional networking system of technologies: OSPF/IP, OSPF/MPLS, FRR and EIGRP. This addition improves the performance of the system regarding the mentioned technologies and a within specific level. The semi-central management of the controller to the OSPF/IP routed flows in network solved the congestion problem and provided better flows' distribution. Flooding updates to the controller only to divert specific flows consumes less bandwidth than flooding updates to all the routers of the network to change the *cost* of the links, which may cause more number of flows to be diverted.

Regarding the OSPF/MPLS routed flows, the central management of the controller that is represented by involving it in allocating, establishing and restoring the primary MPLS tunnels made the establishment and the restoration easier with less consumed time. The bandwidth reserved for a tunnel is closer to the flow's current data rate passing via the tunnel. There is no need to apply the periodic MPLS TE AutoBandwidth allocator, which allows the network operator to automatically adjust bandwidth needs based on the observation of the highest average bandwidth noticed at every specific interval [9].

For the FRR mechanism, involving the controller in calculating the backup tunnels made the operation easier and allowed more bandwidth sharing among them. All the improvements led to better use of the network resources. The improvement regarding all technologies depends on several factors like:

- Network topology.
- Network size.
- Available resources (links' bandwidth)
- The amount of the applied load.
- The pattern of the applied load.
- The distance between the involved routers and the controller (the location of the controller inside the network).
- The production of the controller (which is represented by its algorithm and capabilities).

We believe that the best place to apply this approach is within an *intranet* network.

8.2 Drawbacks to Apply This Approach

Normally, every solution that deals with specific problems creates other problems, which are referred to as side effects of the treatment. The problems that are facing our approach are:

- **Size of the network:** Implementing the controller as a semi-central manager to the network involves informing the controller about all the important events of the network. In huge network, there are many routers and each represents a head-end router, which forwards many flows. This involves sending many *Flow status updating* messages to the controller.

  If the flows are changing frequently and abruptly, this involves the head-end routers sending *Flow status updating* messages to the controller whenever the flow become more or less 5 percent of its previous value (exceeds the *safety factor* value).

  Sending many *Flow status updating* messages during a small period of time may consume significant portion of the bandwidth and involve the controller with many mathematical operations.

  In a huge network, the distance between the controller and some faraway routers affects the time for the messages to be exchanged between the controller and those routers.

  In MPLS network, the controller alone is responsible of establishing the MPLS tunnels. In the huge network, it may take a longer time to establish or change the paths of many tunnels.

- **Reliability of the channel that connects the routers with the controller:** The channel between the routers and the controller is designed to be as much reliable as possible. However in the real world, an ideal channel cannot be implemented. There is a small possibility that either the *Real time updating* message or the *Control* message will be dropped. This causes small delay until the sender re-sends the dropped message.

8.3 Proposed Solutions

To solve the problem of network size, we propose dividing the network area into several areas and locating a controller at each area then connecting the controllers together. One of the controllers can be the master controller and the others are slave controllers.

8.4 Encouragements to Apply This Approach

- Improving the network performance through adding a controller and developing the routers by adding the *Operations* model, which represents some kind of process/application layer that has the ability to control the models of the other layers costs extra expenses. However, if the links (cables) of the network extend for hundreds of miles and passes through urban areas, our improvement is still cheaper than exchanging those cables with new ones of higher bandwidth.
- In spite of all the mentioned drawbacks that prevent applying this approach, the ascending technology is an encouraging factor to apply this approach.
- Regarding **section 2.21.2** of **chapter 2** that discusses the traffic management based on long term statistical records, the FDA of **chapter 4** and the CAA of **chapter 7** can deal with unpredictable traffic demand. The BPCA of **chapter 6** can calculate all possible backup paths regardless the paths of the primary tunnels and the network topology. This provides the network with dynamic features.

8.5 Further Work

We believe that our research can be further extended when dealing with implemented technologies as follows:

8.5.1 Further Work Regarding the FDA and the CAA Algorithms

FDA was designed to improve the distribution of the OSPF routed flows, while CAA was designed to improve the distribution of the EIGRP routed flows. The basic idea of designing both of the algorithms is adopting the SDN notion, which is represented by using the flow table temporarily that forward flows to avoid the congestion. If the current placements of the FDA or the CAA do not provide much better improvements to both the system performance and the QoS, the network manager can add more placements to those algorithms.

8.5.2 Further Work Regarding the OSPF/IP and MPLS Routed flows

In **chapter 4**, our network applies FDT and FDA along with a flow table to deal with congestion problem in an OSPF/IP routed flows distribution. In **chapter 5**, our network applies FDT and FDA along with *RSVP* model and messages to deal with tunnel creation time and bandwidth allocation in an OSPF/MPLS routed flows distribution. Our network has the ability to deal with the distribution of both OSPF/IP routed flows and OSPF/MPLS routed flows together at the same time. The network now deals with the distribution of *UDP Data* packets. Its capabilities can be extended to deal with services like: Voice over Internet Protocol (VOIP) and Videoconferencing (VC).

8.5.3 Further Work Regarding the Network Failure Issue

In **chapter 6** that is related to the network failure problem, our approach deals with the link failure problem only. We believe that the controller's capabilities can be extended to deal with node failure problem in both of IP and MPLS routed flows.

Another improvement can be added to the controller is by making it capable of calculating the routing table of each router in the same way the router performs this operation to itself. The routing table of each router that is calculated by the controller is identical to the routing table of the same router calculated by the router regarding the current network topology. There is no need for the *Routing table* messages as well as the testing of flows distribution after failure and re-convergence will be faster. This involves adding an *Improved OSPF* model to the controller which makes the network closer to the SDN system.

8.5.4 Further Work Regarding the EIGRP Algorithm

In **chapter 7** (EIGRP), the controller may order the splitting of specific flows to avoid congestion. If the divided flow consists of different sub-flows, the controller's capabilities can be extended to deal with those sub-flows individually. Every group of sub-flows can be forwarded via a specific path and there is no need to re-arrange the packets according to their order after reaching the destination. Thus, the controller can achieve per–packet load balancing [50] without affecting the sequence of the packets.

8.6 Adopting the SDN Notion in Other Technologies

We believe that some other networking technologies can adopt the SDN notion in order to achieve better performance. In Wireless Sensor Network (WSN), the controller can take into consideration the wireless transmission parameters such as *power* consumption.

8.7 Finally

The methodology used to develop the performance of the conventional computer networks by adopting the notion of the SDN networks succeeded in providing better results.

Thank you for reading my thesis.

References

[1] Cisco Networking Academy. (19 December 2013). *Exploring the Modern Computer Network: Types, Functions, and Hardware* [The Expanding Network]. Available: http://www.ciscopress.com/articles/article.asp?p=2158215&seqNum=7.

[2] J. Ramkumar and V. B. Kirubanand, "Notice of violation of IEEE publication principles network performance management by using stable algorithms," in *Computer Technology and Development (ICCTD), 2010 2nd International Conference On,* 2010, pp. 692-696.

[3] M. P. Clark, *Data Networks, IP and the Internet Protocols, Design and Operation.* Chichester, West Sussex, England: John Wiley & Sons Ltd, 2003.

[4] Shao Liu *et al*, "Congestion location detection: Methodology, algorithm, and performance," in *Quality of Service, 2009. IWQoS. 17th International Workshop On,* 2009, pp. 1-9.

[5] A. Al-Shabibi and B. Martin, "MultiRoute - a congestion-aware multipath routing protocol," in *High Performance Switching and Routing (HPSR), 2010 International Conference On,* 2010, pp. 88-93.

[6] Zhou Haijun, Pan Jin and Shen PuBing, "Cost adaptive OSPF," in *Computational Intelligence and Multimedia Applications, 2003. ICCIMA 2003. Proceedings. Fifth International Conference On,* 2003, pp. 55-60.

[7] N. Goldberg *et al*, "Local unicast routing control agent," in *Military Communications Conference, 2009. MILCOM 2009. IEEE,* 2009, pp. 1-7.

[8] Cisco Support Community. (22 July 2009). *How MPLS Traffic Engineering works*. Available: https://supportforums.cisco.com/document/10431/how-mpls-traffic-engineering-works.

[9] I. Cisco Systems. Advanced topics in MPLS-TE deployment: White paper. [Online]. 2009. Available: http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/multiprotocol-label-switching-traffic-engineering/whitepaper_c11-551235.pdf.

[10] K. Kar, M. Kodialam and T. V. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications," *Selected Areas in Communications, IEEE Journal On,* vol. 18, pp. 2566-2579, 2000.

[11] G. B. Figueiredo, N. L. S. da Fonseca and J. A. S. Monteiro, "A minimum interference routing algorithm," in *Communications, 2004 IEEE International Conference On,* 2004, pp. 1942-1947 Vol.4.

[12] K. M. F. Elsayed, "HCASP: A hop-constrained adaptive shortest-path algorithm for routing bandwidth-guaranteed tunnels in MPLS networks," in *Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium On,* 2004, pp. 846-851 Vol.2.

[13] S. Machajewski. (8 July 2015). *What is a Computer Network? - Types & Definition*. Available: http://study.com/academy/lesson/what-is-a-computer-network-types-definition-quiz.html.

[14] T. Lammle, *Cisco Certified Network Associate Study Guide*. Indianapolis, Indiana: Wiley Publishing, 2007.

[15] L. Peterson and B. Davie. *Computer Networks a System Approach* (5th ed.) 2010[Online]. Available: https://docs.google.com/file/d/0B21HoBq6u9TsdXdzUGdlZzJjODg/edit?pref=2&pli=1.

[16] I. Cisco Systems. (23 April 2014). *Internetworking Basics*. Available: http://www.cisco.com/cpress/cc/td/cpress/fund/ith/ith01gb.htm#xtocid166840.

[17] P. Zandbergen. (2 April 2016). *Types of Networks: LAN, WAN, WLAN, MAN, SAN, PAN, EPN & VPN*. Available: http://study.com/academy/lesson/types-of-networks-lan-wan-wlan-man-san-pan-epn-vpn.html.

[18] I. Cisco Systems. *Internetworking Technologies Handbook* (4th ed.) 2003[Online]. Available: http://docstore.mik.ua/cisco/pdf/routing/Cisco.Press.Internetworking.Technologies.Handbook.Fourth.Edition.eBook-kB.pdf.

[19] K. Summerhill. Elements for SD-WAN success. *SlidePlayer* [Online]. pp. 25. Available: http://slideplayer.com/slide/4159840/.

[20] I. Cisco Systems. (10 August 2005). *How Does OSPF Generate Default Routes?* [Document ID:13692]. Available: http://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/13692-21.html.

[21] M. Murhammer *et al*. *A Comprehensive Guide to Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management* (1st ed.) 1999[Online]. Available: http://www.redbooks.ibm.com/redbooks/pdfs/sg245234.pdf.

[22] M. Binder. (3 April 2008). *Secure Remote Data Access for Home Users*. Available: http://www.tomshardware.com/reviews/secure-remote-access,1803-2.html.

[23] D. Kreutz *et al*, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, pp. 14-76, 2015.

[24] I. aryaka. (3 September 2013). *Why SDN concepts need to extend into the WAN*. Available: http://www.aryaka.com/blog/why-sdn-concepts-need-to-extend-into-the-wan/.

[25] I. Cisco Systems. (30 January 2014). *Multiprotocol Label Switching Overview*. Available: http://www.cisco.com/c/en/us/td/docs/ios/12_2/switch/configuration/guide/fswtch_c/xcftagov.pdf.

[26] I. Cisco Systems. (02 January 2008). *Route Selection in Cisco Routers* [Document ID:8651]. Available: http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/8651-21.html.

[27] O. Bonaventure. *Computer Networking : Principles, Protocols and Practice* 2014[Online]. Available: http://cnp3book.info.ucl.ac.be/_downloads/cnp3bis.pdf.

[28] I. Cisco Systems. *IP Routing: ISIS Configuration Guide, Cisco IOS Release 12.4T* 2012[Online]. Available: http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_isis/configuration/12-4t/irs-12-4t-book.pdf.

[29] I. Cisco Systems. (02 March 2015). *OSPF: Frequently Asked Questions* [Document ID: 9237]. Available: http://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/9237-9.html#q3.

[30] K. Németh, A. Kőrösi and G. Rétvári, "Optimal OSPF traffic engineering using legacy equal cost multipath load balancing," in *IFIP Networking Conference, 2013,* 2013, pp. 1-9.

[31] D. Katz, K. Kompella and D. Yeung. "Traffic engineering (TE) extensions to OSPF version 2". *RFC 3060,* September 2003.

[32] I. Cisco Systems. (10 August 2005). *Introduction to EIGRP* [Document ID:13669]. Available: http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13669-1.html.

[33] I. Cisco Systems. (03 June 2009). *How Does Unequal Cost Path Load Balancing (Variance) Work in IGRP and EIGRP?* [Document ID:13677]. Available: http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13677-19.html.

[34] J. Quittek *et al*. "Requirements for IP flow information export (IPFIX)". *RFC 3917,* 2004.

[35] W. Stallings. Software-defined networks and OpenFlow. *The Internet Protocol Journal. Cisco Systems.* [Online]. 2013. Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_16-1/161_sdn.html.

[36] E. Rodriguez. (29 May 2014). *TCP vs. UDP*. Available: http://www.skullbox.net/tcpudp.php.

[37] L. D. Ghein. *MPLS Fundamentals* (1st ed.) 2006[Online]. Available: http://www.gregenterprises.com/mpls-fundamentals.9781587051975.27251.pdf.

[38] A. Gupta and J. W. Atwood, "Feedback mechanism for MPLS path assignment using RSVP-TE," in *Telecommunications, 2006. AICT-ICIW '06. International Conference on Internet and Web Applications and Services/Advanced International Conference On,* 2006, pp. 61-61.

[39] I. Cisco Systems. (26 July 2002). *Multiprotocol Label Switching (MPLS) on Cisco Routers*. Available: http://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/fs_rtr22.html.

[40] I. Cisco Systems. Software-defined networking: Why we like it and how we are building on it. [Online]. 2013. Available:

http://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/cis13090_sdn_sled_white_paper.pdf.

[41] D. Klein and M. Jarschel. "An OpenFlow extension for the OMNeT++ INET framework". [Online]. 2013. Available: https://www3.informatik.uni-wuerzburg.de/research/ngn/ofomnet/paper-acm_with_font.pdf.

[42] C. Dixon *et al*, "Software defined networking to support the software defined environment," *IBM Journal of Research and Development*, vol. 58, pp. 3:1-3:14, 2014.

[43] G. Huston and Telstra. Measuring IP network performance. *The Internet Protocol Journal. Cisco Systems.* [Online]. 2003. Available: http://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-23/measuring-ip.html.

[44] M. Flannagan, R. Froom and K. Turek. *Cisco Catalyst QoS: Quality of Service in Campus Networks* 2003[Online]. Available: http://docstore.mik.ua/cisco/pdf/routing/Cisco.Press,.Cisco.Catalyst.QoS.Quality.of.Service.in.Campus.Networks.(2030).KB.pdf.

[45] L. B. Lim *et al*, "RED and WRED performance analysis based on superposition of N MMBP arrival proccess," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 66-73.

[46] D. COMER. *Computer Networks and Internets* (5th ed.) 2009.

[47] Y. Zhou and W. Zhuang, "Throughput Analysis of Cooperative Communication in Wireless *Ad Hoc* Networks With Frequency Reuse," *IEEE Transactions on Wireless Communications*, vol. 14, pp. 205-218, 2015.

[48] C. Demichelis and P. Chimento. IP packet delay variation metric for IP performance metrics (IPPM). *RFC 3393* 2002.

[49] I. Cisco Systems. *SNMP Configuration Guide, Cisco IOS Release 12.4T* 2011.

[50] I. Cisco Systems. (08 January 2015). *How Does Load Balancing Work?* [Document ID:5212]. Available: http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5212-46.html.

[51] I. Cisco Systems. (08 May 2013). *What Is Administrative Distance?* [Document ID:15986]. Available: http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/15986-admin-distance.html.

[52] I. Cisco Systems. (05 August 2004). *MPLS Traffic Engineering (TE)--Fast Reroute (FRR) Link and Node Protection*. Available: http://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/gslnh29.html.

[53] J. Moy. "OSPF version 2". *RFC 2328* [Online]. April 1998. Available: https://www.ietf.org/rfc/rfc2328.txt.

[54] N. McKeown *et al*. OpenFlow: Enabling innovation in campus networks. [Online]. 2008. Available: http://archive.openflow.org/documents/openflow-wp-latest.pdf.

[55] M. P. Fernandez, "Comparing OpenFlow controller paradigms scalability: Reactive and proactive," in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference On,* 2013, pp. 1009-1016.

[56] Peng Wang, Julong Lan and Shuqiao Chen, "OpenFlow based flow slice load balancing," *Communications, China,* vol. 11, pp. 72-82, 2014.

[57] N. Handigo *et al*. Plug-n-serve: Load-balancing web traffic using OpenFlow. [Online]. 2009. Available: http://conferences.sigcomm.org/sigcomm/2009/demos/sigcomm-pd-2009-final26.pdf.

[58] Hui Long *et al*, "LABERIO: Dynamic load-balanced routing in OpenFlow-enabled networks," in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference On*, 2013, pp. 290-297.

[59] M. Koerner and Odej Kao, "Multiple service load-balancing with OpenFlow," in *High Performance Switching and Routing (HPSR), 2012 IEEE 13th International Conference On*, 2012, pp. 210-214.

[60] Yang Yu *et al*, "A framework of using OpenFlow to handle transient link failure," in *Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference On*, 2011, pp. 2050-2053.

[61] Hailong Zhang *et al*, "SDN-based ECMP algorithm for data center networks," in *Computing, Communications and IT Applications Conference (ComComAp), 2014 IEEE,* 2014, pp. 13-18.

[62] K. Phemius and M. Bouet, "OpenFlow: Why latency does matter," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium On*, 2013, pp. 680-683.

[63] Omnet Team. OMNeT++ user manual version 4.3. Omnet Inc. Staunton, Virginia , USA. 2011.

[64] Omnet Team, "INET framework for OMNeT++ Manual," Omnet Inc., Staunton, Virginia , USA, 2012.

[65] A. Tiwari and A. Sahoo, "A local coefficient based load sensitive routing protocol for providing QoS," in *12th International Conference on Parallel and Distributed Systems - (ICPADS'06),* 2006, pp. 8 pp.

[66] E. Oki and A. Iwaki, "Load-Balanced IP Routing Scheme Based on Shortest Paths in Hose Model," *Communications, IEEE Transactions On,* vol. 58, pp. 2088-2096, 2010.

[67] M. Antic *et al*, "Two phase load balanced routing using OSPF," *Selected Areas in Communications, IEEE Journal On,* vol. 28, pp. 51-59, 2010.

[68] A. Sahoo, "An OSPF based load sensitive QoS routing algorithm using alternate paths," in *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference On,* 2002, pp. 236-241.

[69] J. M. Abdul-Jabbar, O. A. Hazim and Z. N. Abdulkader, "Wavenet-based dual-path congestion control routing mechanism," in *Electrical, Communication, Computer, Power, and Control Engineering (ICECCPCE), 2013 International Conference On,* 2013, pp. 58-63.

[70] I. M. Kamrul and E. Oki, "PSO: preventive start-time optimization of OSPF link weights to counter network failure," *IEEE Communications Letters,* vol. 14, pp. 581-583, 2010.

[71] A. Peculea *et al*, "A novel framework for the development of congestion avoidance mechanisms," in *9th RoEduNet IEEE International Conference,* 2010, pp. 265-270.

[72] V. D. Chemalamarri, P. Nanda and K. F. Navarro, "SYMPHONY - A controller architecture for hybrid software defined networks," in *2015 Fourth European Workshop on Software Defined Networks,* 2015, pp. 55-60.

[73] M. Caria, T. Das and A. Jukan, "Divide and conquer: Partitioning OSPF networks with SDN," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM),* 2015, pp. 467-474.

[74] Y. Nakahodo, T. Naito and E. Oki, "Implementation of smart-OSPF in hybrid software-defined network," in *Network Infrastructure and Digital Content (IC-NIDC), 2014 4th IEEE International Conference On,* 2014, pp. 374-378.

[75] A. S. Alzahrani and M. E. Woodward, "Residual bandwidth as localized QoS routing metric," in *Software, Telecommunications and Computer Networks, 2008. SoftCOM 2008. 16th International Conference On,* 2008, pp. 125-129.

[76] Man-Ching Yuen, Weijia Jia and Chi-Chung Cheung, "Simple mathematical modeling of efficient path selection for QoS routing in load balancing," in *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference On,* 2004, pp. 217-220 Vol.1.

[77] I. Fiţigău and G. Toderean, "Network performance evaluation for RIP, OSPF and EIGRP routing protocols," in *Electronics, Computers and Artificial Intelligence (ECAI), 2013 International Conference On,* 2013, pp. 1-4.

[78] I. Cisco Systems. (2 May 2016). *MPLS FAQ For Beginners* [Document ID:4649]. Available: http://www.cisco.com/c/en/us/support/docs/multiprotocol-label-switching-mpls/mpls/4649-mpls-faq-4649.html.

[79] J. P. Ashwini, M. Sushma and H. A. Sanjay, "Queuing delay aware path selection algorithm as extension to OSPF," in *Electronics Computer Technology (ICECT), 2011 3rd International Conference On,* 2011, pp. 99-103.

[80] R. Braden *et al*. Resource ReSerVation protocol (RSVP) -- version 1 functional specification. [Online]. 1997. Available: https://tools.ietf.org/html/rfc2205.

[81] D. Awduche *et al*. RSVP-TE: Extensions to RSVP for LSP tunnels. [Online]. 2011. Available: https://tools.ietf.org/html/rfc3209.

[82] R. Braden and L. Zhang. Resource ReSerVation protocol (RSVP) - version 1 message processing rules. [Online]. 1997. Available: https://tools.ietf.org/html/rfc2209.

[83] J. Kempf *et al*, "OpenFlow MPLS and the open source label switched router," in *Teletraffic Congress (ITC), 2011 23rd International*, 2011, pp. 8-14.

[84] U. Lakshman and L. Lobo. "MPLS traffic engineering," in *MPLS Configuration on Cisco IOS Software*Anonymous 2006, [Online]. Available: http://www.ciscopress.com/articles/article.asp?p=426640&seqNum=3.

[85] I. Cisco Systems. (5 August 2004). *MPLS Traffic Engineering: Inter-AS TE*. Available: http://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/gsintast.html.

[86] M. Zhu, W. Ye and S. Feng, "A new dynamic routing algorithm based on minimum interference in MPLS networks," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008, pp. 1-4.

[87] P. Ashwood *et al*. Improving topology data base accuracy with label switched Path Feedback in constraint based label distribution protocol. [Online]. 2003. Available: https://tools.ietf.org/html/draft-ietf-mpls-te-feed-06.

[88] Xiaogang Tu *et al*, "Splicing MPLS and OpenFlow tunnels based on SDN paradigm," in *Cloud Engineering (IC2E), 2014 IEEE International Conference On,* 2014, pp. 489-493.

[89] Kun Hao and Zhigang Jin, "An on-line routing algorithm based on the off-line optimal computing in MPLS," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference On*, 2009, pp. 1-5.

[90] Chongwen Wang and Jing Hu, "Improvement of running tunnel based on OSPF TE," in *Web Information Systems and Mining (WISM), 2010 International Conference On*, 2010, pp. 3-7.

[91] I. Cisco Systems. (29 June 2007). *MPLS Traffic Engineering: RSVP Hello State Timer*. Available: http://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/gsrsvpht.html.

[92] S. Fahmy and R. Jain. Resource ReSerVation protocol (RSVP). CRC Press LLC. 2000[Online]. Available: http://www.cse.wustl.edu/~jain/books/ftp/rsvp.pdf.

[93] D. Tipper *et al*, "An analysis of the congestion effects of link failures in wide area networks," *IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 179-192, 1994.

[94] M. Shand and S. Bryant. IP fast reroute framework. [Online]. 2010. Available: https://tools.ietf.org/html/rfc5714.

[95] I. Cisco Systems. (10 August 2005). *OSPF Design Guide* [Document ID:7039]. Available: http://www.cisco.com/c/en/us/support/docs/ip/open-shortest-path-first-ospf/7039-1.html.

[96] A. Mereu *et al*, "Primary and backup paths optimal design for traffic engineering in hybrid IGP/MPLS networks," in *Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop On*, 2009, pp. 273-280.

[97] P. Pan, G. Swallow and A. Atlas. Fast reroute extensions to RSVP-TE for LSP tunnels. [Online]. 2005. Available: https://tools.ietf.org/html/rfc4090.

[98] A. Hassan *et al*, "Investigation of fast reroute mechanisms in an optical testbed environment," in *High-Capacity Optical Networks and Enabling Technologies (HONET), 2010,* 2010, pp. 247-251.

[99] M. Yuksel, K. K. Ramakrishnan and R. D. Doverspike, "Cross-layer failure restoration techniques for a robust IPTV service," in *Local and Metropolitan Area Networks, 2008. LANMAN 2008. 16th IEEE Workshop On*, 2008, pp. 49-54.

[100] I. Cisco Systems. MPLS traffic engineering (TE) - autotunnel primary and backup. [Online]. 2004. Available: http://www.cisco.com/c/en/us/td/docs/ios/mpls/configuration/guide/12_2sy/mp_12_2sy_book/mp_te_autotunnel.html.

[101] D. Wang and G. Li, "Efficient Distributed Bandwidth Management for MPLS Fast Reroute," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 486-495, 2008.

[102] M. Kodialam and T. V. Lakshman, "Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE,* 2001, pp. 376-385 vol.1.

[103] S. Raza, F. Aslam and Z. A. Uzmi, "Online routing of bandwidth guaranteed paths with local restoration using optimized aggregate usage information," in *IEEE International Conference on Communications, 2005. ICC 2005. 2005,* 2005, pp. 201-207 Vol. 1.

[104] S. Kaptchouang, I. A. Ouédraogo and E. Oki, "Preventive Start-Time Optimization of Link Weights With Link Reinforcement," *IEEE Communications Letters,* vol. 18, pp. 1179-1182, 2014.

[105] Di Yuan, "A bicriteria optimization approach for robust OSPF routing," in *IP Operations & Management, 2003. (IPOM 2003). 3rd IEEE Workshop On*, 2003, pp. 91-98.

[106] V. Janani and R. Chandrasekar, "Enhanced fast emergency path schema (EFEP-S) to redcue packet loss during multiple independent link failure in OSPF routing," in *Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference On,* 2013, pp. 1-6.

[107] R. Bartos and M. Raman, "A heuristic approach to service restoration in MPLS networks," in *Communications, 2001. ICC 2001. IEEE International Conference On,* 2001, pp. 117-121 vol.1.

[108] J. Kang and M. J. Reed, "Bandwidth protection in MPLS networks using p-cycle structure," in *Design of Reliable Communication Networks, 2003. (DRCN 2003). Proceedings. Fourth International Workshop On*, 2003, pp. 356-362.

[109] M. S. Kiaei, C. Assi and B. Jaumard, "A Survey on the p-Cycle Protection Method," *IEEE Communications Surveys & Tutorials*, vol. 11, pp. 53-70, 2009.

[110] O. Klopfenstein, "Robust pre-provisioning of local protection resources in MPLS networks," in *Design and Reliable Communication Networks, 2007. DRCN 2007. 6th International Workshop On*, 2007, pp. 1-7.

[111] R. K. Sinha *et al*, "Network design for tolerating multiple link failures using fast re-route (FRR)," in *Design of Reliable Communication Networks (DRCN), 2014 10th International Conference on The*, 2014, pp. 1-8.

[112] A. Jarry, "On the complexity of computing shortest fast reroute paths," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress On*, 2010, pp. 595-600.

[113] Y. Tsegaye and T. Geberehana. "OSPF Convergence Times," , 2012.

[114] J. Expósito, V. Trujillo and E. Gamess, "Easy-EIGRP: A didactic application for teaching and learning of the enhanced interior gateway routing protocol," in *Networking and Services (ICNS), 2010 Sixth International Conference On*, 2010, pp. 340-345.

[115] I. Cisco Systems. (05 January 2015). *Enhanced Interior Gateway Routing Protocol* [Document ID:16406]. Available: http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/16406-eigrp-toc.html.

[116] Y. N. Krishnan and G. Shobha, "Performance analysis of OSPF and EIGRP routing protocols for greener internetworking," in *Green High Performance Computing (ICGHPC), 2013 IEEE International Conference On*, 2013, pp. 1-4.

[117] S. G. Thorenoor, "Dynamic routing protocol implementation decision between EIGRP, OSPF and RIP based on technical background using OPNET modeler," in *Computer and Network Technology (ICCNT), 2010 Second International Conference On*, 2010, pp. 191-195.

[118] C. Zhao, Y. Liu and K. Liu, "A more efficient diffusing update algorithm for loop-free routing," in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, 2009, pp. 1-4.

[119] A. Snigurov and V. Chakrian, "Improvement of EIGRP protocol routing algorithm based on information security metrics," in *Problems of Infocommunications Science and Technology (PIC S&T), 2015 Second International Scientific-Practical Conference*, 2015, pp. 263-265.