VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

DĚKANÁT FAKULTY ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DYNAMIC METRIC IN OSPF NETWORKS

DIZERTAČNÍ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE Ing. TOMÁŠ MÁCHA
AUTHOR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
DĚKANÁT FAKULTY ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DYNAMIC METRIC IN OSPF NETWORKS

DYNAMICKÁ METRIKA V OSPF SÍTÍCH

DIZERTAČNÍ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE
AUTHOR
Ing. TOMÁŠ MÁCHA

VEDOUCÍ PRÁCE
SUPERVISOR
doc. Ing. VÍT NOVOTNÝ, Ph.D.

BRNO 2015

ABSTRACT

The massive growth of the Internet has led to increased requirements for reliable network infrastructure. The effectiveness of network communication depends on the ability of routers to determine the best path to send and forward packets to the desired destination. Open Shortest Path First (OSPF) protocol represents one of the most widely used routing protocols and its improvement to keep pace with the rapidly changing Internet environment would be greatly appreciated. The main deficiency of this protocol, among others mentioned in the thesis, is that its metric calculation algorithm does not take current link load into consideration. This is most likely to be the bottleneck in the path what has the most negative impact on network performance. To overcome the limitations of OSPF protocol and to improve the performance of routing in OSPF networks, a novel method based on dynamic adaptation to changing network conditions and alternative metric strategy is proposed. This method solves the problem of absence of traffic awareness and inconveniently congested links that decrease the network utilization. The method is also put into practice. The performance of new method is analyzed and verified by running the tests of the proposed algorithm on real network devices.

KEYWORDS

OSPF, metric, LSA, EWMA, load balancing

ABSTRAKT

Masivní vývoj Internetu vedl ke zvýšeným požadavkům na spolehlivou síťovou infrastrukturu. Efektivita komunikace v síti závisí na schopnosti směrovačů určit nejlepší cestu pro odesílání a přeposílání paketů ke koncovému zařízení. Jelikož OSPF v současné době představuje jeden z nejpoužívanějších směrovacích protokolů, jakýkoli přínos, který by pomohl udržet krok s rychle se měnícím prostředí Internetu, je velmi vítán. Významným omezením OSPF protokolu je, mimo jiné, absence informovanosti algoritmu pro výpočet metriky o aktuálním vytížení linky. Tato vlastnost představuje tzv. slabé místo, což má negativní vliv na výkonnost sítě. Z tohoto důvodu byla navržena nová metoda založená na dynamické adaptaci měnících se síťových podmínek a alternativní strategii OSPF metrik. Navržená metoda řeší problém neinformovanosti OSPF metriky o síťovém provozu a nevhodně vytížených linek, které snižují výkonnost sítě. Práce rovněž přináší praktickou realizaci, kdy vlastnosti nové metody jsou testovány a ověřeny spuštěním testů algoritmu v reálných zařízeních.

KLÍČOVÁ SLOVA

OSPF, metrika, LSA, EWMA, load balancing

MÁCHA, T. *Dynamic Metric in OSPF Networks.* Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, 2015. 119 p. Supervised by doc. Ing. Vít Novotný, Ph.D.

DECLARATION

I declare that I have elaborated my doctoral thesis on the theme of "Dynamic Metric in OSPF Networks" independently, under the supervision of the doctoral thesis supervisor and with the use of technical literature and other sources of information which are all quoted in the thesis and detailed in the list of literature at the end of the thesis.

As the author of the doctoral thesis I furthermore declare that, concerning the creation of this doctoral thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone's personal copyright and I am fully aware of the consequences in the case of breaking Regulation § 11 and the following of the Copyright Act No 121/2000 Vol., including the possible consequences of criminal law resulted from Regulation § 152 of Criminal Act No 140/1961 Vol.

Brno ....

(Author's signature)

I would like to thank my supervisor doc. Ing. Vít Novotný, PhD. for the continuous support of my study and research.

Dedicated to my wife Lucie and daughter Anna.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AD	Administrative Distance
AS	Autonomous System
BDR	Backup Designated Router
BGP	Border Gateway Protocol
CLNP	Connection-less Network Protocol
CLNS	Connection-less Network Service
DCE	Direct Code Execution
DM-SPF	Dynamic Metric Shortest Path First
DR	Designated Router
DUAL	Diffusing Update Algorithm
EIGRP	Enhanced Interior Gateway Routing Protocol
EGP	Exterior Gateway Protocol
EMA	Exponential Moving Average
EWMA	Exponentially Weighted Moving Average
GUI	Graphical User Interface
IEFT	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Routing Protocol
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPX	Internetwork Packet eXchange
IS-IS	Intermediate System-to-Intermediate System
ISO	International Organization for Standardization
ITU	International Telecommunication Union
LCL	Lower Control Limit
LSA	Link State Advertisement
LSAck	Link State Acknowledge
LSDB	Link State Database
LSU	Link State Update
MA	Moving Average
MOSPF	Multicast Open Shortest Path First
MPLS	Multiprotocol Label Switching
NBMA	Non-Broadcast Multiple-Access
NIC	Network Interface Card
NSSA	Not-So-Stubby Area
NGN	Next-Generation Network
OID	Object Identifier
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PDU	Protocol Data Unit
QoS	Quality of Service

RFC	Request for Comments
RIP	Routing Information Protocol
SIP	Session Initiation Protocol
SMA	Simple Moving Average
SNMP	Simple Network Management Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
TLV	Type Length Value
TTL	Time To Live
UCL	Upper Control Limit
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
WAP	Wireless Access Point

LIST OF PHYSICAL CONSTANTS

α	shortest path chain to obtained as a result of Dijkstra's algorithm
β	set of subgraphs where alternative unequal cost paths could be found
γ	list of possible optimization points for DM-SPF algorithm
c_t	an actual observation in time period t
c_0	starting value (set to zero or to the mean of former observations)
E	set of edges in graph G
ε_i	list of all edges from α that have to be added to β_i for it to be a cycle
σ	standard deviation
δ	set of subgraphs where second-best alternative subpaths could be found
G	original graph used for shortest path tree calculation
λ	parameter representing weight or smoothing factor
L	width of control limits UCL and LCL
m	path metric of a second-best path for a optimization point
q	path metric over the α chain
$q(\,)$	function mapping graph edge to its cost
V	set of vertices in graph G
w_i	assigned weight for weighted moving average

INTRODUCTION

The exchange of different types of information is essential today. The networking technologies have leaked into many people's lives and many innovations have been made in networking and transmission technologies. The data networks used in our everyday lives range from local network to global internetworks. Larger networks contain multiple network devices like routers that need to forward data effectively and dynamically.

There are several techniques to interconnect networks, especially two leading technologies in networking concept – routing and switching. Routing is at the core of networking and Internet technology and cannot be completely separated from any other network processes. The choice of the best path is not about the obvious criteria, like path length or number of hops between network devices. The path chosen may be the shortest in the topology graph but the high network traffic can make the quality of service served to the users insufficient. Therefore initial research in routing with consideration of traffic optimization is needed.

Since OSPF represents one of the most widely used routing protocols, any valuable improvement to keep pace with the rapidly changing Internet environment would be greatly appreciated. The limitation of this protocol is that its link cost calculation algorithm does not take actual link load into consideration. OSPF is not traffic aware. The traffic of individual flows may change over time, possibly leading to a situation where some links carry mostly idle connections and others are congested. The OSPF metric is static what causes bottlenecks in the path.

This thesis proposes a novel approach intended for real-time, dynamic detection, measurement and analysis of changes of the network traffic in a way that relieves congested links and splits the traffic in multiple paths of the network if possible. The proposed path determination is based on the utilization of the individual links at that point in time. This method offers an approach of more efficient path usage as the bandwidth requirements grow and provides wished traffic-aware routing.

In order to ensure real-time dynamic control of processes such as occurrences of events in networks, Exponentially Weighted Moving Average technique was applied. For this thesis, detected raw data are link load observations of routers. Link load values represent data from which the moving average is computed.

Moreover, the shortcomings of default OSPF protocol are illustrated on a simple testbed. The OSPF routing system deployed for this testbed uses the shortest path routing algorithm without any improvements. Included tables and graphs confirm the bounded characteristics of OSPF, especially the absence of traffic awareness.

Several simulations in Matlab are performed to verify the behavior of method before deployment to real devices and to ease the testing and development of projected changes to the OSPF protocol. Simulations are used to explore and gain more insides into the new model.

After the predicted behavior of improved routing in OSPF networks is validated by the simulations, practical examination is deployed. When selecting a suitable router, a number of aspects is taken into account. For example the possibility to upload own firmware, number of ports, performance, price and others. Linksys WRT54GL routers meet the highest number of requests. The goal is to test some main scenarios covering all possibilities. Then compare default OSPF and proposed mechanism based on values resulting from tests performed in real network environment.

The thesis is organized as follows. The thesis contains 7 chapters. Chapter 1 presents the basic introduction to the thesis. The first section provides a brief review of the Internet Protocol Suite. The second part focuses on the Internet layer, especially Internet Protocol and related routing function that is the main object of interest. The main routing protocol representatives are described. Since the thesis is focused on routing in OSPF networks, a detailed description of OSPF protocol is provided. Other routing protocols are explained in terms of the basic functions. Chapter 2 discusses thesis objectives and several partial tasks which need to be accomplished to successfully fulfill the main goal of the thesis. Simple testbed for experimentations to illustrate the shortcomings of OSPF protocol is described in chapter 3. Mentioned routers run default OSPF routing algorithm and the results confirm basic OSPF disadvantages. The output data are summarized in graphs. Chapter 4 is devoted to the proposal of a novel method. Chapter 5 shows a technique where an interactive computing environment models the behavior of method proposed before practical implementation. After the evaluation of the method in Matlab, testing in the network under real conditions follows in chapter 6. The benefits and disadvantages of the new method are summarized in Chapter 7. This chapter also describes the comparison of default OSPF protocol and method proposed. The last chapter concludes the work by responding to the questions from the Chapter 2, discusses the results and contributions.

1 STATE OF THE ART

The critical point for modern telecommunication networks is the distribution of services over a global communication network. For the successful communication between systems and the provision of quality of service, a communication control needs to be fulfilled. Such a control is achieved by a deployment of hierarchical model of communication. The model divides the whole communication system into several layers, which are briefly described below. The thesis focuses on the efficient selecting of a path and moving data units across that path from a source to a destination providing by the Internet layer. One of the most important functions of the Internet layer is a routing.

This chapter summarizes fundamental ingredients of IP-based traffic and focuses on the properties of routing, so as to get a better understanding in different standards. The focus is on principles of internetworking and route discovery concepts. Since the thesis focuses on routing in OSPF networks, a broad range of technical details of OSPF protocol is described.

This chapter relies on credible books and Requests for Comments (RFCs) related to the particular topic.

1.1 Networking models

There are two basic types of networking models:

- International Organization for Standardization/Open Systems Connections (ISO/OSI),
- Transmission Control Protocol (TCP) and Internet Protocol (TCP/IP).

The ISO/OSI protocol stack architecture was defined by ISO [6], [7] and the protocol stack contains seven protocol layers. At each interface between two network elements, a suite of protocols for proper data exchange must be used. Layered protocols are designed so that the specific destination layer receives the same object sent by the source layer. Each layer provides specific functions and services to its upper layer. The purpose of specification of multiple layers of protocols is that network nodes communicate properly. The information is processed by the seven protocols and then transferred over physical media. At the destination, the received information is processed in reverse order and at the end data are delivered to the appropriate process. However, there is no need for communicating systems to implement all seven layers. The number of layers is influenced by the functionality.

TCP/IP model congregates standard protocols used for the Internet and other similar networks to exchange data between end components and intermediate components by the help of common rules. These rules about the format of specific data unit and processing are represented by a suite of communication protocols, especially Transmission Control Protocol (TCP) and Internet Protocol (IP). TCP/IP is sometimes called Internet Protocol Suite. The specifications of the Internet Protocol Suite must be followed to meet the requirements of the Internet system. The policies are specified in technical reports RFC

1122 [1] and RFC 1123 [2]. Since the TCP/IP model is a system of open standards, most of the official documentation is available through a series of Requests for Comment (RFCs). As in the ISO/OSI model, the TCP/IP model uses the layered protocol model to clarify the functions of networking. Upper layers are closer to users and lower layers make data ready for physical transmission over the network. According to [1], the TCP/IP Protocol Suite organizes the protocols and methods into four layers: Application Layer, Transport Layer, Internet Layer and Link Layer. This communication architecture was based on the well-known ISO/OSI model with seven layers. However, different literatures offer different approaches and additional modifications to the layered model. For example, to match ISO/OSI model, a five layer model, with Physical and Data Link layers in place of Link Layer is described. The names of the layers also vary and the Internet layer is called the Network layer.

1.1.1 Application Layer

The application layer is the top layer of the Internet Protocol Suite that provides user interfaces and support for services. The Application layer includes all processes that make use of Transport layer protocols. The most popular and widespread services include World Wide Web (WWW), electronic mail, file transfer etc. [22].

1.1.2 Transport Layer

The transport layer offers a choice between three important protocols: Transmission Control Protocol, User Datagram Protocol (UDP) and Stream Control Transmission Protocol (SCTP).

Transmission Control Protocol

Transmission Control Protocol was originally defined in RFC 793 [4] and provides connection-oriented, reliable transmission of data in an IP environment. TCP represents a dominant transport protocol used in the Internet. TCP establishes a logical end-to-end connection between two hosts. Several extensions relating to congestion control have been proposed.

User Datagram Protocol

User Datagram Protocol is a connection-less transport layer protocol defined in RFC 768 [5]. The protocol offers no reliability, flow control or error-recovery functions to IP which is used as an underlying protocol. This allows applications to exchange information with a minimum of protocol overhead. For example real-time services, such as VoIP, do not require a completely reliable transport.

Stream Control Transmission Protocol

Stream Control Transmission Protocol represents another transport layer protocol. It is a reliable connection-oriented protocol originally designed for telephony signalling over IP networks. The protocol extends TCP and offers some new features like multistreaming and multihoming which supports more than one path between hosts for flexibility.

1.1.3 Internet Layer

The Internet layer comprises network, control, mapping, group management and routing protocols. At this layer, mostly the Internet Protocol for data transfer is used.

Internet Protocol

Internet Protocol version 4 defined in RFC 791 (IPv4) [3] is designed for use in interconnected networks. The IP offers addressing and control information that enables packets to be routed. IP version 5 was an experimental Stream Transport protocol which never came into practice. IP version 6 provides different addressing system with expanded capacity [22].

1.1.4 Network Access

Link layer is the lowest layer of the hierarchy which includes certain network technology. The layer is responsible for encapsulation of IP packets into the frames transmitted by the network. It also converts the IP address into address that is appropriate for the physical network.

1.2 Routing

An important function of the Internet layer is routing. Routing is one of the main procedures of the Internet and enables establishment of robust and efficient network. Routing is a method of finding a path and forwarding packets across that path from a source to a desired destination. If possible, the data unit is sent directly to the destination, if the destination is on the same network. Otherwise, the data unit is sent to the internetwork environment and routing devices on the network must choose directions so that the data unit reaches desired destination.

Special devices, which interconnect networks and pass packets from one to the other, are called routers. Each port of a router represents different network, broadcast domain and collision domain. At each router, the packet is received, stored in a buffer, processed and examined. If the destination IP address does not belong to any of the router's directly connected networks, the router must forward this packet to another router. According to the destination address and records in the routing table, the packet is forwarded to given output interface. Destination networks are added to the routing table using either a dynamic routing protocol or by configuring static routes. Dynamic routes are learned automatically by the router, using some dynamic routing protocol while static routes are configured manually by the administrator. If the entry for the destination network is not in the routing table but there is a default route, the router will forward the packet out the interface indicated in the routing table for the default route. If the routing table does not contain any entry for the destination network and there is no default route, the router will drop the packet. The routing table needs to have the most accurate state of paths that the router can use. Out-of-date records can cause that packets may not be forwarded to the most appropriate next hop and delays or packets loss may occur. If the routing information is not setup manually, the router can learn the information dynamically from other routers in the same network. Any network associated to the individual interface is configured as a directly connected route.

The behavior of routers depends on a routing protocol. A routing protocol defines a set of rules used by a router (or any other entity that performs routing) for communication with neighboring routers. Routing protocols provide the format of sent message, the process of sharing information about the path, the process of sending error messages and initialization and termination of session and dynamic routing table management. There are two principal routing protocol groups: Interior Gateway Protocols (IGP) and Exterior Gateway Protocols (EGP).

Interior Gateway Protocol is designed to distribute routing information to the routers within a single Autonomous System (AS). Autonomous System is a set of routers and networks using a common routing policy. There are following examples of IGPs:

- Routing Information Protocol (RIP),
- Open Shortest Path First (OSPF),
- Intermediate System-to-Intermediate System (IS-IS),
- Interior Gateway Routing Protocol (IGRP),
- Enhanced Interior Gateway Routing Protocol (EIGRP).

Exterior Gateway Protocol is designed to distribute routing information between autonomous systems. There are following EGPs:

- Exterior Gateway Protocol (EGP) (not used anymore),
- Border Gateway Protocol (BGP).

There are two types of routing algorithms. The basic routing algorithms are distance vector routing and link state routing. Distance vector routing is used for example by RIP or IGRP. The need to overcome some limitations of distance vector routing protocols led to the creation of the link state routing protocols. Link state routing protocols are based on Dijkstra's algorithm and represented by OSPF and IS-IS.

1.2.1 Routing Information Protocol

Routing Information Protocol defined in RFC 1058 [23] is a distance vector routing protocol based on Bellman-Ford approach. It represents the first routing protocol used in the network based on Internet Protocol Suite. Due to deficiencies, the original RIP was evolved to RIPv2 standardized in RFC 2453 [24]. Operational procedures, timers, and stability functions of RIPv1 remain the same in RIPv2. The difference is that RIPv2 supports classless routing, authentication and multi-cast instead of broadcast. To support IPv6, the RIP next generation (RIPng) was defined in RFC 2080 [27].

RIP uses a hop count as a routing metric. The hop count is a number of routes the packet has to pass through to reach its destination with cost of 1. It is limited to networks including 15 hops. A hop count of 16 is considered an infinite distance. Each node has a distance vector with its estimated distances from itself to all other nodes. RIP stores information where to send a packet and how many hops are needed [25]. The route with the least number of hops is determined to be the best.

If the route becomes unavailable, an update is needed. The protocol sends routing-update messages at regular intervals (30 second) using a single message format. These messages are typically carried within UDP datagrams. When a router receives the update message with new or changed parameters, the metric value increases by 1 and then the new route is reflected. The router informs neighbor routers about the change independently on the scheduled updates [26]. Due to bouncing effect, the split-horizon mechanism is implemented to prevent undesirable information to be propagated.

This protocol does not solve every possible routing problem but remains popular for a small network environment. Since it is a distance-vector protocol, the algorithm itself limits the protocol to adapt to a considerable varying network situation.

Bellman-Ford algorithm

Bellman-Ford algorithm solves the problem of the single-source shortest path in weighted directed graphs.

Each router maintains a single routing table of routes from itself to the destination. Each entry in the table indicates a destination network and the number of hops to the destination. The routing tables are periodically sent to the neighboring routers. When a router receives routing tables from its neighbors, it examines the shortest paths to the destinations, compares this new information and updates it accordingly.

The algorithm is easy to implement and configure. On the other hand, sending a copy of the entire routing table in regular intervals increases network traffic. Another disadvantage is in high convergence time and hop count metric [26].

RIPv2 packet format

Figure 1.1 depicts the format of a RIPv2 packet. The packet consists of a fixed header followed by a set of route entries. RIPv2 message can contain entries for up to 25 routes.

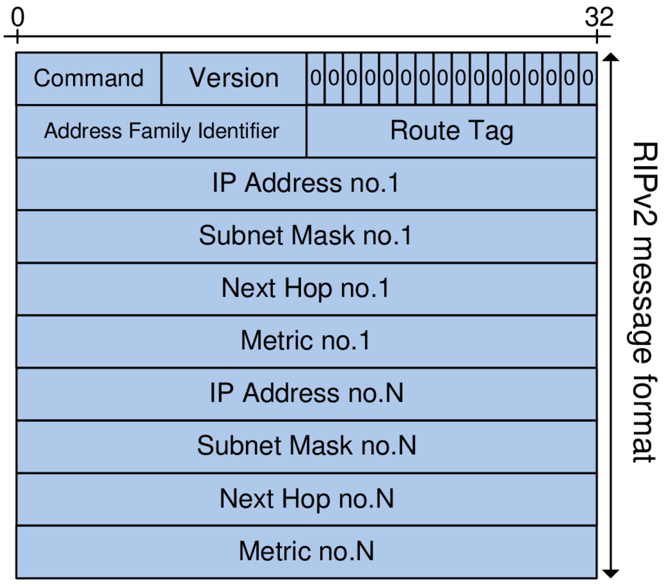


Figure 1.1 RIPv2 packet format

The fields of the RIPv2 packet are as follows:

- **Command** – This field identifies the type of RIP message (request or response).
- **Version** – Identifies the RIP version number. For RIPv2 the value is 0x02.
- **Address Family Identifier** – Identifies the type of address in the entry.
- **Route Tag** – Identifies the additional information that specifies a method for distinguishing between internal and external routes [24].

The fields of the RIPv2 entries are as follows:

- **IP Address** – Indicates the IP address of the destination of the route.
- **Subnet Mask** – This field specifies the subnet mask associated with this address.
- **Next Hop** – Identifies the IP address of the device to use as the next hop.
- **Metric** – Indicates the number of hops between 1 and 15 for a valid route and 16 for an unreachable route.

1.2.2 Open Shortest Path First

This section introduces the Open Shortest Path First protocol. The original OSPF was specified in RFC 1131 [10]. The current version of OSPF Version 2 is specified in RFC 2328 [11]. In order to support IPv6, a new version OSPVv3, specified in RFC 2740 [12], was created. OSPFv3 uses the same fundamental mechanisms as OSPFv2 but it is not backward compatible with OSPFv2.

OSPF is the most widely used link state protocol classified as an Interior Gateway Protocol. It was designed to overcome some limitations of Routing Information Protocol. The protocol uses a method based on Dijkstra's algorithm that solves the shortest path problem [11]. The algorithm dynamically determines a path of minimal total cost between the nodes. It allows routers to be selected dynamically based on the current state of the network.

The OSPF metric indicates the relative cost of the link. It is often inversely proportional to the bandwidth. A higher bandwidth results to a lower metric. The cost of the entire path is a sum of the path's particular links.

Each router periodically sends Link State Advertisement (LSA) messages with the description of connections to the router's neighbors. The LSAs are flooded through the particular area and include the calculated metric of these connections. By flooding LSAs throughout an area, all routers will build an identical Link State Databases (LSDB). From the Link State Database, the shortest path tree to other nodes is calculated according to the shortest path first algorithm. This tree characteristic allows creating a routing table which is used for the routing decisions according to the destination address. Detailed description of these fundamental procedures is written below.

OSPF area types

In OSPF ASs, all routes must keep a copy of the link state database. Larger AS brings larger databases and memory and processor demands on its routers. Therefore OSPF deploys OSPF areas. An area is a group of network segments and their attached devices. An OSPF autonomous system consists of all the OSPF areas and the routers within them. LSA flooding and the calculation of Dijkstra's algorithm is limited within an area.

- **Backbone Area** – The backbone area forms the core of an OSPF network and all other areas are connected to exchange and route information. The backbone area is also called Area 0.
- **Regular Area** – The regular areas are connected to the backbone areas. Regular areas can have several subtypes:
  - **Stub Area –** The stub area accepts the information about routes within an autonomous system but does not receive information external to the autonomous system.
  - **Totally Stubby Area** – Is a Cisco proprietary area type similar to the stub area. Totally stubby area does not receive information external to the autonomous system or routes from other areas.
  - **Not-So-Stubby Area** – Is also similar to the stub area. The not-so-stubby areas cannot accept information about routes external to the autonomous system but can import external routes from autonomous system and send them to other areas.
  - **Totally Not-So-Stubby Area** – Is a Cisco proprietary area type and an extension to not-so-stubby area that does not accept external routes or summary routes from other areas.

OSPF router types

When an OSPF autonomous system is divided into areas the routers are classified as follows:

- **Internal Router (IR)** – IR is a router with all interfaces connected to the same area.

- **Backbone Router (BR)** – BR is a router with at least one interface connected to the backbone area.
- **Area Border Router (ABR)** – ABR is a router with interfaces connected to two or more areas. ABRs perform the transmission of information from one area to another.
- **Autonomous System Boundary Router (ASBR)** – ASBR is a router with at least one interface connected to a different routing domain. The ASBR is located between OSPF autonomous system and non-OSPF network (e.g. RIP network).

OSPF packet format

OSPF routers communicate through specific OSPF packets. Each packet type and its field's definitions are described in succeeding text.

OSPF header

Each OSPF packet begins with a fixed 24-byte header. Figure 1.2 illustrates common OSPF header format. This header contains all the information an OSPF router needs to determine whether the packet should be accepted for further processing or discarded.

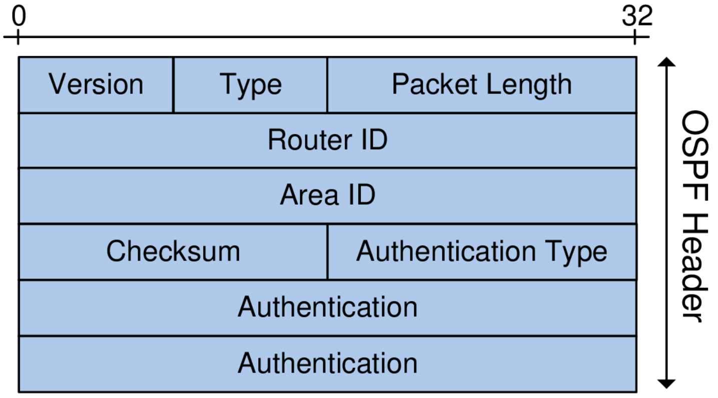


Figure 1.2 OSPF header format

The fields of the OSPF header are as follows:

- **Version** – Identifies the OSPF version number.
- **Type** – Indicates one of the OSPF types described in Table 1.1.
- **Packet length** – Indicates the total length of the packet.
- **Router ID** – Indicates an identification of a router.
- **Area ID** – Is used to designate the area number to which this packet belongs.
- **Checksum** – Is used to check the entire OSPF packet except Authentication.
- **Authentication Type** – Indicates the type of authentication used for this message. The OSPF authentication types are: no authentication, clear-text password authentication and cryptographic authentication (MD5 [38]).
- **Authentication** – 64-bit field containing authentication information.

There are five OSPF packet types listed in Table 1.1. Each type is designed to support a specific function.

Table 1.1 OSPF Packet types

Type	Packet name	Protocol Function
1	Hello	Discover/maintain neighbors
2	Database Description (DBD)	Summarize database contents
3	Link State Request (LSR)	Database download
4	Link State Update (LSU)	Database update
5	Link State Acknowledgement (LSAck)	Flooding acknowledgement

OSPF packet body

After the common header, a specific type of packet according to Table 1.1 follows. Different types of OSPF packets demand different sizes of a packet body. The size of a packet depends on a network topology, transferred entries or LSAs. LSA types are described in more detail below.

Hello packets are used to form a neighbor relationship between two routers. Figure 1.3 shows its format. Hello packets are periodically sent to multicast address 224.0.0.5 on all router's interfaces. Hello packets are sent every 10 seconds by default.

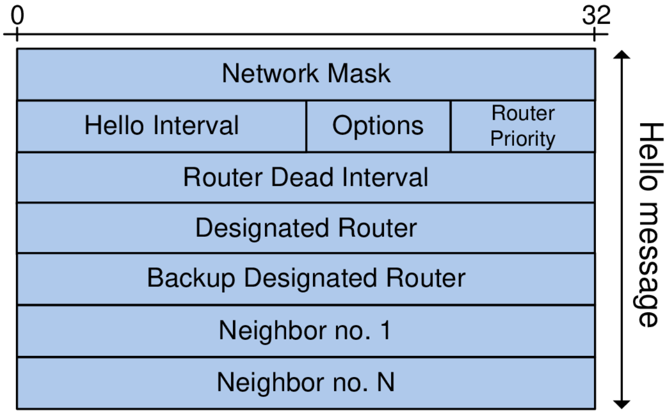


Figure 1.3 OSPF Hello packet format

The fields of the Hello packet are as follows:

- **Network Mask** – Contains a mask for the network.
- **Hello Interval** – Represents a period in seconds between Hello packets.
- **Options** – Represents the optional capabilities supported by the router. The OSPF options field is present in OSPF Hello packets, Database Description packets and all link state advertisements.
    - DN Bit – Bit is used to prevent looping in BGP/MPLS IP Virtual Private Networks as defined in RFC 4576 [15].
    - O Bit – Bit is used for receiving and forwarding Opaque LSAs.
    - DC Bit – Bit is used for demand circuit capabilities.
    - EA Bit – Bit is used for receiving and forwarding External-Attributes-LSAs.
    - N/P Bit – Bit is used for NSSA option.
    - MC Bit – Bit is used for multicast OSPF.
    - E Bit – Bit is used for AS-External-LSA option.

  - MT – Bit was originally defined as a T bit (unused ToS capability) and has been redefined as MT bit for description of router's multi-topology link exclusion capability as defined in RFC 4915 [16].
- **Router Priority** – Indicates router's priority and helps with Designated Router and Backup Designated Router electing.
- **Router Dead Interval** – Represents the number in seconds before a non-responding neighbor is considered dead.
- **Designated Router** – Identifies the Designated Router.
- **Backup Designated Router** – Identifies the Backup Designated Router.
- **Neighbor** – Contains the IP addresses of all neighbors from which this router has received Hello packets recently.

Database Description packets are used to initialize network topology database. To synchronize the databases, an asymmetric exchange is performed and master router and slave router are selected. After agreeing on the roles, the description of their databases is exchanged. Figure 1.4 illustrates common Database Description packet format. The format of the Database Description packet is very similar to both Link State Request and Link State Acknowledgment packets.

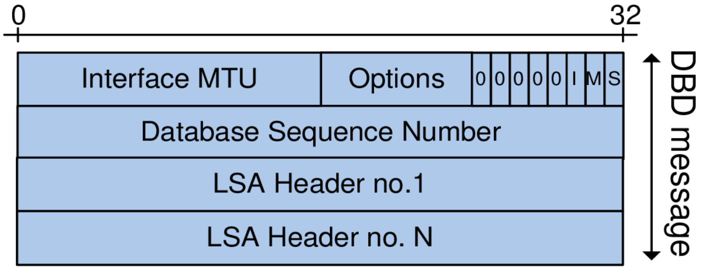


Figure 1.4 OSPF Database Description packet format

The fields of the Database Description packet are as follows:

- **Interface MTU** – This field gives the size of the largest data unit that can be sent through the associated interface.
- **Options** – Represents the optional capabilities supported by the router.
- **I Bit** – The value set to 1 indicates that this is the first packet in DBD exchange.
- **M Bit** – The value set to 1 indicates that more packets will follow.
- **S Bit** – The value set to 1 indicates that that the router is a master in the DBD exchange process. If this bit is set to 0, it means that the router is a slave.
- **Database Sequence Number** – Is used to sequence the collection of DBD Packets.
- **LSA Header** – Contains LSA Headers. LSA header is described below.

Link State Request packets are needed to request updated information of the neighbor's database. These packets contain a set of 32-bit link state record identifiers. The requested neighbor responds with the most updated information about those links. Link State Request packet format is shown in Figure 1.5.

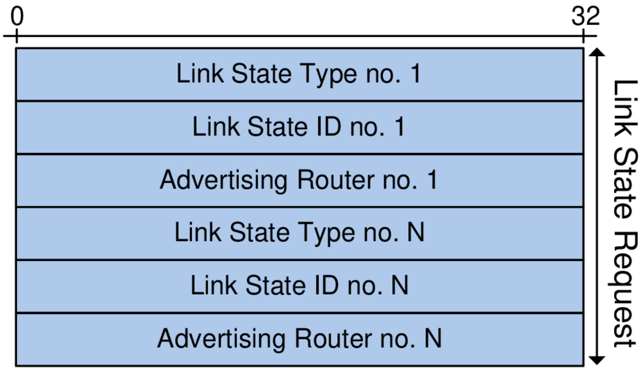


Figure 1.5 OSPF Link State Request packet format

The fields of the Link State Request packet are as follows:

- **Link State Type** – Identifies what type of LSA is being requested.
- **Link State ID** – Represents the identifier of the LSA.
- **Advertising Router** – Identifies the router that is originating this LSA.

Link State Update packets, shown in Figure 1.6, consist of a list of advertisements and implement the flooding of LSAs which can be sent in response to LSR. Link State Update packet carries one or more LSAs.

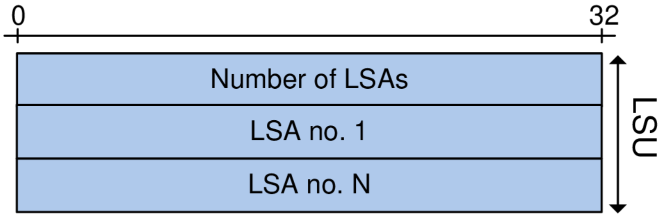


Figure 1.6 OSPF Link State Update packet format

The fields of the Link State Update packet are as follows:

- **Number of LSA** – Indicates the number of LSAs included in this packet.
- **LSA** – One or more LSAs are included.

Link State Acknowledgement is sent in response to Link State Update packets. LSAcks ensure reliable transport and information exchange. If an LSA is not acknowledged, it is retransmitted. The body of this packet is a list of LSA headers. Figure 1.7 shows the format of LSAck packet format.

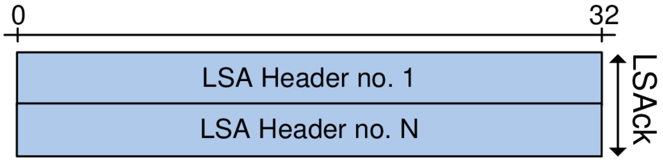


Figure 1.7 OSPF Link State Acknowledgement packet format

The fields of the Link State Acknowledgement packet are as follows:

- **LSA Header** – This field contains LSA header.

OSPF LSA header

Each OSPF LSA packet starts with a 20-bytes header. Figure 1.8 illustrates common OSPF LSA header format.

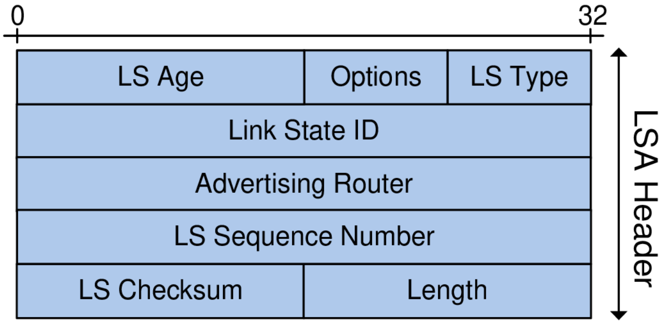


Figure 1.8 OSPF LSA header format

The LSA types defined in OSPF are shown in Table 1.2. The fields of the OSPF LSA header are as follows:

- **LS Age** – Describes time in seconds since the LSA was originated.
- **Options** – Indicates which of several optional OSPF capabilities the router supports. Options bits are exchanged between routers in DBD packets.
- **LS Type** – Indicates one of the LSA types described in Table 1.2.
- **Link State ID** – Is used to distinguish each LSA of the same LS Type.
- **Advertising Router** – Contains the value of the originating router's OSPF Router ID.
- **LS Sequence Number** – An LSA is considered to be more recent if it has higher sequence number. If the sequence numbers are equal, a higher checksum number is dominant.
- **LS Checksum** – Is used at the receiver to check the contents of the LSA except the LS Age field.
- **Length** – Defines the length of the header and the LSA contents.

Table 1.2 OSPF LSA types

Type	LSA name
1	Router-LSA
2	Network-LSA
3	Summary-LSA
4	ASBR-Summary-LSA
5	AS-External-LSA
6	Group-Membership-LSA
7	NSSA-LSA
8	External-Attribute-LSA
9	Opaque-LSA (link-local scope)
10	Opaque-LSA (area-local scope)
11	Opaque-LSA (AS scope)

Router-LSAs are generated by all routers in an area. It describes the states of the router's links to the area. Router-LSAs are flooded only within a particular area and cannot cross an Area Border Router. Figure 1.9 illustrates a Router-LSA format.

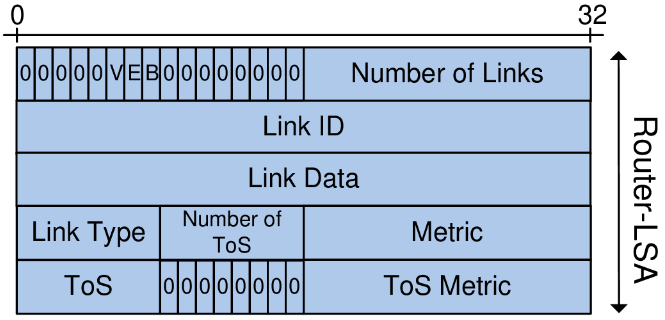


Figure 1.9 OSPF Router-LSA format

The fields of the Router-LSA packet are as follows:

- **V Bit** – Virtual link endpoint bit is set to one if the originating router is an endpoint of a virtual link.
- **E Bit** – Is set to one if the originating router is an ASBR.
- **B Bit** – Is set to one if the originating router is an ABR.
- **Number of Links** – Describes the number of router links.
- **Link ID** – Identifies the object which is connected to the link. This 32-bit field can represent neighboring router's Router ID, IP address of the DR's interface, IP network or subnet address.
- **Link Data** – This field is connected to the Link Type and provides extra information for the link.
- **Link Type** – The Link Type field describes the type of a connection the link provides. Router-LSA's Link Types and Link State IDs are described in Table 1.3.
- **Number of ToS** – Specifies the number of Types of Service metrics. Type of Service is not used anymore and set to all-zero.
- **Metric** – Defines the cost of the link to the destination.
- **ToS** – This field specifies Type of Service value (normal service, minimize monetary cost, maximize reliability, maximize throughput and minimize delay) defined in RFC 1349 [13].
- **ToS Metric** – Describes the metric associated with the specified ToS value.

Table 1.3 Link descriptions in the Router-LSA

Link Type	Description	Link State ID
1	Point-to-point connection to another router	Neighbor router ID
2	Connection to a transit network	Interface address of DR
3	Connection to a stub network	IP network
4	Virtual link	Neighbor router ID

The default OSPF metrics are summarized in Table 1.4.

Table 1.4 Default OSPF metrics

Technology	10^8/bps	Metric
Gigabit Ethernet	10^8/1 000 000 000 bps	1
Fast Ethernet	10^8/100 000 000 bps	1
Token Ring (16)	10^8/16 000 000 bps	6
Ethernet	10^8/10 000 000 bps	10
E1	10^8/2 048 000 bps	48
T1	10^8/1 544 000 bps	64
64 kbps link	10^8/64 000 bps	1562
56 kbps link	10^8/56 000 bps	1785
9.6 kbps link	10^8/9 600 bps	10 416

An example of IP packet related to OSPF Router LSA is shown in Figure 1.10. The OSPF information is encapsulated into the packet. The header contains control information for synchronization and the management of data transmission on the links. OSPF protocol runs directly on top of IP, in which the protocol field with the value of 89 specifies encapsulated OSPF protocol. OSPF packet header is included in every OSPF packet. Payload data are contained in the body of the packet.

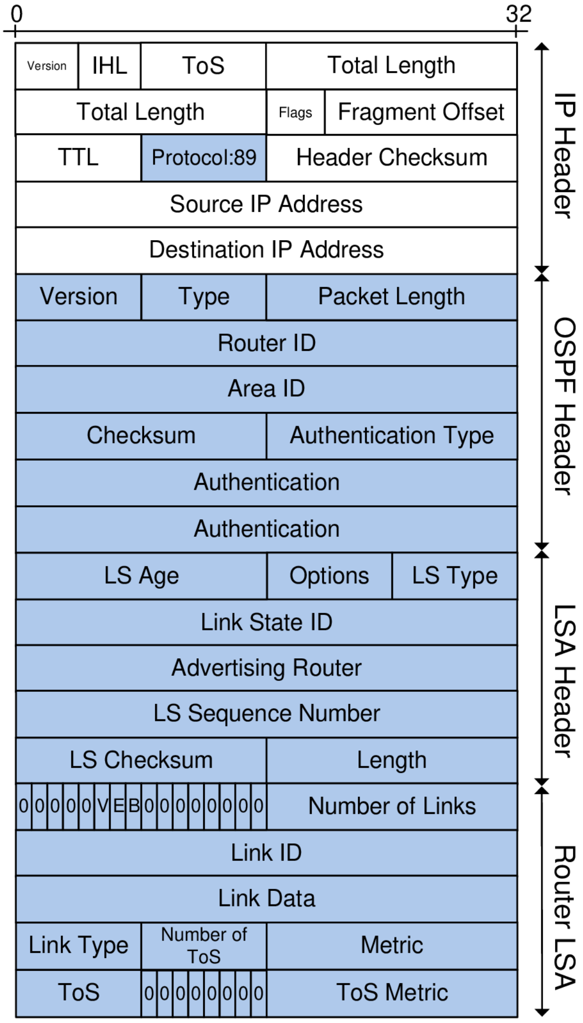


Figure 1.10 Example of IP packet related to OSPF Router LSA

Network-LSA is generated by Designated Routers for every broadcast or Non-Broadcast Multiple-Access network within an area. Network-LSA is similar to the Router-LSA. It is flooded to all routers only within an area and does not cross an ABR. The difference is that the Network-LSA is the collection of all the link state information in the network. Figure 1.11 illustrates the format of the Network-LSA.

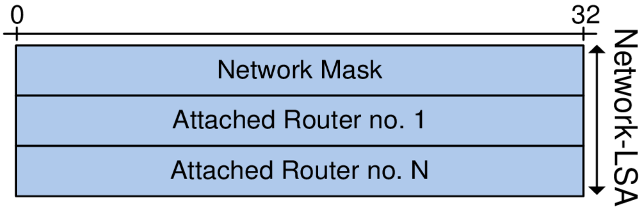


Figure 1.11 OSPF Network-LSA format

The fields of the Network-LSA packet are as follows:

- **Network Mask** – Indicates the network mask associated with the network.
- **Attached Router** – Contains a set of router IDs associated with the link.

Summary-LSAs are generated by Area Border Routers and describes route to a destination outside the area in the OSPF network. Summary-LSAs enable routers to exchange information between two or more areas. Figure 1.12 depicts the packet format of Summary-LSA.

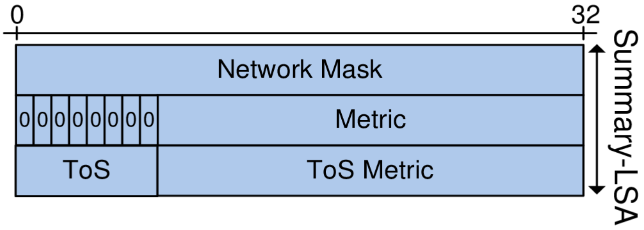


Figure 1.12 OSPF Summary-LSA format

The fields of the Summary-LSA packet are as follows:

- **Network Mask** – Indicates the network mask associated with the network.
- **Metric** – Defines the cost of the link to the destination network.
- **ToS** – This field defines the Type of Service It is not used anymore and set to all-zero.
- **ToS Metric** – Describes the metric associated with the specified ToS value. Type of Service is not used anymore and is set to all-zero.

ASBR-Summary-LSAs are similar to Summary-LSAs. They are generated by Area Border Routers and describes route to Autonomous System Border Router. In contrast with the Summary-LSA, the ASBR-Summary-LSA describes routes that are external to the OSPF network. The packet format is identical to Summary-LSA in Figure 1.12.

AS-External-LSAs are generated by Autonomous System Border Routers and describe routes to networks outside the OSPF autonomous system. AS-External-LSA packet format is shown in Figure 1.13.

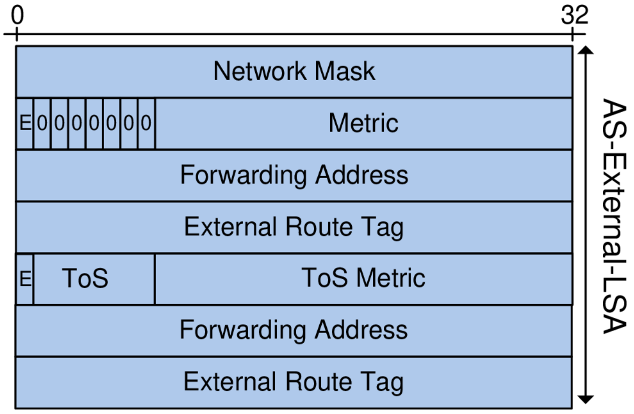


Figure 1.13 OSPF AS-External-LSA format

The fields of the AS-External-LSA packet are as follows:

- **Network Mask** – Indicates the network mask associated with the network.
- **E Bit** – External Metric bit specifies the type of external metric to be used.
- **Metric** – Defines the cost of the link to the destination.
- **Forwarding Address** – Is the address to which the data traffic for the advertised destination is forwarded.
- **External Route Tag** – This field specifies an arbitrary tag which is not used by the OSPF itself.
- **E Bit** – External Metric bit specifies the type of external metric to be used.
- **ToS** – Defines the Type of Service that the following cost is relevant to.
- **ToS Metric** – Describes the metric associated with the specified ToS value.
- **Forwarding Address** – Is the address to which the data traffic for the advertised destination is forwarded.
- **External Route Tag** – Specifies an arbitrary tag which is not used by OSPF itself.

Group-Membership-LSA is specific to a single OSPF area and was defined for Multicast extensions to OSPF MOSPF. MOSPF works by including multicast information in OSPF Link State Advertisements [8]. The Group-Membership-LSA consists of the standard 20- byte LSA header followed by a specification of transit vertex. The vertex is specified by its Vertex Type and Vertex ID (Figure 1.14). Group-Membership-LSA is not currently used.

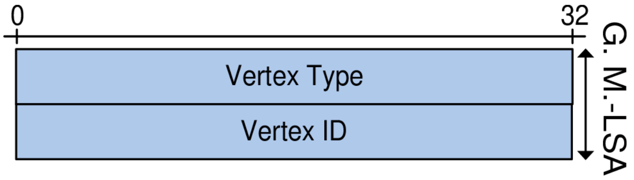


Figure 1.14 OSPF Group-Membership-LSA format

The fields of the Group-Membership-LSA packet are as follows:

- **Vertex Type** – Indicates whether the destination is a router or a transit network.
- **Vertex ID** – Specifies the originating router's router ID.

NSSA-LSA is generated by a Not-So-Stubby Area (NSSA) ASBR and allows importing of external routes into the stub area in a limited fashion. NSSA is an extension of OSPF stub area [9]. NSSA-LSA packet has a similar packet structure as AS-External-LSA as shown in Figure 1.15.

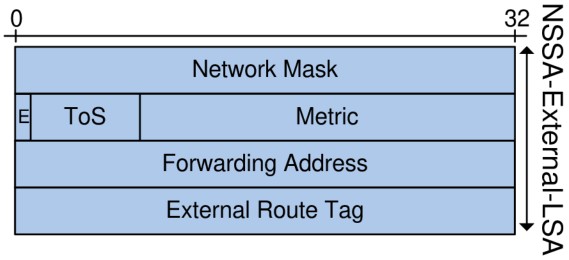


Figure 1.15 OSPF NSSA-LSA format

The fields of the NSSA-External-LSA packet are as follows:

- **Network Mask** – Indicates the network mask associated with the network.
- **E Bit** – External Metric bit specifies the type of external metric to be used.
- **ToS** – This field defines Type of Service that the following cost is relevant to.
- **Metric** – Describes the metric associated with the specified ToS value.
- **Forwarding Address** – Is the address to which the data traffic for the advertised destination is forwarded.
- **External Route Tag** – This field specifies an arbitrary tag which is not used by OSPF itself.

External-Attribute-LSA is used when BGP information is carried across OSPF autonomous system. Most of the OSPFv2 implementations have never supported this feature.

Opaque-LSAs are defined in RFC 2370 [14]. There are three types of Opaque-LSAs: Opaque-LSA link-local scope, area-local scope and AS scope. All three types consist of a standard LSA header followed by a 32-bit of application-specific data, for example an extension to be used in MPLS networks. Opaque-LSA (link-local scope) is not flooded beyond the local subnetwork. Opaque-LSA (area-local scope) is not flooded beyond their associated area. Opaque-LSA (AS scope) is flooded throughout the entire autonomous system except the stub areas.

OSPF operations

The operations of OSPF slightly vary upon the type of network in which it operates. The following discussion focuses on the overall operations belonging to all network types. OSPF operations can be divided into the three categories:

- Neighbor and adjacency initialization
- LSA flooding
- SPF tree calculation

Neighbor and adjacency initialization

OSPF implements the Hello protocol that enables routers to learn about each other. OSPF routers send Hello packets out all interfaces participating in the OSPF process. The routers confirm the neighborhood when the sending and receiving of the Hello packets is complete. After the exchanges have been completed and the parameters have been agreed, routers are considered to be merely adjacent. Figure 1.16 shows simplified information exchange procedure.

After the merely adjacency is established, the routers exchange information containing descriptions of the router's links, interfaces, router's neighbors and the state of each link to the adjacent router. The information is placed in the link state advertisement packets. Because of the various types of link state information, OSPF defines multiple LSA types (Table 1.2). The process of LSAs propagation is called LSA flooding. The LSA flooding provides fast convergence after a topology change or a periodic refresh at long-time intervals, such as 30 minutes. LSA flooding is described in the next section in detail. Sending LSAs instead of the whole databases reduces the amount of network traffic and the size of the routers' topological databases. The routers that receive the LSAs record the information into a database called Link State Database (LSDB) and forward the LSAs on to their respective neighbors. This allows all routers participating in the OSPF process to have the same view of the network, although from their own perspective. After the LSDB synchronization, the routers are considered as fully adjacent.

Each router uses the information in LSDB to calculate a shortest path tree. OSPF uses the Dijkstra's algorithm, also referred to the Shortest Path First algorithm, to determine the shortest path to all known destinations. After the calculation a routing table can be established.

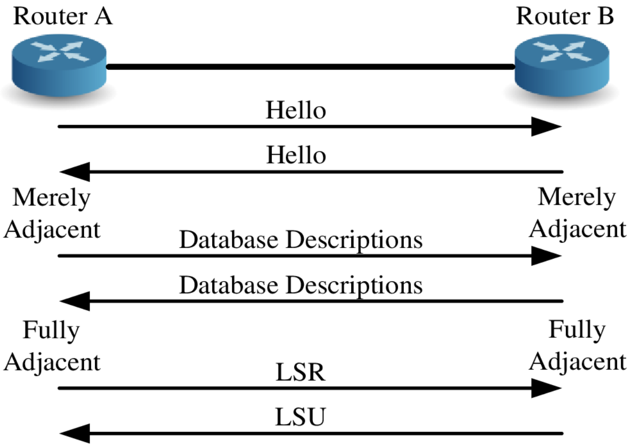


Figure 1.16 The information exchange between OSPF neighbors

LSA flooding

The flooding process initiates when a router wishes to update one or more of its self-originated LSAs. Maximum rate at which a router is able to update an LSA is every 5 seconds (in the case of failure) and in order to achieve reliability, a router must refresh LSAs every 30 minutes (in the case of stable network). LSAs are flooded within Link State Update packets out of all router's interfaces. The LSU can contain one or more LSAs. The LSU packets are received and examined by the router's neighbors. Any LSA that reaches the maximum age of 60 minutes is discarded.

According to Figure 1.17, when a router receives any LSA, it performs the following operations. If the router receives any unknown LSA entry, the entry is added to the router's Link State Database acknowledged by sending a Link State Acknowledgement message. The update is then flooded to the other routers. Dijkstra's algorithm is used to select the shortest path and finally a routing table can be calculated. Dijkstra's algorithm is described in the next section in detail.

If the entry already exists, the LSA sequence number is examined. Any LSA is considered to be more recent if it has higher sequence number. In the case of equal sequence numbers, the router ignores this LSA entry. In the case of higher sequence number, the entry is added to the database, the LSAck is sent, the update is flooded to other routers, the algorithm to find the shortest path is executed and the routing table is updated. In the case of lower sequence number, the source is notified of newer information [21], [11].

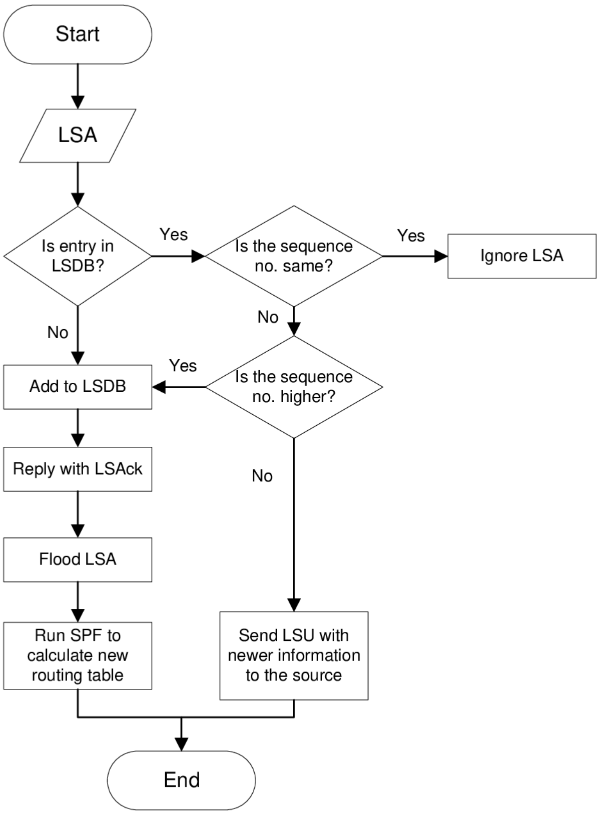

Start
LSA
Is entry in LSDB?
Yes
Is the sequence no. same?
Yes
Ignore LSA
No
No
Yes
Add to LSDB
Is the sequence no. higher?
Reply with LSAck
No
Flood LSA
Run SPF to calculate new routing table
Send LSU with newer information to the source
End

Figure 1.17 LSA operations [21]

Dijkstra’s algorithm

A special mathematical algorithm called Dijkstra’s algorithm [17] can be used to determine the shortest path tree from a given vertex in the graph $G = (V, E)$ where V is a set of vertices or nodes and E is a set of edges, to the remaining vertices in the graph. A tree is a special case of graph. The tree may be defined as a connected graph without any cycle. In the graph each edge has a weight assigned to it. The question is to find a path from one vertex to another vertex such that the sum of the weights on the path is as minimal as possible. Such a path is called the shortest path and the weights are represented by costs or metrics. All the weights in the graph should be non-negative. If the weights are negative then the current shortest path cannot be obtained. In other words, the algorithm is used to determine the metric of the shortest path from the given vertex to every vertex in V. The path is the sum of the metrics of the edges on the route.

Consider a directed weighted graph $G = \{V, E\}$ where $V = \{1, 2, \dots n\}$. The focus is to find the shortest path from the source to every vertex in V. It is useful to introduce some more notations. For a vertex $v \in V$ in a two-dimensional array of weights, let $v[i][j]$ be the weight of the edge from vertex i to vertex j. If there is no such edge, the weight is considered to be infinity. Figure 1.18 illustrates an example of a weighted directed graph containing six vertices with assigned weights and its adjacency matrix.

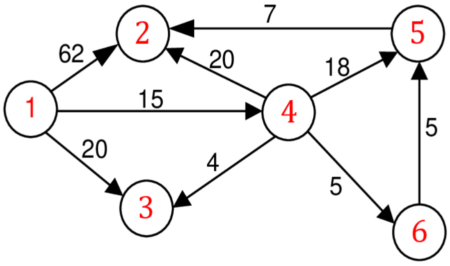
Dijkstra’s algorithm divides the set of vertices V into two lists: list S containing a set of considered vertices and a tentative list S' containing not considered vertices. As the algorithm progress, the list S expands and the list S' reduces when the vertices move to the list S. The algorithm stops when the list S' is empty [18].

For simplicity, vertex s is considered to be the source. Initially, the list S contains only the vertex s. At each step, vertex v with the shortest distance to the vertex s is added to the list S. Array d is used to record the weight of the shortest path to each vertex. It means that $d[v]$ contains the weight of the current shortest path from vertex s to vertex v.

A modification of the path from vertex s to vertex v for including vertex u improving the shortest path estimate for vertex v is called a relaxation procedure. In the case of Dijkstra’s algorithm, the relaxation is performed for an edge (u, v) in E. If there is a path from vertex s to vertex u of weight $d[u]$ and the edge $e = (u, v)$ out of vertex u, then there is a path from vertex s to vertex v of weight $d[u] + w(e)$. If this weight is smaller than the best previously known weight $d[v]$, the specific edge relaxation operation is as follows: [19], [20]

$$\text{if } d[u] + \ w(e) < d[v] \text{ then } d[v] = d[u] + w(e). \tag{1.1}$$

Dijkstra’s algorithm is applied to the directed weighted graph in Figure 1.18 to calculate the distances with respect to a vertex s (node 1). The graph consists of six nodes and assigned weights.



	1	2	3	4	5	6
1	0	62	20	15	∞	∞
2	∞	0	∞	∞	∞	∞
3	∞	∞	0	∞	∞	∞
4	∞	20	4	0	18	5
5	∞	7	∞	∞	0	∞
6	∞	∞	∞	∞	5	0

Figure 1.18 A weighted directed graph and its adjacency matrix

Initial stage

From Figure 1.19 it can be seen that node 1's neighbors are node 2, node 3 and node 4. List S considers only the node 1: $S = \{1\}$ and $S' = \{2, 3, 4, 5, 6\}$. The shortest path to these three nodes can be readily found while the weights of the rest of the nodes remain infinity:

$d[2] = 62, d[3] = 20, d[4] = 15, d[5] = \infty, d[6] = \infty$.

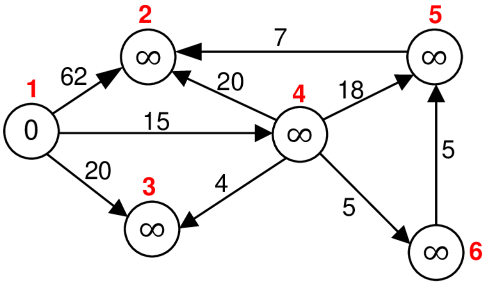


Figure 1.19 Initial stage of Dijkstra's algorithm, S = {1}

First iteration

For the minimum weight calculation, the relaxation procedure can be rewrite from (1.1) to:

$$d[v] = \min\{d[v], d[u] + w(e)\}. \tag{1.2}$$

In the first iteration, node 4 is selected and moved to the list S because $d[4]$ is the minimal weight. Thus, list S considers two nodes: $S = \{1, 4\}$ and $S' = \{2, 3, 5, 6\}$. Node 4 becomes an operating node and the weights of its directly connected nodes are taken into account. Node 4 is directly connected to node 2, node 3, node 5 and node 6, as shown in Figure 1.20, with the following weights: $d[2] = \min(62, 15 + 20)$, $d[3] = \min(20, 15 + 4)$, $d[5] = \min(\infty, 15 + 18)$ and $d[6] = \min(\infty, 15 + 5)$.

According to the formula (1.2), the new shortest paths are found and checked if there is any improvement. After the comparison of newly computed and current shortest paths from node 1 to the particular nodes, the shortest paths are selected and assigned to the particular nodes (node 3 and node 4).

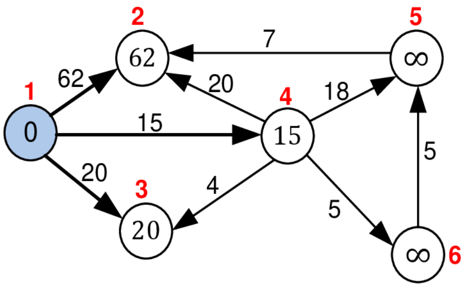


Figure 1.20 First stage of Dijkstra's algorithm, S = {1, 4}

Second iteration

In the second iteration, node 3 is selected ($d[3] = 19$) as an operating node and moved to the list S: $S = \{1, 4, 3\}$ and $S' = \{2, 5, 6\}$. For the rest of the nodes, the weights remains and no improvement is necessary because no paths come from the node 3: $d[2] = 35, d[5] = 33, d[6] = 20$. Figure 1.21 illustrates the second stage of Dijkstra's algorithm of this example.

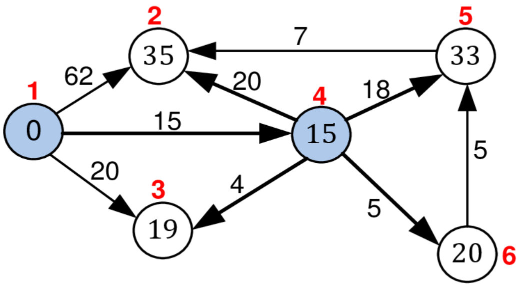


Figure 1.21 Second stage of Dijkstra's algorithm, S = {1, 4, 3}

Third iteration

Node 6 is selected as an operating node because $d[6] = 20$ and represents the smallest value among the non-considered nodes. Thus, $S = \{1, 4, 3, 6\}$, $S' = \{2, 5\}$ and the weights of node 2 and node 5 remain: $d[2] = 35, d[5] = 33$ as shown in Figure 1.22. This iteration is completed.

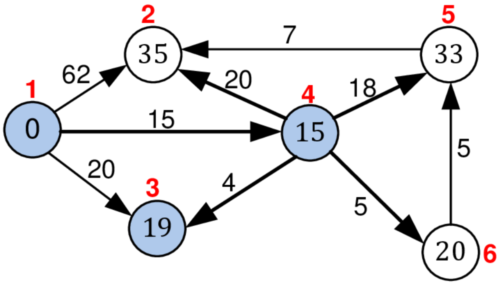


Figure 1.22 Third stage of Dijkstra's algorithm, S = {1, 4, 3, 6}

Fourth iteration

In this iteration, the only improvement is related to node 5. A new shortest path from the source node to the node 5 is found and assigned: $d[5] = \min(33, 20 + 5)$. The node 5 is selected and moved to the list S: $S = \{1, 4, 3, 6, 5\}$ and node 2 ($d[2] = 35$) remains in S'. Figure 1.23 illustrates the fourth iteration of Dijkstra's algorithm.

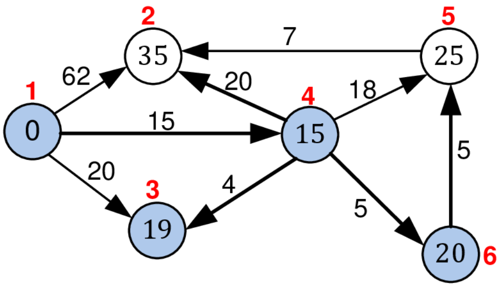


Figure 1.23 Fourth stage of Dijkstra's algorithm, S = {1, 4, 3, 6, 5}

Fifth iteration

This iteration is the final iteration because the last node 2 is moved to the list S and the list S' becomes empty: $S = \{1, 4, 3, 6, 5, 2\}$, $S' = \{\,\}$. The process is continued as before. A new shortest path from the source node to the node 2 is found: $d[2] = \min(35, 25 + 7)$. As shown in Figure 1.24, after the fifth iteration the algorithm is completed.

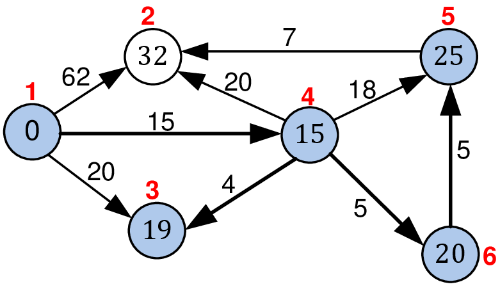


Figure 1.24 Fifth stage of Dijkstra's algorithm, S = {1, 4, 3, 6, 5, 2}

Final stage

Table 1.5 summarizes the initial stage and all the iterations until all nodes are considered in the list S. Figure 1.25 gives a completed visual illustration of Dijkstra's algorithm run on the example.

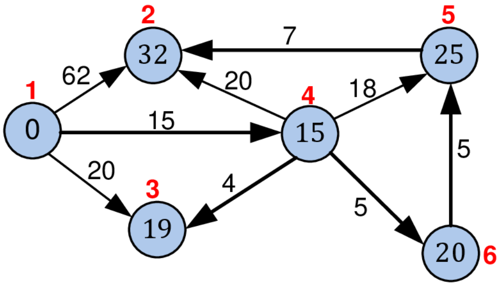


Figure 1.25 The final stage of Dijkstra's algorithm

For any and all subsequent purposes, the resulting vertex s after the last iteration of Dijkstra's algorithm is referenced to as chain alpha.

Table 1.5 Summarization of Dijkstra's algorithm computation

Iteration	S	*d*[2]	*d*[3]	*d*[4]	*d*[5]	*d*[6]
Initial	{1}	62	20	15	∞	∞
1	{1, 4}	35	19	15	33	20
2	{1, 4, 3}	35	19	15	33	20
3	{1, 4, 3, 6}	35	19	15	33	20
4	{1, 4, 3, 6, 5}	35	19	15	25	20
5	{1, 4, 3, 6, 5, 2}	32	19	15	25	20

1.2.3 Integrated Intermediate System-to-Intermediate System

The base specification of IS-IS protocol was defined in ISO/IEC 10589 as an international standard. The intra-domain protocol is able to operate in OSI Connection-less Network Service (CLNS). CLNS is a service provided by Connection-less Network Protocol (CLNP) which is a network layer protocol defined in OSI. To relate IS-IS to an IP environment, RFC 1195 [28] was released. The resulting Integrated IS-IS protocol is able to perform routing for OSI protocol stacks and IP simultaneously. The protocol is designed to support traffic to IP hosts, OSI end systems and both traffics.

The protocol is quite similar to OSPF. It is designed to provide IGP functionality. Integrated IS-IS uses the Dijkstra's algorithm to calculate the router's routing table from the collected Link State Packets [29], [30].

As in OSPF, IS-IS uses cost as a metric. Integrated IS-IS defines two types of metrics: internal metric and external metric. The internal metric is used within the routing domain while the external metric is used outside the domain. A route using internal metric is always preferred to a route using external metric.

IS-IS packet formats

Integrated IS-IS protocol uses three different packet formats:

- **Hello packets** – These packets are used to establish and maintain adjacencies between IS-IS neighbors.
- **Link State Packets** – These packets are used to exchange link state information between IS-IS nodes.
- **Sequence Number Packets** – These packets provides the control functions and mechanisms for the synchronization of databases.

Each type of IS-IS packet has a complex format with different logical parts. Each packet consists of a header, additional header fields and a number of variable-length fields. Common header consists of eight octets containing variable Type-Length-Value (TLV) fields. Figure 1.26 illustrates a packet format shared by all IS-IS packets.

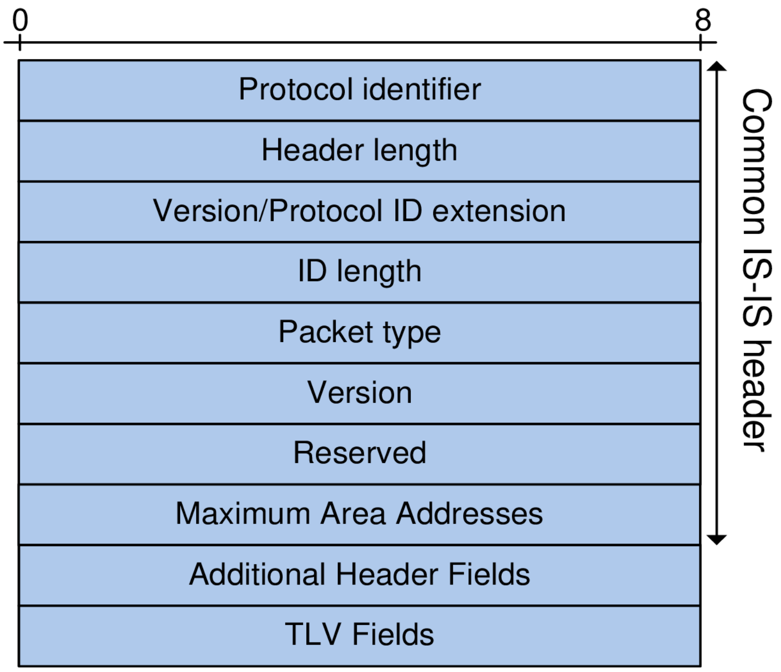


Figure 1.26 IS-IS packet format

The fields of the common IS-IS header have the following meaning:

- **Protocol identifier** – This field identifies the IS-IS protocol.
- **Header length** – Identifies the length of header in octets.
- **Version/Protocol ID extension** – Currently contains a value of 1 in IS-IS specification.
- **ID length** – This field contains the size of the source ID.
- **Packet type** – Defines the type of IS-IS packet.
- **Version** – Again a value of 1 after the Packet type.
- **Reserved** – Indicates unused bits (all set to zero).
- **Maximum Area Addresses** – Specifies the number of addresses in this area.

The attributes of TLV are as follows:

- **Type** – A binary code that indicates a specific TLV.
- **Length** – Indicates the length of the TLV in octets.
- **Value** – Indicates the content of the TLV.

1.2.4 Interior Gateway Routing Protocol

The Interior Gateway Routing Protocol is a proprietary routing protocol developed by Cisco Systems [31]. It represents a distance vector Internal Gateway Protocol as RIP. In comparison with RIP, IGRP is designed for larger and more complex networks. IGRP also overcomes some limitations of RIP, such as small hop-count limit and a single metric.

IGRP metric

IGRP uses a composite metric consisting of following separate parameters:

- **Delay** – Specifies the delay of all the links to a destination. Links with lower end-to-end delay are preferred. Delay variable has the unit of tens of microseconds.
- **Bandwidth** – Indicates the value of transfer rate of the link in kilobits. The bandwidth variable is defined as 10^7/bandwidth associated with the link by the router or administrator in kilobits per second.
- **Reliability** – Indicates the reliability of the link in the terms of transmission errors where the more reliable link means the better path.
- **Load** – Stands for a variable value related to the utilization of the link. Load value varies between 1 and 255 where the value of 255 represents congestion.

The formula of a composite metric is as follows:

$$IGRP\ metric = (K_1 \cdot B + \frac{K_2 \cdot B}{256 - L} + K_3 \cdot D) \cdot \left(\frac{K_5}{R + K_4}\right) \tag{1.3}$$

where K_1, K_2, K_3, K_4 and K_5 are constants, B is bandwidth, L is load, D is delay and R is reliability. Default values of these constants are $K_1 = K_3 = 1$ and $K_2 = K_4 = K_5 = 0$ (not used). The formula (1.3) may be simplified into:

$$IGRP\ metric = B + D. \tag{1.4}$$

From the equation (1.3) can be seen that if $K_5 = 0$ then IGRP $\text{metric} = 0$, thus, the term $\frac{K_5}{R+K_4}$ is ignored in the equation [31], [32].

IGRP timers

It is important to define several timers for distance vector protocol to control the ability of learning and deleting routes. There are four time constants: Update, Invalid, Hold-down and Flush. A routing update is broadcasted every 90 seconds. A route becomes invalid if it is not updated after 270 seconds from the last update. When a destination becomes unreachable, no new path is accepted for this destination for 280 seconds. After 630 seconds is the invalid route removed from the routing table (flush timer).

IGRP packet format

Figure 1.27 shows the IGRP packet format containing two entries. The packet consists of a common header and individual route entries. Each packet can carry up to 104 entries. No field is unused.

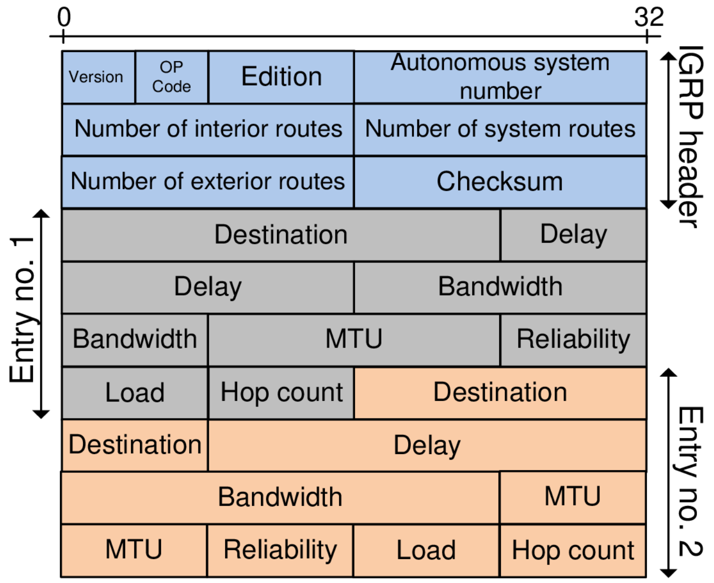


Figure 1.27 IGRP packet format

The fields of the IGRP header are as follows:

- **Version** – Is always set to one.
- **OP Code** – Differs between IGRP Request packet and IGRP Update packet.
- **Edition** – Represents a counter that avoids accepting an old update.
- **Autonomous system number** – This field defines the ID number of the IGRP process.
- **Number of interior routes** – This field indicates the number of entries in an update message that are subnets of a directly connected network.
- **Number of system routes** – This field is similar to the previous one and indicates the number of entries that are not directly connected.
- **Number of exterior routes** – This field indicates the number of entries of default networks.
- **Checksum** – The checksum is used to check the IGRP header and all the entries.

The fields of the IGRP entry are as follows:

- **Destination** – This field represents the destination network.
- **Delay** – Indicates the total sum of all links to the destination.
- **Bandwidth** – Represents the value of the transfer rate of the link in kilobits, as explained before.
- **MTU** – Represents the smallest MTU of any link along the route to the destination.
- **Reliability** – The value reflects the total outgoing error rates of the interfaces along the route to the destination.
- **Load** – The value reflects the total outgoing load of the interfaces along the route to the destination.
- **Hop count** – Indicates the number of hops along the route to the destination [32].

1.2.5 Enhanced Interior Gateway Routing Protocol

Because of the limitations of IGRP and RIP, an enhanced version of IGRP was defined. This enhanced version is called Enhanced Interior Gateway Routing Protocol. EIGRP is a proprietary routing protocol developed by Cisco Systems. Since EIGRP is not an open protocol, no RFCs are available.

EIGRP can be considered an advanced distance vector routing protocol. The routing protocol uses a Diffusing Update Algorithm (DUAL) and it should be differentiated from classical distance vector routing protocols. Thus, Cisco Systems has called it „hybrid" protocol. It combines the advantages of link state routing protocols and distance vector routing protocols [21].

DUAL is a finite-state machine that uses diffusing computations to perform the shortest path routing while selecting loop-free path to each destination.

EIGRP metric

EIGRP uses the same equation (1.3) like IGRP to calculate the metric. However, there is a change. This change results from the size of the Metric field. IGRP Metric field is 24 bits large and EIGRP Metric field is 32 bits large. EIGRP metric can be obtained by multiplying the result of the IGRP metric by a value of 256 (the difference of 8 bits). The EIGRP metric can be calculated according to the following formula:

$$EIGRP\ metric = \left(K_1 \cdot B + \frac{K_2 \cdot B}{256 - L} + K_3 \cdot D\right) \cdot \left(\frac{K_5}{R + K_4}\right) \cdot 256 \tag{1.5}$$

where, as mentioned before, K_1, K_2, K_3, K_4 and K_5 are constants, B is the bandwidth, L is the load, D is the delay and R is the reliability. Default values of these constants are $K_1 = K_3 = 1$ and $K_2 = K_4 = K_5 = 0$ (not used) [21], [33]. The formula (1.5) may be simplified into:

$$EIGRP\ metric = (B + D) \cdot 256. \tag{1.6}$$

EIGRP packet format

EIGRP uses five different types of packets: Hello, Update, Query, Reply and Acknowledgement. Hello packets are used to discover EIGRP neighbors. Update packets are sent to EIGRP neighbors when a new route is discovered or when topology synchronization is needed. To achieve fast convergence, Query packets are sent to neighbors to search for the lost routes. Reply packets are sent as a response to the Query packets.

Each EIGRP packet, as shown in Figure 1.28, consists of a common header followed by TLVs carrying route entries. The most common TLVs are: EIGRP parameter TLV, the EIGRP IP internal route TLV and the EIGRP IP external route TLV. The EIGRP parameter TLV carries the values of K constants and hold time. The internal route TLV and the external route TLV contain one route entry each and metric information for the route. There are other types of TLVs, for example AppleTalk specific TLV, IPX specific TLV and so on [21].

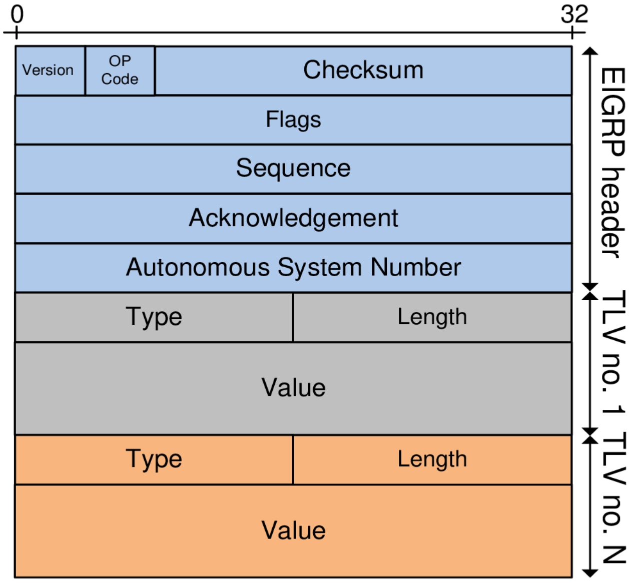


Figure 1.28 EIGRP header format followed by TLVs

The fields of the EIGRP header are as follows:

- **Version** – Specifies the version of EIGRP.
- **OP Code** – Specifies the type of packet. OP Code 1 specifies the Update packet, OP Code 3 specifies the Query packet, OP Code 4 specifies the Reply packet and OP Code 5 specifies the Hello packet.
- **Checksum** – The checksum is used to check the EIGRP packet except the IP header.
- **Flags** – This field defines two flags: Init and Conditional receive.
- **Sequence** – This field specifies the sequence number.
- **Acknowledgement** – This field is used to acknowledge packets.
- **Autonomous System Number** – This field specifies the identification of the EIGRP domain.

The fields of EIGRP TLVs are as follows:

- **Type** – This field specifies the type of TLV.
- **Length** – This field specifies the length of TLV.
- **Value** – This field specifies the content of TLV.

1.2.6 Exterior Gateway Protocol

Exterior Gateway Protocol is the oldest protocol used for interconnection between two different autonomous systems. EGP was replaced by other Exterior Gateway Protocol called Border Gateway Protocol and is not used any more. However, for the purpose of an introduction, the protocol is briefly mentioned.

EGP was formally specified in RFC 904 [34]. The protocol was used to convey network reachability information. It was more a reachability protocol than a routing protocol. Without any metrics, the routing decisions were made by EGP routers maintaining a database of reachable networks. The information about how to reach the routers crossed the entire Internet.

1.2.7 Border Gateway Protocol

The first version of Border Gateway Protocol (BGP) was defined in RFC 1105 [35] followed by several versions containing modifications of previous ones. The newest version, version 4, is described in RFC 1771 [36] and updated in RFC 4271 [37]. BGP is the most widely-used Exterior Gateway Protocol. It is a distance-path protocol which provides routing between multiple autonomous systems or domains. BGP uses TCP as a reliable transport protocol.

The function of BGP is to exchange information about network reachability, routing decisions and other rules based on paths with its neighbors. The path is a list of every AS between the router and the destination. Every router receives the reachability information from its neighbors and maintains a routing table that shows all suitable paths to a particular network. The idea is that networks are not interested in the inner details, BGP connects network and AS together. BGP uses different attributes in route selection process. The value of BGP attribute, such as next hop, weight, path length, is assigned to the link. The protocol is usually used by Internet Service Providers (ISP) [36].

<u>BGP packet format</u>

BGP uses a common packet format consisting of a header and a body, as shown in Figure 1.29. The body is variable according to the type of BGP packet.

BGP uses four different types of packets: Open, Update, Notification and Keep-alive. The Open packet is the first sent packet used to establish a connection between BGP sides. The procedure involves identification and initiation of a link and negotiation of session parameters. An Update packet provides routing updates to other BGP devices. The Notification packet is used in the case of error condition or closed connection. The Keep-alive packets are sent periodically to make sure that a device is still operational.

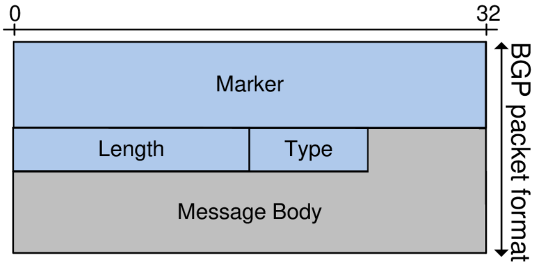


Figure 1.29 BGP packet format

The fields of the BGP packet are as follows:

- **Marker** – This 16-byte large field optionally contains authentication and synchronization.
- **Length** – Indicates the total length.

- **Type** – Specifies the type of the BGP packet.

BGP Update packet format

BGP Update packet is one of the most complex packet structures in the TCP/IP system. The packet is used to transfer routing information between BGP devices. It allows a BGP device to create new routes, modify existing ones and withdraw invalid ones. The packet format consists of the header and fields optionally included in the body. The fields are as follows: Unfeasible Routes Length, Withdrawn Routes, Total Path Attribute Length, Path Attributes and Network Layer Reachability Information.

1.2.8 Comparison of routing protocols

Each routing protocol has some advantages and some disadvantages in selecting the best path if multiple routes are available. Table 1.6 summarizes mentioned routing protocols except unused EGP.

Table 1.6 Comparison of routing protocols

Protocol	Category	Metric	Algorithm	Transport
RIPv2	distance vector	hop count	Bellman-Ford	UDP port 520
OSPF	link state	cost	Dijkstra's	IP Protocol 89
IS-IS	link state	manual cost	Dijkstra's	Layer 2
IGRP	distance vector	composite	Bellman-Ford	IP Protocol 9
EIGRP	hybrid	composite	DUAL	IP Protocol 88
BGP	path vector	multiple attributes	Path selection	TCP port 179

RIP was one of the first routing protocols used for intradomain routing. The advantages of RIP are its easy configuration and implementation. Shortcomings of the RIP include hop-count limitation of 15 hops, slow convergence time and increased network traffic caused by updates. RIP also hardly manages large networks. OSPF was designed to overcome some limitations of RIP. It provides quick convergence and more intelligent routing metrics.

Another protocol that uses shortest path first algorithm and performs routing for the IP and OSI protocol stack simultaneously is called Integrated IS-IS. Integrated IS-IS resembles OSPF in many ways but it is not widely deployed in the Internet.

IGRP and EIGRP are Cisco proprietary protocols, but they have much in common with previous routing protocols. IGRP is based on RIP designed for larger and more complex networks. Cisco protocols use the same equation to calculate the metric but different algorithm for path selection. EIGRP has fixed some shortcomings of classical distance vector routing protocols and represents a robust routing protocol. EIGRP is an enhanced version of IGRP and has replaced IGRP in many systems. EIGRP provides very fast convergence, lower bandwidth utilization and loop-free forwarding table. Nevertheless, the configuration of EIGRP is very easy.

For sharing routing information between different ASs or domains, a flexible EGP protocol called BGP is currently used. The advantage of BGP is that it uses path attributes to implement routing policies but the deploying of BGP is costly and complex

1.3 Next-Generation Networks

Increasing requirements for personalization, consumer's mobility and agile services call for a new communication environment. Consumers are looking for easier and better ways of reaching out to each other over whatever terminal or access technologies, which are available at the moment. Users want to share their latest experiences anywhere and anytime, therefore the network infrastructure must provide sufficient network resources for high-value services. A natural way to fulfill these demands is the evolution towards an all-IP environment which appears to be a strong trend. This trend converges towards a Next-Generation Network (NGN) specified in ITU.T Y.2001 [39]. Various views on NGN have been expressed, however, the heart of the NGNs forms an IP-based network.

IP Multimedia Subsystem (IMS) represents such an architecture providing multi-access to required services and large-scale interoperability. The idea of IMS is to integrate traditional telecommunication services with the Internet Protocol. Therefore, this architecture uses two of the most successful representatives in communications, namely fixed/mobile networks and the Internet. The IMS technology originated from the Third Generation Partnership Project (3GPP) Release 5 specifications [40].

In spite of cellular networks providing mobility and a wide range of services, the main reason for creating IMS is to offer more than the mere Quality of Service (QoS) support. The term QoS is widely used in the telecommunication world today. All IMS solutions should guarantee the QoS that customers need and demand. Deployment of the new services depends on the QoS level that the IMS technology is capable to provide.

Since the IMS involves a large amount of protocols, it is important to define ways in which different end system can reach the end-to-end QoS for a connection. One of the questions is how to use lower layer QoS mechanism to achieve upper layer QoS within the network. There is a need to ensure the interoperability among different layers, domains and networks. The network capacity is currently adequate for the majority of applications. In spite of that, it often happens that the user's perceived qualities of network traffic characteristics are not satisfactory. To provide end-to-end QoS, it is necessary to manage the QoS within each domain along the path [61], [62], [63], [64].

In the IP domains, there are some well-known mechanisms for QoS provisioning, such as Differentiated Services (DiffServ) and Integrated Services (IntServ). The utilization of these mechanisms in the IMS is still an open issue. The end-to end QoS in the IMS architecture introduces several challenges which have to be faced [59], [60].

2 THESIS OBJECTIVES

The Internet or internetworking communication systems are based on universal network-level interconnections. A network-level interconnection enables delivering of data units from their original source to its desired destination. Internetworking also deals with the complexity of various underlying communication technologies that makes such interconnected networks function. The Internet Protocol plays a central role and represents the most important protocol in the Internet architecture. All network components that operate at the network layer or higher layers use the Internet Protocol. The protocol is responsible for the exchange and evaluation of necessary information needed for the appropriate routing.

Since OSPF currently represents one of the most widely used routing protocol, any valuable improvement to keep pace with the rapidly changing Internet environment would be greatly appreciated. As the name implies, OSPF is an open standard and anyone can intervene. Modern routing domains need to maintain a very high level of service availability but with the growing demands of users, the data networks may become heavily loaded and data links congested. Detailed description is mentioned in Figure 3.1. Such congestion causes performance degradation and in some cases also service disruption. OSPF networks are often heavily loaded which leads in redundancy in the form of network devices or communication mediums installed within the infrastructure. It is a method of ensuring network availability and continuity of services in the case of unplanned failure. In complex networks, it is often that similar or almost equivalent paths exist toward a destination. Thus, the mentioned alternative paths may provide a potential in order to balance traffic in the best way between multiple paths. The question is how to balance the network traffic between various paths when the particular link becomes heavily loaded.

The main goal of the thesis is to propose, verify and analyze more efficient mechanism to improve routing in OSPF networks. To achieve this purpose, a new complex solution that extends current methods of the OSPF routing is examined. A novel approach to calculate OSPF metric will split the traffic from the congested links in multiple paths, if necessary, according to involved logic. The method considers link load as an additional parameter for a final metric calculation. This solves the problem of absence of traffic awareness and inconveniently congested links; what decreases network utilization and increases network performance.

The goal is to keep the traffic load on all links at acceptable levels if it is possible from the network infrastructure point of view. To realize the main goal of the thesis, some partial aims need to be accomplished that are summarized in following points:

- To perform detailed analysis of routing protocols, especially OSPF, and adopt all its features and principles. Processed in Chapter 1.
- To illustrate the shortcomings of OSPF protocol, simple testbed for experimentations will be created. Processed in Chapter 3.

- To propose a novel method for link load sensitive metric strategy. The proposal will be based on mathematical functions leading to improve routing. Processed in Chapter 4.
- The method proposed will be investigated in Matlab environment. The collected data from simulation sets will be analyzed and evaluated. Processed in Chapter 5.
- To transfer the theoretical plane into practical form and to allow transparent and replicable testing of method proposed, a testbed will be used. This environment will correspond to simulated network structure. This procedure will ensure compatibility and prove theoretical expectations. Processed in Chapter 6.
- To prove and verify the benefits of new method. The data collected during simulations and experimental testing will be compared, analyzed and evaluated. Processed in Chapter 7.

3 DEFAULT OSPF ROUTING ALGORITHM

The limitations of default OSPF are described on a testing architecture. The architecture is based on the explanatory example mentioned in Figure 3.1. For instance, if there are two links with the same bandwidth and the utilization of the first link is very low and for the second link very high, OSPF assigns both links the same metric. This topology consists of six routers and users or servers. This example and other configurations are discussed with the emphasis on IPv4. Most of the routers are connected with Fast Ethernet link which corresponds to the cost of 1. Standard Ethernet technology is used to connect Router E with Router B. Assigned metric between these routers is 10. In this scenario, User 1 and User 2 communicate to the servers. Based on the composite metric, calculated as a sum of individual metrics on the path to the destination, the entire data traffic passes the link between Router D and Router E. There is no traffic on paths from Router B to Router E and from Router I to Router E. The reason is simple. Current OSPF characteristic leads to use the shortest path, which means the path from Router D to Router E. With increasing users' demands on the servers, a particular part of the network, the link between Router D and Router E, may become congested. Such congestion causes performance degradation and in some cases also service disruption.

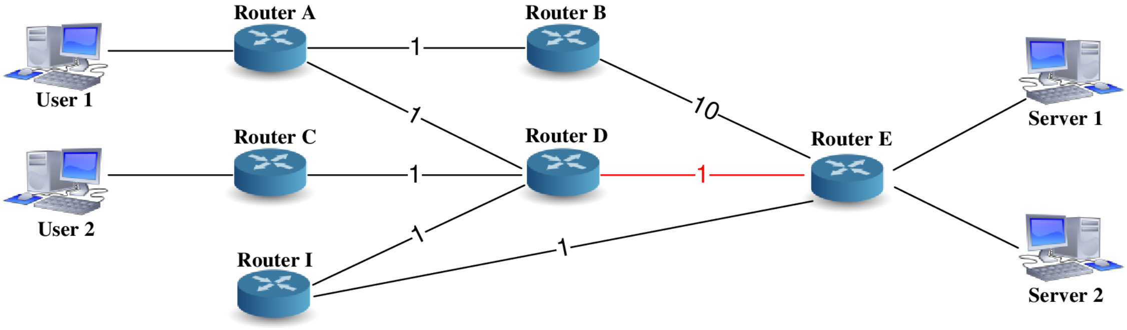


Figure 3.1 An example of possible congestion

Some attempts to provide improved routing were proposed in the past. The recent works related to routing algorithms improvement [45], [46], [47], [48] can be used to choose a path with specific bandwidth requirements. New Cost Adaptive OSPF (CA-OSPF) is proposed in [44]. However, this solution is not applicable for heavily loaded networks where the CA-OSPF cannot improve the network performance. An implementation of QoS routing extensions to the OSPF is proposed in [49]. Also some balance heuristics were proposed to avoid congestion and utilize low loaded links [50], [51]. However all of these solutions focus only on the case of full link congestion and there is no rerouting of traffic until the congestion appears. Also, if the demands on traffic changes rapidly, another approach is needed. Finally the convergence is reached when this load balancing gets to a stabilized state, but the convergence time is very long, reaching link congestion a few times on different links in the process. This equilibrium is again broken when the amount of traffic changes and convergence process has to start all over.

3.1 Testbed with default OSPF routing algorithm

To illustrate the shortcomings of OSPF protocol in detail, simple testbed for experimentations is created. When designing a new network, it is important to document such network. At a minimum, the documentation should include a topology diagram that indicates the physical connectivity and an addressing table that lists all of the following information: device names, interfaces used in the design, IP addresses and subnet masks and finally default gateway addresses for end devices, such as PCs. The physical layout of the test network with default OSPF routing algorithm is shown in Figure 3.2.

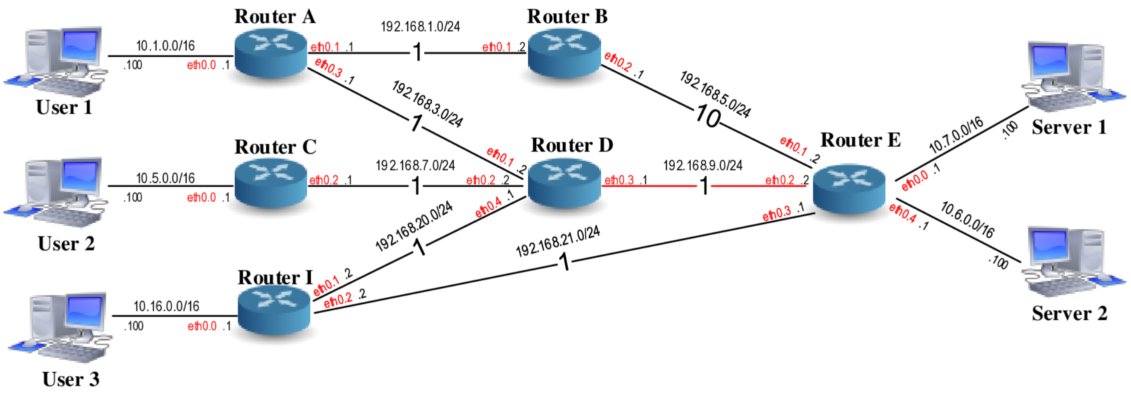


Figure 3.2 Testbed with default OSPF routing algorithm

The IP addressing scheme for the topology shown in Figure 3.2 is summarized in Table 3.1. The table contains network addresses and subnet masks of the interfaces, along with the interface types and numbers. A PC such as User 1 or Server 1 is normally configured with a single host IP address because it only has a single network interface, usually an Ethernet NIC. Routers have multiple interfaces, therefore, each interface must be a member of a different network. It can be seen that each router has five interfaces named from eth0.0 to eth0.4. The picture of configuration of routers is shown in Figure 3.3.

Figure 3.3 Configuration of routers in testbed with default OSPF routing algorithm

Table 3.1 Addressing scheme used in testbed with default OSPF routing algorithm

Device	Interface	IP address	Netmask	Default Gateway
Router A	eth0.0	10.1.0.1	255.255.0.0	N/A
	eth0.1	192.168.1.1	255.255.255.0	N/A
	eth0.2	N/A	N/A	N/A
	eth0.3	192.168.3.1	255.255.255.0	N/A
	eth0.4	N/A	N/A	N/A
Router B	eth0.0	N/A	N/A	N/A
	eth0.1	192.168.1.2	255.255.255.0	N/A
	eth0.2	192.168.5.1	255.255.255.0	N/A
	eth0.3	N/A	N/A	N/A
	eth0.4	N/A	N/A	N/A
Router C	eth0.0	10.5.0.1	255.255.0.0	N/A
	eth0.1	N/A	N/A	N/A
	eth0.2	192.168.7.1	255.255.255.0	N/A
	eth0.3	N/A	N/A	N/A
	eth0.4	N/A	N/A	N/A
Router D	eth0.0	N/A	N/A	N/A
	eth0.1	192.168.3.2	255.255.255.0	N/A
	eth0.2	192.168.7.2	255.255.255.0	N/A
	eth0.3	192.168.9.1	255.255.255.0	N/A
	eth0.4	192.168.20.1	255.255.255.0	N/A
Router E	eth0.0	10.7.0.1	255.255.0.0	N/A
	eth0.1	192.168.5.2	255.255.255.0	N/A
	eth0.2	192.168.9.2	255.255.255.0	N/A
	eth0.3	192.168.21.1	255.255.255.0	N/A
	eth0.4	10.6.0.1	255.255.0.0	N/A
Router I	eth0.0	10.16.0.1	255.255.0.0	N/A
	eth0.1	192.168.20.2	255.255.255.0	N/A
	eth0.2	192.168.21.2	255.255.255.0	N/A
	eth0.3	N/A	N/A	N/A
	eth0.4	N/A	N/A	N/A
User 1	N/A	10.1.0.100	255.255.255.0	10.1.0.1
User 2	N/A	10.5.0.100	255.255.255.0	10.5.0.1
User 3	N/A	N/A	N/A	N/A
Server 1	N/A	10.7.0.100	255.255.255.0	10.7.0.1
Server 2	N/A	10.6.0.100	255.255.255.0	10.6.0.1

3.2 Network design

The Network layer transports the packets between the hosts with as little impact on the network performance as possible. The layer does not care about the type of data contained inside of a packet. This responsibility belongs to related upper layers. The upper layers decide about reliability of the data transmission. The TCP/IP communication process includes several steps. User data is created at the application layer. The segmentation and encapsulation of data is processed on its way down the protocol stack. Also address identifiers are added to the data. The data is then transported through the network and received on the destination side. The decapsulation and reassembly process prepares data for the application layer of destination.

Routing is done packet-by-packet and hop-by-hop. Each packet is independently routed through the network towards the destination. The packet is autonomous since it contains all the necessary information related to addressing. No packet can be forwarded without a route. When a network topology is created, all the routers learn about its own directly connected networks. This is achieved when the interfaces are correctly configured and activated. The interface information includes an IP address, mask, type of network, metric and a neighbor router. Directly connected networks are the primary networks for routing decisions. If there is no route determined and no default route, the packet is discarded.

All mentioned processes are described on a specific example in this chapter in detail.

3.2.1 Users and servers

There are three users running OS Windows 7 and two servers running OS Windows Server 2012 r2. Users use FTP clients and Windows Servers use FTP servers for data transmission. User 3 is mostly used for other testing, monitoring and capturing packets.

Network configuration is mentioned in Table 3.1. Each device is in a different network and has its own gateway. The last octet of its IP address is always set to 100. Network packet analyzer called Wireshark is installed on each device. Wireshark is used to learn and debug network protocols and to troubleshoot network problems. Monitoring tool called PRTG installed on Server 1 is used for monitoring network traffic.

3.2.2 Routers

There are several types of routers. When selecting a device, number of factors such as cost, speed, type of interfaces, expandability, manageability and additional features need to be considered. The cost of any routing device depends on its capacity, features and optional technologies. When selecting a router, the number of ports mean a critical decision. For the purpose of the thesis, OpenWrt operating system compatibility is necessary. The relevant type of router is Linksys WRT54GL version 1.1. The WRT54GL is a Wireless Access Point (WAP), broadband router and firewall with 4+1 ports. The device provides wireless access support for 802.11b/g products but the wireless technology is not used at all in this thesis. The manufacturer claims 54 Mbps bandwidth. According to the real measurement, the maximum bandwidth matches 54 Mbps. The router has the following features:

- 200 MHz 32 bit processor,
- 16 MB of RAM,
- 4 MB of flash memory [53].

Simplified bootup process focused on loading of configuration files run as follows. After the operating system is loaded, the startup-config file in Nonvolatile RAM (NVRAM) is copied into RAM and stored as the running-config file. NVRAM (Nonvolatile RAM) does not lose its information when power is turned off, while RAM is volatile memory and loses its content when the router is turned off. Operating system executes the configuration commands in the running-config. Any changes entered by the administrator are saved in the running-config and implemented by the operating system.

Its firmware is based on Linux. However, the firmware directly installed to the router is not open for testing. The firmware is not open-source and no modifications can be done. For the usage of basic router's functionality, the original Linksys firmware is fine. However, for the purpose of this thesis, a third party firmware supporting extra functionality has to be considered. GNU/Linux distribution called OpenWrt is used. OpenWrt is an open-source operating system providing a fully writable file system. The OpenWrt distribution is meant to be customizable and flexible. It supports several router models from various vendors. The idea is to access crucial functionalities of the device that are not originally available. This system enables the implementation of algorithm proposed to a real device.

It is important to install all the components correctly. One of the options is to use Linksys web Graphical User Interface (GUI). Before upgrading new firmware, it is necessary to download an image called *open-wrt54g-squashfs.bin*. The next step is to upload this firmware image to the router. After reboot, the router is available at *luci* interface *http://192.168.1.1* or Telnet. After a new password has been set, Telnet is disabled and secure data communication through Secure Shell (SSH) is enabled. The installation is complete and the new firmware is ready to use. However, this basic firmware does not support any routing technique. The solution is to use Quagga routing software.

Quagga represents a routing software offering implementations of routing protocols such as OSPF, RIP, IS-IS and BGP. This routing software suite is based on an original routing suite and represents a fork of GNU called Zebra. Quagga is distributed under the GNU GPL and supports an interactive user interface. The original routing software is made as a process program providing all of the functionalities of a routing protocol. Quagga, however, covers a set of daemons working together to create a routing table. The configuration of the routing table can be changed and monitored dynamically. Quagga system architecture is described in Figure 3.4. The ospfd daemon supports the OSPF version 2 protocol. Zebra represents the kernel routing table manager for modifying the kernel routing table [52] [54].

The ospfd daemon contains its own configuration file and terminal interface. The source code is a complete implementation of the OSPF version 2. All the OSPF interface types and areas are supported.

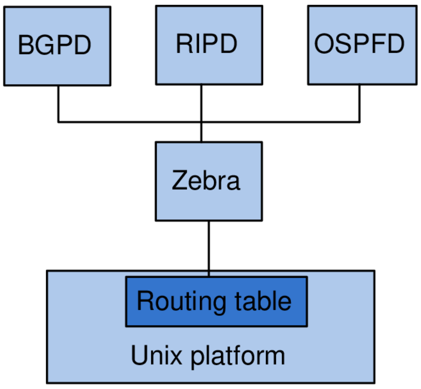


Figure 3.4 Quagga system architecture

To run ospfd daemon correctly, it is important to install all necessary packages (quagga and quagga-ospfd) from *backfire/10.03/brcm-2.4/packages/* respectively as follows:

librt_0.9.30.1-42_brcm-2.4.ipk,
quagga_0.98.6-5_brcm-2.4.ipk,
quagga-libzebra_0.98.6-5_brcm-2.4.ipk,
quagga-libospf_0.98.6-5_brcm-2.4.ipk,
quagga-ospfd_0.98.6-5_brcm-2.4.ipk.

Quagga has two user modes: Normal mode for observing and Administrator mode for changing the configuration. Quagga also provides integrated user interface shell *vtysh*. This tool connects ospfd daemon with UNIX domain socket. The daemon enables user command line for configuring and monitoring ospfd daemon. The package related to the *vtysh* is called:

quagga-vtysh_0.98.6-5_brcm-2.4.ipk.

The Quagga suite has several daemons to monitor, maintain and exchange routing information. For example *wathcquagga* daemon is the watchdog program for monitoring Quagga routing daemons status. *Zebra* daemon handles the kernel routing table management and redistribution among different routing protocols [54]. Each daemon has its own configuration file. The best way is to use the *vtysh* which access all the daemons at the same time.

OSPF configuration is done in OSPF configuration file *ospfd.conf* placed in the file */etc/ospfd.conf*. The file can be modified with any text editor. In this configuration file, routing daemon configurations and others can be defined. To make ospfd work, interface information from zebra is needed. From this reason, zebra has to be running before calling ospfd [52].

3.2.3 Router configuration

When configuring a router, some basic tasks are performed. Within the configuration belong naming the router, interface configuration and ospf routing configuration. The global configuration mode enables the configuration of global parameters or other configuration submodes (interface).

Below is an example of configuring Router A. The `enable` command is used to enter the privileged EXEC mode. This mode allows the user to make configuration changes on the router.

```
Router>enable
Router#configure terminal
Router(config)#hostname RouterA
```

The configuration of the interfaces is done by interface configuration commands. After the router's interface is configured and activated, it must be in "up" state. Once the interface is in this state, the network of that interface is added to the routing table as a directly connected network. For the mentioned topology, the eth0.1 interface needs to be configured as follows:

```
RouterA(config)#interface eth0.1
RouterA(config-if)#ip address 192.168.1.1 255.255.255.0
RouterA(config-if)#no shutdown
```

Before any static or dynamic routing is configured on Router A, the router only knows about its own directly connected networks. No static routes are used on any of used routers. To enable a dynamic routing protocol, the global configuration mode has to be entered and the `router` command has to be typed. The commands will start the OSPF routing process, set the identification parameter (router-ID) for the routing process and define the networks and area. All the routers are assigned to area 0. When configuring OSPF, the following syntax is used:

```
RouterA(config-router)# router ospf
RouterA(config-router)# ospf router-id 0.0.0.1
RouterA(config-router)# network 192.168.1.0 0.255.255.255 area 0
RouterA(config-router)# network 192.168.3.0 0.255.255.255 area 0
RouterA(config-router)# network 10.1.0.0 0.0.255.255 area 0
```

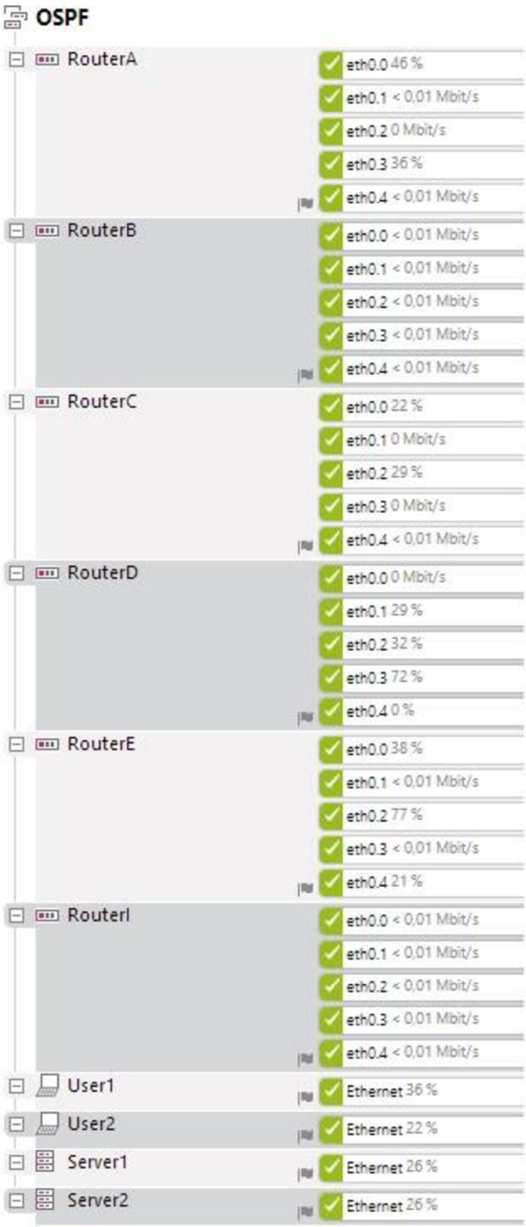
Detailed configurations of each router are summarized in Appendix A. The routing table of Router A shows, except directly connected networks, dynamically learned routes to the destination networks.

3.3 Verifying OSPF configuration

The physical layout of the test network mentioned in Figure 3.2 consists of six routers, three users and two servers. There are 12 networks connected by point-to-point links. The solid lines between the nodes represent communication links. Routers are using mainly Fast Ethernet technology and their cost corresponds to the metric of 1. Ethernet technology is used to connect Router E with Router B. Assigned metric between these routers is 10. Each router determines its own cost to each destination in the topology. In other words, each router calculates the SPF algorithm and determines the cost from its own perspective. This chapter focuses mostly on detailed description of Router A. To understand the whole scenario, routing tables from all the routers (Router A – Router I) has to be checked. These routing tables are summarized in Appendix A.

There is a standard method for configuring and monitoring IP protocols with the help of the Simple Network Management Protocol (SNMP) [55]. SNMP is a UDP-based network protocol to monitor network-attached devices. SNMP is often used to constantly monitor network traffic and the interfaces of the routers. SNMP uses UDP port 161 for sending and receiving requests. To name an object, the Object Identifier (OID) as unique identifier of managed device is used.

SNMP detects the increasing number of packets coming through one of the router's interface, suggesting that the interface is about to get congested.



OSPF

RouterA
- ✓ eth0.0 46 %
- ✓ eth0.1 < 0,01 Mbit/s
- ✓ eth0.2 0 Mbit/s
- ✓ eth0.3 36 %
- ✓ eth0.4 < 0,01 Mbit/s

RouterB
- ✓ eth0.0 < 0,01 Mbit/s
- ✓ eth0.1 < 0,01 Mbit/s
- ✓ eth0.2 < 0,01 Mbit/s
- ✓ eth0.3 < 0,01 Mbit/s
- ✓ eth0.4 < 0,01 Mbit/s

RouterC
- ✓ eth0.0 22 %
- ✓ eth0.1 0 Mbit/s
- ✓ eth0.2 29 %
- ✓ eth0.3 0 Mbit/s
- ✓ eth0.4 < 0,01 Mbit/s

RouterD
- ✓ eth0.0 0 Mbit/s
- ✓ eth0.1 29 %
- ✓ eth0.2 32 %
- ✓ eth0.3 72 %
- ✓ eth0.4 0 %

RouterE
- ✓ eth0.0 38 %
- ✓ eth0.1 < 0,01 Mbit/s
- ✓ eth0.2 77 %
- ✓ eth0.3 < 0,01 Mbit/s
- ✓ eth0.4 21 %

RouterI
- ✓ eth0.0 < 0,01 Mbit/s
- ✓ eth0.1 < 0,01 Mbit/s
- ✓ eth0.2 < 0,01 Mbit/s
- ✓ eth0.3 < 0,01 Mbit/s
- ✓ eth0.4 < 0,01 Mbit/s

User1
- ✓ Ethernet 36 %

User2
- ✓ Ethernet 22 %

Server1
- ✓ Ethernet 26 %

Server2
- ✓ Ethernet 26 %

Figure 3.5 List of all routers' interfaces in PRTG monitoring tool

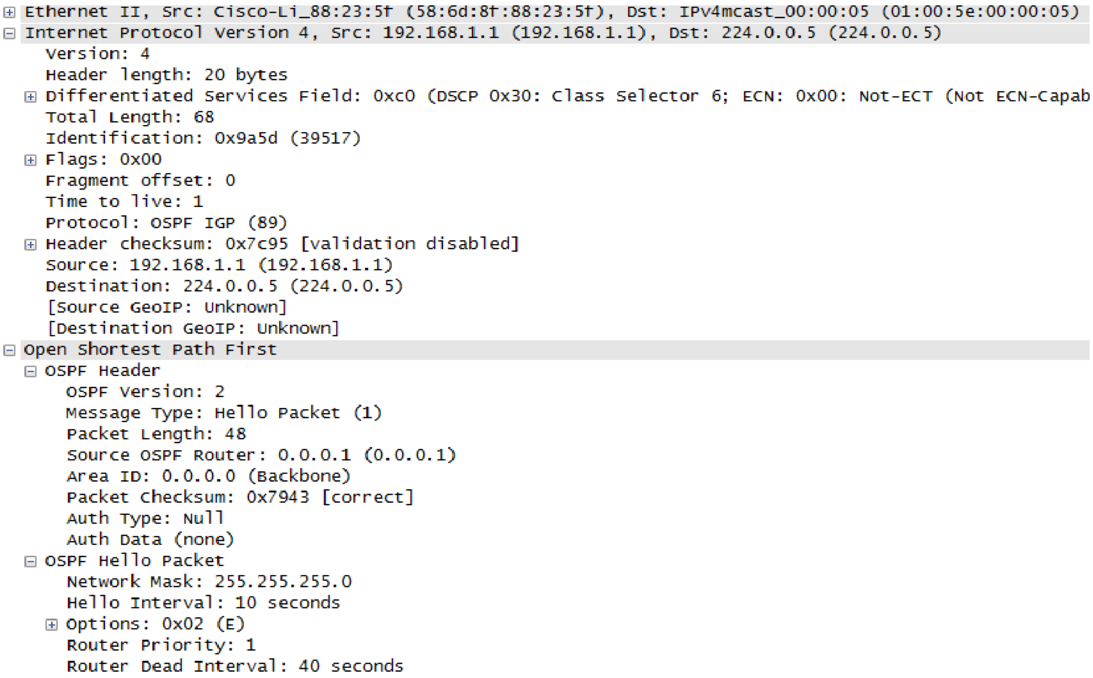
Figure 3.5 shows all devices and interfaces monitored by PRTG monitoring system based on SNMP. This figure shows the percentage of utilization of individual interfaces and can be estimated which interfaces are used and which interfaces remain inactive during the data transmission. For example, idle interfaces are marked as 0 Mbit/s or < 0.01 Mbit/s. Only one network interface is monitored of each user and server.

3.3.1 Troubleshooting

Once the testbed is operational, it is time to monitor its performance. This chapter includes various troubleshooting methods and tools.

The process of neighbor and adjacency initialization is started with Hello protocol. Router A sends Hello packets to discover if there are any neighbors. Router B and D reply with their own Hello packets and form an adjacency. Hello packets are frequently sent between the routers to monitor their state. If no Hello packets arrive, the router is considered "down".

Figure 3.6 shows captured OSPF Hello packet sent by Router A. As mentioned in Chapter 1.2.2, the multicast IP address is 224.0.0.5. From the packet can be seen configured intervals, router ID (0.0.0.1) and area ID is set to 0. Before two OSPF routers form an adjacency, they must agree on three values: Hello interval, Dead interval, and network type. OSPF Hello packets are sent every 10 seconds and dead interval is set to 40 seconds.



```
⊞ Ethernet II, Src: Cisco-Li_88:23:5f (58:6d:8f:88:23:5f), Dst: IPv4mcast_00:00:05 (01:00:5e:00:00:05)
⊟ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 224.0.0.5 (224.0.0.5)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0xc0 (DSCP 0x30: Class Selector 6; ECN: 0x00: Not-ECT (Not ECN-Capab
    Total Length: 68
    Identification: 0x9a5d (39517)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 1
    Protocol: OSPF IGP (89)
  ⊞ Header checksum: 0x7c95 [validation disabled]
    Source: 192.168.1.1 (192.168.1.1)
    Destination: 224.0.0.5 (224.0.0.5)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
⊟ Open Shortest Path First
  ⊟ OSPF Header
      OSPF Version: 2
      Message Type: Hello Packet (1)
      Packet Length: 48
      Source OSPF Router: 0.0.0.1 (0.0.0.1)
      Area ID: 0.0.0.0 (Backbone)
      Packet Checksum: 0x7943 [correct]
      Auth Type: Null
      Auth Data (none)
  ⊟ OSPF Hello Packet
      Network Mask: 255.255.255.0
      Hello Interval: 10 seconds
    ⊞ Options: 0x02 (E)
      Router Priority: 1
      Router Dead Interval: 40 seconds
```

Figure 3.6 Captured Hello packet

After the adjacencies are established, the LSA flooding is delivered to all the routers. When all link state information has been flooded to all routers in an area and the network is stable, OSPF is a quiet protocol. Hello packets are exchanged between neighbors every 10 seconds and periodic refresh of LSAs happens every 30 minutes. Link state protocol uses regular updates as a way of ensuring information. It is very important for link state databases to stay synchronized. Flooding process is implemented by LSU packets and acknowledged by LSAck packets separately. Detailed specification of the LSU packet is described in Chapter 1.2.2. However, it is important to mention that each LSU packet contains Router LSA carrying metric information.

Router A is now able to construct an SPF tree of the network. Router A is considered as a source router. The source router is the first router and becomes an operating router. Its Link State Database entries are examined first. Table 3.2 summarizes Router A’s simplified database including neighboring routers. To ensure correct and loop free routing, database synchronization is crucial in OSPF networks. Router A may skip the first record of LSP received by Router B because it already knows that it is connected to Router B on network 192.168.1.0/24. Router A can use the second record and create a link from Router B to Router E with the network 192.168.5.0/24 and a cost of 10. This new information is added to the SPF tree. Using the LSP from Router D, Router A may ignore the first record related to network 192.168.3.0/24. Router A has learned that Router D has networks 192.168.7.0/24 and 192.168.20.0/24 and creates new links. However, shorter path is found to Router E using link 192.168.9.0/24 and added to the SPF tree. Router A has now constructed the complete SPF tree. Link 192.168.5.0/24 is not used to reach any network. Each router creates its own SPF tree independently.

Table 3.2 Router A's simplified database

Device	Interface	Network	Neighbor	Metric
Directly connected	eth0.0	10.1.0.0/16	Directly attached	N/A
	eth0.1	192.168.1.0/24	Router B	1
	eth0.3	192.168.3.0/24	Router D	1
LSPs from Router B	eth0.1	192.168.1.0/24	Router A	1
	eth0.2	192.168.5.0/24	Router E	10
LSPs from Router D	eth0.1	192.168.3.0/24	Router A	1
	eth0.2	192.168.7.0/24	Router C	1
	eth0.3	192.168.9.0/24	Router E	1
	eth0.4	192.168.20.0/24	Router I	1

After the complete SPF tree is constructed, Router A creates its own routing table. The `show ip route` command displays the routing table. The output shows Router A's routing table with directly connected and dynamic routes. For example network 10.1.0.0/16 have a combination of both types. However, directly connected networks are prioritized. The first number in the brackets means the Administrative Distance for OSPF protocol and the second number describes the metric.

```
RouterA> show ip route
Codes: C - connected, S - static, O - OSPF, > - selected route,

O   10.1.0.0/16 [110/1] is directly connected, eth0.0, 00:30:16
C>* 10.1.0.0/16 is directly connected, eth0.0
O>* 10.5.0.0/16 [110/3] via 192.168.3.2, eth0.3, 00:29:25
O>* 10.6.0.0/16 [110/3] via 192.168.3.2, eth0.3, 00:29:21
O>* 10.7.0.0/16 [110/3] via 192.168.3.2, eth0.3, 00:29:21
O>* 10.16.0.0/16 [110/3] via 192.168.3.2, eth0.3, 00:29:25
C>* 127.0.0.0/8 is directly connected, lo
O   192.168.1.0/24 [110/1] is directly connected, eth0.1, 00:00:29
C>* 192.168.1.0/24 is directly connected, eth0.1
C>* 192.168.2.0/24 is directly connected, eth0.2
O   192.168.3.0/24 [110/1] is directly connected, eth0.3, 00:30:16
C>* 192.168.3.0/24 is directly connected, eth0.3
O>* 192.168.5.0/24 [110/11] via 192.168.1.2, eth0.1, 00:00:29
O>* 192.168.7.0/24 [110/2] via 192.168.3.2, eth0.3, 00:29:29
O>* 192.168.9.0/24 [110/2] via 192.168.3.2, eth0.3, 00:29:29
O>* 192.168.20.0/24 [110/2] via 192.168.3.2, eth0.3, 00:29:29
O>* 192.168.21.0/24 [110/3] via 192.168.3.2, eth0.3, 00:29:25
```

The next description follows a packet to be sent from User 1 to Server 1 in detail. When a packet travels through network from the source to the destination device, the IP addresses does not change. However, the physical address changes every hop in a frame. User 1 encapsulates the packet into Ethernet frame with the destination MAC address of Router A's interface eth0.0. Router A receives the Ethernet frame, checks the MAC address if it really corresponds to its eth0.0 and copy the frame into its buffer. The EtherType field in the header is 0x0800 which indicates that the encapsulated protocol in the payload of an Ethernet frame corresponds to Internet Protocol v4. Router A decapsulates the frame. But the destination IP address does not match any of the directly connected networks. To forward the packet, there has to be routes that represents the destination. Router A makes a forwarding decision for each packet that arrives at the interface and starts the routing process. According to its routing table, the router determine the best path to forward the packet. The next hop for network 10.7.0.0/16 is 192.168.3.2 on interface eth0.3. The IP packet is encapsulated in a new Ethernet frame with a new destination MAC of Router D's interface eth0.1. The packet arrives at Router D. Router D examines the MAC address if it corresponds to the interface eth0.1 and copy the packet to the buffer. The process is now similar as the process of Router A. Router D decapsulates the frame and searches the routing table for the destination IP address. Router D's routing table has a route to the 10.7.0.0/16 route, with a next-hop IP address of 192.168.9.2 and an exit interface of eth0.3. The packet is encapsulated into a new frame with a new destination MAC of Router E's interface eth0.2 and sent out the eth0.3 exit interface. The packet arrives at Router E. Router E again examines destination MAC address, decapsulates the frame and check the destination IP address. The searching of the routing table results in a network that is one of Router E's directly connected networks. It means that the packet can be sent directly to Server 1 and does not need to be sent to another router. The packet is encapsulated into a new frame and sent out Router E's interface eth0.0 to Server 1. The packet arrives at Server 1. Server 1 examines the destination MAC address of its Ethernet Network Interface Card (NIC). Server 1 decapsulates the frame and deliver the packet to operating system. Packet is forwarded from the originating source, through Router A, D and E to Server 1.

When users and servers generate some traffic, Fast Ethernet link between Router D and E, marked as red, become congested. Table 3.3 summarizes routing table records pointing to Server 1 from all the routers. In other words, the table highlights the shortest paths to the network 10.7.0.0/16. Router A calculates the shortest path to Router E passing Router D. The cost of an OSPF route is the accumulated value. Table 3.3 shows a cost of 3 because there are three Fast Ethernet links attached on the path. The calculation is similar for Routers B, C and I. The red link is still congested while the Ethernet links between Router B and E, with metric of 10, remain unused.

Table 3.3 Selected records from all routing tables pointing to the Server 1

Router	Interface	Target	Netmask	Gateway	Metric
Router A	eth0.3	10.7.0.0	255.255.0.0	192.168.3.2	3
Router B	eth0.1	10.7.0.0	255.255.0.0	192.168.1.1	4
Router C	eth0.2	10.7.0.0	255.255.0.0	192.168.7.2	3
Router D	eth0.3	10.7.0.0	255.255.0.0	192.168.9.2	2
Router E	eth0.0	10.7.0.0	255.255.0.0	Directly attached	N/A
Router I	eth0.1	10.7.0.0	255.255.0.0	192.168.21.1	2

It is important to determine if the packet is being forwarded as expected, if and why the packet is being sent elsewhere, or if the packet has been discarded. Well-known utilities like Ping or Traceroute are used for troubleshooting TCP/IP network connectivity. Below is an example of ping to test the ability of User 1 to communicate across mentioned network. When forwarding a packet between routers, the packet itself remains unchanged, with the exception of the Time To Live (TTL) field. Default TTL value for devices running Windows OS is 128. Each router decrements the TTL value by 1 every time it forwards the packet. In this case, the packet is three times forwarded (Router E, D and A) and the final TTL value equals to 125.

All pings are successful and the operation of network is verified. It means that the users' communication on the network and the operation of routers is working well. Traceroute or Tracert are utilities used to observe the path between hosts and generates a list of hops that are successfully reached along the path.

The outputs of the `ping` and `tracert` commands are mentioned below. The direction from Server 1 (10.7.0.100) to User1 (10.1.0.100) is as follows:

```
Pinging 10.1.0.100 with 32 bytes of data:
Reply from 10.1.0.100: bytes=32 time=2ms TTL=125

Tracing route to 10.1.0.100
  1    <1 ms    <1 ms    <1 ms  10.7.0.1
  2     3 ms    <1 ms    <1 ms  192.168.9.1
  3     1 ms     1 ms     1 ms  192.168.3.1
  4     1 ms     1 ms     1 ms  10.1.0.100
Trace complete.
```

The opposite direction from User1 (10.1.0.100) server (10.7.0.100):

```
Pinging 10.7.0.100 with 32 bytes of data:
Reply from 10.7.0.100: bytes=32 time=2ms TTL=125

Tracing route to 10.7.0.100
  1    <1 ms    <1 ms    <1 ms  10.1.0.1
  2     1 ms    <1 ms    <1 ms  192.168.3.2
  3     1 ms     1 ms     1 ms  192.168.9.2
  4     1 ms     1 ms     1 ms  10.7.0.100
Trace complete.
```

When the metric between Router D and Router E is set to 10, the situation changed as expected. The direction from Server 1 (10.7.0.100) to User1 (10.1.0.100) is as follows:

```
Pinging 10.1.0.100 with 32 bytes of data:
Reply from 10.1.0.100: bytes=32 time=2ms TTL=124

Tracing route to 10.1.0.100 over a maximum of 30 hops

  1    <1 ms    <1 ms    <1 ms  10.7.0.1
  2     1 ms    <1 ms    <1 ms  192.168.21.2
  3     4 ms     1 ms     1 ms  192.168.20.1
  4     4 ms     1 ms     1 ms  192.168.3.1
  5     4 ms     1 ms     1 ms  10.1.0.100
```

The opposite direction from User1 (10.1.0.100) Server 1 (10.7.0.100):

```
Pinging 10.7.0.100 with 32 bytes of data:
Reply from 10.7.0.100: bytes=32 time=2ms TTL=124

Tracing route to 10.7.0.100
  1    <1 ms    <1 ms    <1 ms  10.1.0.1
  2     1 ms    <1 ms    <1 ms  192.168.3.2
  3     1 ms     2 ms     3 ms  192.168.20.2
  4     1 ms     3 ms     4 ms  192.168.21.1
  5     1 ms     3 ms     4 ms  10.7.0.100
Trace complete.
```

There are several ways in which to verify proper OSPF configuration and operation on a router. Show commands such as `show ip ospf`, `show ip ospf interface` and `show ip ospf neighbor` are mostly used in this thesis. The first mentioned command is used to display OSPF information for one or all OSPF processes running on the router. Below is an output from Router A:

```
RouterE> show ip ospf
 OSPF Routing Process, Router ID: 0.0.0.1
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
 RFC1583Compatibility flag is disabled
 SPF schedule delay 1 secs, Hold time between two SPFs 1 secs
 Refresh timer 10 secs
 Number of external LSA 0. Checksum Sum 0x00000000
 Number of areas attached to this router: 1

 Area ID: 0.0.0.0 (Backbone)
   Number of interfaces in this area: Total: 3, Active: 3
   Number of fully adjacent neighbors in this area: 2
   Area has no authentication
   SPF algorithm executed 6 times
   Number of LSA 13
   Number of router LSA 6. Checksum Sum 0x0002ddaf
   Number of network LSA 7. Checksum Sum 0x000402e4
   Number of summary LSA 0. Checksum Sum 0x00000000
   Number of ASBR summary LSA 0. Checksum Sum 0x00000000
   Number of NSSA LSA 0. Checksum Sum 0x00000000
```

The next section presents a series of fundamental graphs related mainly to the utilization of the devices. Measured data are captured with the help of PRTG monitoring tool. The goal is to gradually increase the load of traffic on servers. In this scenario, each five minutes new amount of traffic is generated on the users and sent to the servers. The observation interval is set to 30 minutes.

The behavior of default OSPF routing process is shown in Figure 3.7. There are three interfaces connecting different routers - interface eth0.1 connecting Router B, interface eth0.2 connecting Router D and interface eth0.3 connecting Router I. Since the shortest path from User 1, User 2 to Server 1, Server 2 passes through Router D (eth0.3) and Router E (eth0.2), this link may become highly utilized.

From the graph can be seen that there is no traffic on interface eth0.1 and eth0.3 of Router E while interface eth0.2 is utilized. Three interfaces in the chart correspond to three curves. The blue curve shows the total traffic utilization of interface eth0.1, the orange curve deals with interface eth0.3. The red curve shows the total traffic utilization of interface eth0.2.

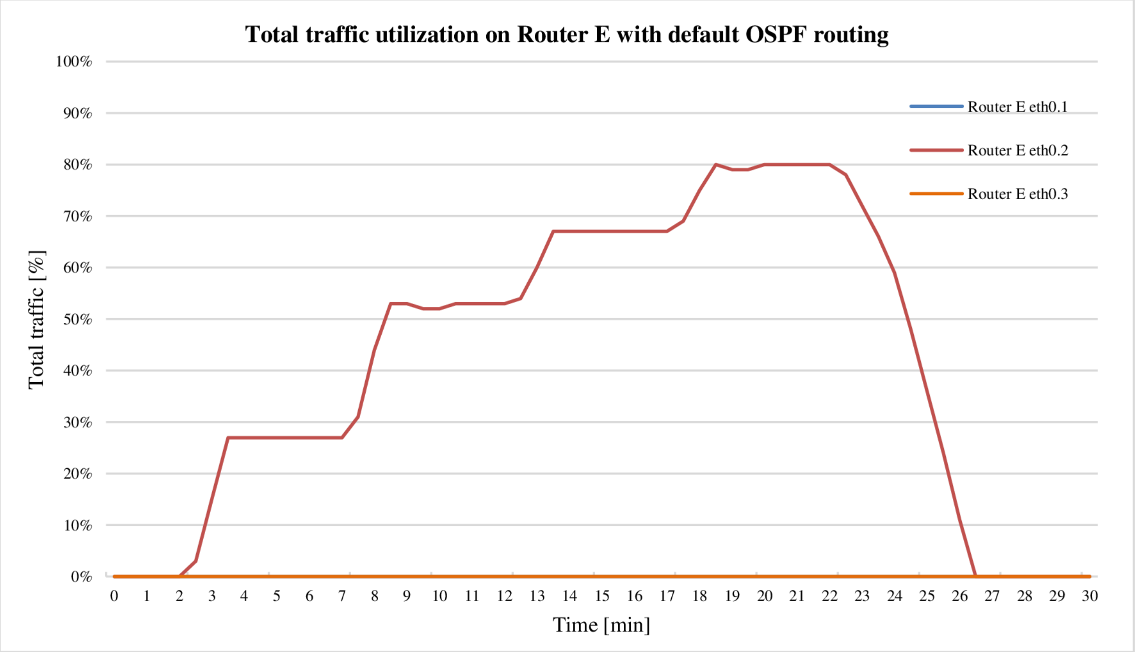


Figure 3.7 Total traffic utilization on Router E with default OSPF routing

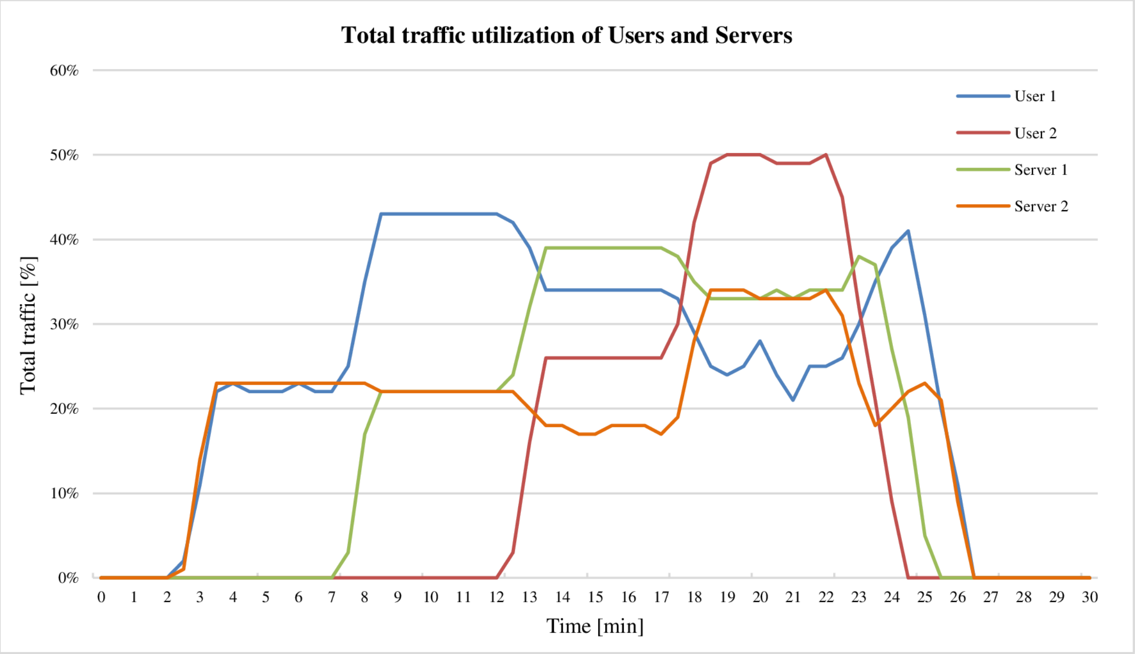


Figure 3.8 Total traffic utilization of Users and Servers

During the second minute, User 1 starts to send data to Server 2 and the red graph increases. The next five minutes is the total traffic utilization of Router E's interface eth0.2 stable about 27% because no other traffic passes. During the seventh minute, User 1 starts to send data to Server 1 and the traffic utilization rises to 54%. Users' and servers' behavior can be observed in Figure 3.8. The characteristics follow described process. After another five minutes (minute 12), User 2 starts to send packets to Server 1 and the utilization rises to value of 67%. Finally, the last connection between User 2 and Server 2 is initialized in 17th minute. At this moment, all the traffic from users to servers passes through the interface eth0.2 which is highly utilized. On the other hand, the bandwidth of remaining two links, which can carry the traffic but are not allowed to, is wasted. To get a complete view of the situation, Figure 3.9 shows the utilization on Router D interface eth0.3 and Router E interface eth0.2.

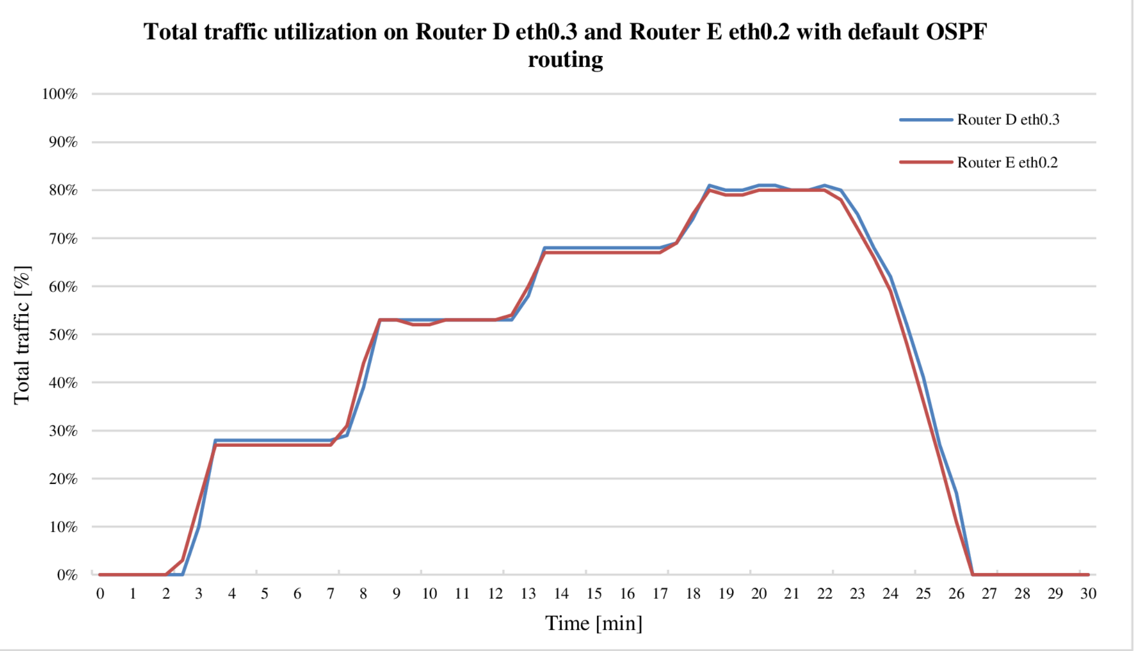


Figure 3.9 Total traffic utilization (Router D eth0.3, Router E eth0.2) with OSPF routing

Packets are not forwarded through Router B and I at all. According to default OSPF routing algorithm, Router B and Router I are not used for packet forwarding.

4 PROPOSAL OF A NEW ROUTING METHOD

The efficiency of packet switched network communication depends on the ability of routers to determine the best path to send and forward packets to desired destination. There are some methods extending OSPF with the purpose of achieving more effective solution to the routing optimization problem. These methods will be shortly described below.

Determining the best path for a packet involves evaluation of multiple paths (if available) to the same destination and selection of the most suitable one to reach that destination, possibly on the per flow basis. OSPF provides support for mentioned multiple paths to the desired destination. The advantage of multi-path algorithm is that it can provide better throughput and redundancy. Preferred load balancing algorithms use bandwidth information to distribute the traffic over more of router's ports. There are two types of load balancing: per-flow and per-packet. Per-flow load balancing distributes the packets according to the addresses while per-packet load balancing generally uses round-robin technique.

When two or more routes to the same destination have identical metric value, a router load balances between these paths. The procedure is called equal cost load balancing. This technique splits the traffic to multipath destination quasi-equally between all the equal metric paths. If a router discovers multiple paths to a destination, the routing table is updated and contains one destination network but multiple exit interfaces, one for each equal cost path. An example can be shown from the Router E's routing table, when focused on a network 192.168.20.0/24. There are two equal routes to this network with the cost of 2, using exit interfaces eth0.2 and eth0.3.

```
RouterE> show ip route
O>* 192.168.20.0/24 [110/2] via 192.168.9.1, eth0.2, 00:09:25
  *                         via 192.168.21.2, eth0.3, 00:09:25
```

If there are different routing metric values for different paths, some routing protocols are able to use unequal cost load balancing. As mentioned in Chapters 1.2.4, 1.2.5, there are two Cisco proprietary protocols that support unequal cost load balancing: IGRP and its enhanced version EIGRP. The protocols use a composite metric which is associated with the following criteria: bandwidth, delay, reliability, load and Maximum Transmission Unit (MTU). By default, only total path delay and bandwidth are considered. Unequal cost load balancing is not further considered for the purpose of this thesis.

Equal cost multipath load balancing is a significant improvement over a single path routing. On the other hand, it is not the final solution to the traffic optimization problem. This is where improved techniques are needed. Method proposed builds on the basis of OSPF protocol mentioned in detail in chapter 1.2.2. The algorithm determines the path of minimal total cost between the nodes based on the bandwidth parameter. However, the default OSPF routing behavior does not consider the current link load as described in chapter 3. The proposed path determination is based on the utilization of the individual links at that point in time. This method offers an approach to more efficient path usage as the load requirements grow. The purpose is to relieve the congested links and split the traffic between multiple paths supporting load balancing, if there are any. The aim is to set link metrics to keep the traffic load on all links at acceptable levels. A new routing mechanism called DM-SPF (Dynamic Metric - Shortest Path First) is introduced. Initial thoughts related to the method are described in [56], [57] and [58].

4.1 DM-SPF (Dynamic Metric – Shortest Path First)

Proposed routing protocol is a set of processes, algorithms and messages used for the deployment of dynamic metric in OSPF networks. The idea is to create a solution to traffic optimization problem to prevent network performance degradations and congestions. The purpose of this method is to gradually adapt the routing to the changing conditions of a network. The method basics are summarized as follows:

- Routers process additional information about the state of links related to current load, over general OSPF.
- Overloaded links initialize rebalancing of the traffic between multiple paths, if there are any such paths.
- New routing tables with new quasi-shortest paths are computed based on the load information.
- The traffic is sent through newly calculated paths, with the intent to reduce the load of overloaded links.

Sending packets through a single path is not the most efficient use of available bandwidth. Instead of passing the traffic to the overloaded link, the routers according to DM-SPF will take into account the link load and try to find less loaded concurrent paths for load balancing. Paths with equal metrics to the same destination may share the load in a per-flow fashion. To ensure that the packets are routed via the best possible paths, an additional information describing the load of various links combined into the metric, is attached. An integer value is assigned for the load of a link as an additive component to the bandwidth based metric. The load statistics are computed using the interface counters generally available for SNMP purposes.

The purpose of DM-SPF distributing the traffic among multiple paths to the same destination is to use available bandwidth more efficiently. The goal is to reduce the amount of transferred packets when links are under a significant traffic load and to avoid any critical bottlenecks.

4.1.1 DM-SPF routing protocol principles

The DM-SPF principle is described in Figure 4.1. In the figure, each path is labeled with a value of its cost. If a user communicates with a server, the entire data traffic passes the Link A, marked red. The metric of the shortest path is 1. There is no traffic going through Link B and C. Under moderate traffic, the utilization is acceptable and no changes are needed. Under heavy traffic, more than 90%, the network behavior significantly changes from routing point of view. According to DM-SPF, overloaded router initializes rebalancing of the traffic between multiple paths. DM-SPF runs the process of finding next suitable path with the lowest metric. The possible path to pass the traffic is Link B and C, so far used as standby links, with the composite metric of 2. It can be seen that the path is not the path with the minimum number of hops. To ensure that the traffic will be balanced between two paths, the cost of overloaded Link A must be increased by 1. After the change, two paths from the user to the server have the same metric of 2. That forces the routers to insert multiple paths to the destination into their routing tables. It means that DM-SPF manipulates the routing process in order to achieve load-balancing. If there are no other suitable paths for load balancing to be found, the routing situation remains and no changes are made. If the network traffic becomes low, less than 10%, the affected router is triggered to use standard OSPF path metric calculation.

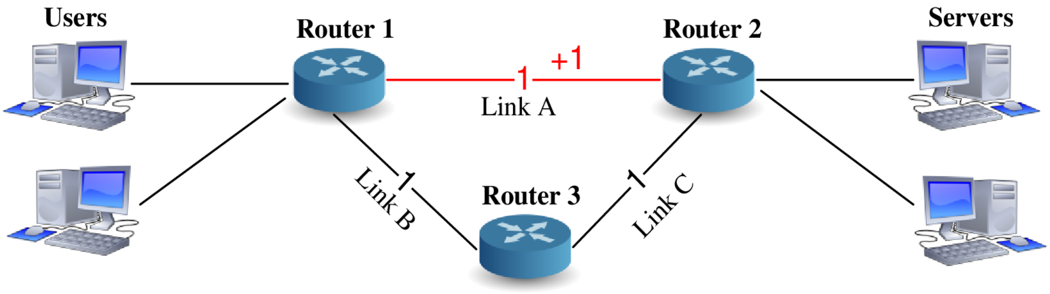


Figure 4.1 Simple DM-SPF explanatory network

After the change, all routers must agree on the network topology and achieve routing convergence. Having explained the introduction how DM-SPF works, what remains is to describe the new metric calculation.

4.1.2 A novel metric calculation

The OSPF RFC 2328 [11] does not specify what the link metric should be. The assignment of link metrics is left up to network administrators. Most of the routers (Cisco routers e.g.) calculate the metric from link bandwidth. The metric is then inversely proportional to the bandwidth. A higher bandwidth results in lower metric. The formula used to calculate the metric of one interface is:

$$OSPF\ metric = \frac{reference\ bandwidth}{bandwidth}. \tag{4.1}$$

OSPF can discriminate between all the possible paths to the same destination and try to use the best one. The metric of any given route is calculated by summing the metrics of all links encountered along that route. Different manufacturers use different reference bandwidth, however, common reference bandwidth is 100 Mbps. The reference bandwidth has to be equal for all the routers in the same domain. Only one metric can be assigned per interface. Often used OSPF metrics, according to formula (4.1), are summarized in Table 1.4. The interface metric is an unsigned 16-bit integer so it can be valued in the range from 1 to 65,535. Metrics are rounded down to the nearest integer.

If the technology is faster than 100 Mbps, the default metric will be always 1 which is not optimal and may cause some inappropriate routing decisions. To solve this problem, it is possible to set new reference bandwidth on all routers in an area or even better in the entire network. The reference bandwidth can be modified using a router configuration command `auto-cost reference-bandwidth`. When altered by network administrator, it has to be manually set the same on all routers in order to shortest path selection to work properly. Routers can calculate metrics using largest value already known to them, but also have to transmit used reference bandwidth, so other routers can correct their values if used reference values are mismatched. Anyway, maintaining the routing table by manual configuration is not always feasible.

Link metrics do not include the link load. The idea is to consider link load as an additive parameter. Solution is that an integer value representing the load of a link is used as an additive component of bandwidth based metric. Together with default cost, the additive metric value reflects the current load of the link so that the routing table contains accurate optimal path information. The novel metric is then based on multiple characteristics of a path - bandwidth and link load. The path selection is then influenced by preferring the path with the highest bandwidth while considering the traffic utilization of a certain link. It means that the novel method bases route selection on multiple submetrics, combining them into a single metric. The smaller is the value of the metric, the more preferable is the path.

If load is to be used as part of a composite metric, DM-SPF must not allow sudden changes in link traffic in order to destabilize the network. The goal is to avoid the impact of unexpected and accidental peak utilization. Such a burst of heavy traffic could trigger an update.

Moving averages

Since the link load is a dynamic variable, moving average is used for the metric calculation. A set of observations of a variable arranged in time from the past to the presence form a time series. Data collected in time order can be averaged over several time periods. Moving averages are often used in analyzing time series data. Moving averages might be useful for measuring the changes in a trend, smoothing fluctuations and as a forecast for the next period. For this thesis, the used raw data are link load observations of routers. Link load values represent data from which the moving average will be computed.

Simple Moving Average (SMA) is a method of computing the average of a number of the most recent n data values in a series. Each observation in the calculation receives the same weight. The formula to calculate SMA is as follows [41]:

$$SMA_t = \frac{c_{t-(n-1)} + c_{t-(n-2)} + \cdots + c_{t-2} + c_{t-1} + c_t}{n} \tag{4.2}$$

where SMA_t is the simple moving average at the end of period t, parameter c expresses an actual observation in time period t and n is the number of periods included in each average. For successive values, when a new value enters, the oldest one is dropped. The formula can be simplified as:

$$SMA_t = \frac{1}{n}\sum_{i=0}^{n-1} c_{t-i}. \tag{4.3}$$

In the computation of the SMA, equal weights were assigned to all periods. Weighted Moving Average (WMA) involves selecting extra weights for different data values. It means that WMA assigns an extra weight to some demand values. The formula to compute WMA with assigned weights is as follows [41]:

$$WMA = \left(\sum_{i=1}^{n} w_i c_i\right)\left(\sum_{i=1}^{n} w_i\right)^{-1} \tag{4.4}$$

where w_i is the assigned weight and c is an current observation in time period t.

Another possibility is to assign the weights deterministically. It means that older observations receive gradually lower weights and the most recent observation receives the maximum weight. The latest observation has weight n, the second latest observation has weight $n - 1$, etc. In this case, WMA increases the importance of the most recent data. Expression for WMA, where weights decrease arithmetically is:

$$WMA_t = \frac{nc_t + (n-1)c_{t-1} + \cdots + 2c_{t-n+2} + c_{t-n+1}}{n + (n-1) + \cdots + 2 + 1}. \tag{4.5}$$

The previous formula (4.5) of WMA computation with arithmetically decreased weights can be expressed as:

$$WMA_t = \frac{2}{n(n+1)}\sum_{i=0}^{n-1}(n-i)c_{t-i}. \tag{4.6}$$

Disadvantages such as small process change detection, and the fact that mentioned moving averages include data for only the number of periods the moving average covers, lead to the use of Exponentially Weighted Moving Average (EWMA) [42]. EWMA, sometimes also called Exponential Moving Average (EMA), solves the shortcomings of SMA. EWMA uses weight factors that decrease exponentially. It means that the weight of each older observation decreases exponentially which brings more importance to the recent observation.

Let c_t be an observation at time *t*, the explicit formulation of EWMA is:

$$EWMA_t = \lambda c_t + \lambda(1-\lambda)c_{t-1} + \lambda(1-\lambda)^2 c_{t-2} + \cdots + \lambda(1-\lambda)^{t-1}c_1 + (1-\lambda)^t c_0$$

$$= \lambda \sum_{i=0}^{t-1} (1-\lambda)^i \, c_{t-i} + (1-\lambda)^t c_0 \qquad (4.7)$$

where $EWMA_t$ represents the exponentially weighted moving average of all past observations with the weights decreasing exponentially. The starting value c_0 equals 0 or is generally being set to the mean of former observations. The effect of the starting constant c_0 decreases as time increases. The formula relies on an effective period for the exponential moving average called weight or smoothing factor λ where $\lambda = \frac{2}{n+1}$ and n is the period of the moving average. The parameter λ determines the rate at which older data enter into the calculation of the EWMA and has dynamic range of $0 < \lambda \leq 1$. The value of $\lambda = 1$ implies that only the most recent observation influences the EWMA. Thus, a large value of λ gives more weight to recent data. The weight of all the older observations is then decreased by the factor (1- λ) [42], [43].

Exponentially Weighted Moving Average can be computed recursively as [42]:

$$EWMA_t = \lambda c_t + (1-\lambda)EWMA_{t-1}. \qquad (4.8)$$

The recursive form of the EWMA calculation simplifies the formula (4.7) and decreases processing and memory demands on computing devices which is important for implementation into real devices with little spare resources. The recursive fashion (4.8) requires only two pieces of information to be processed. The expression can be explained as follows: at time t, the EWMA equals the multiple of lambda times actual observation plus 1 less lambda times previous observation. The predictive form of EWMA is then given by:

$$EWMA_{t+1} = \lambda c_t + (1-\lambda)EWMA_t. \qquad (4.9)$$

EWMA has become popular process-monitoring tool in a process-control field. Due to EWMA's robustness and ability to monitor a dynamic process with memory and drift, this approach is adopted within this thesis.

The application of the EWMA technique is demonstrated using an example depicted in Figure 4.2. The figure shows measured network utilization and the impact of EWMA on the data. The blue line stands for raw link load data, the representative of current load utilization. The red line represents the EWMA applied to the raw data with 5, 30, 60 and 300 seconds time period. It can be seen that the series are smoothed, with much lower variance. The decrease of weights is an exponential function of the weighting factor λ. Such process is demanded to avoid undesired reaction to unexpected jumps in link load.

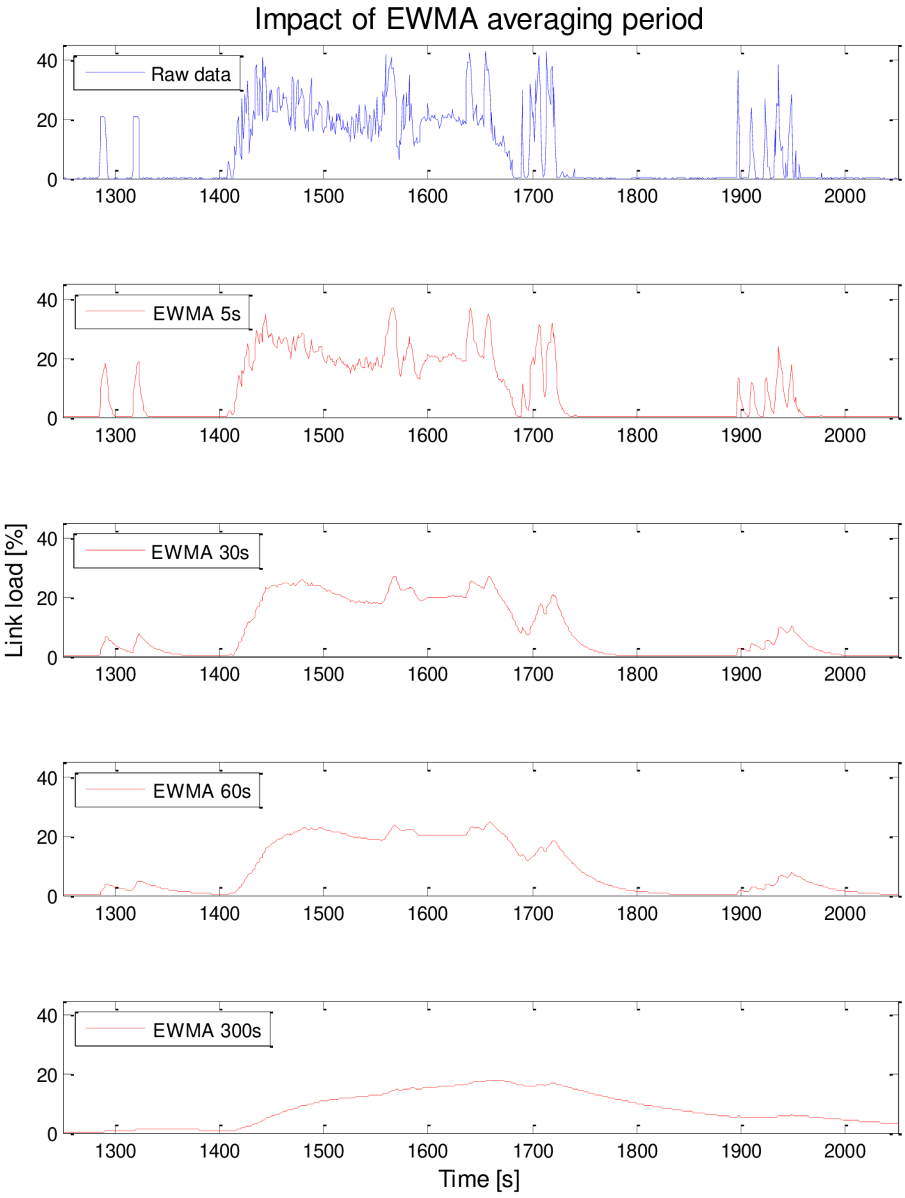


Figure 4.2 Impact of EWMA averaging period

The next step is to find the most suitable smoothing factor. If the value of λ is high, like in the case EWMA five seconds, only few recent observations are considered and the EWMA curve almost copies the load curve. So the parameter λ should be decreased. On the other hand, if the value of λ is close to zero, like in the case EWMA 300 seconds, the moving average is significantly influenced by older values. It can be seen that the convergence time is too long, which is not desirable. Based on these findings, the most suitable period is then 60 seconds for DM-SPF. More recent values are weighted more prominently than older values and burst traffic is filtered. The resulting graph with chosen time period is shown in Figure 4.3 in detail.

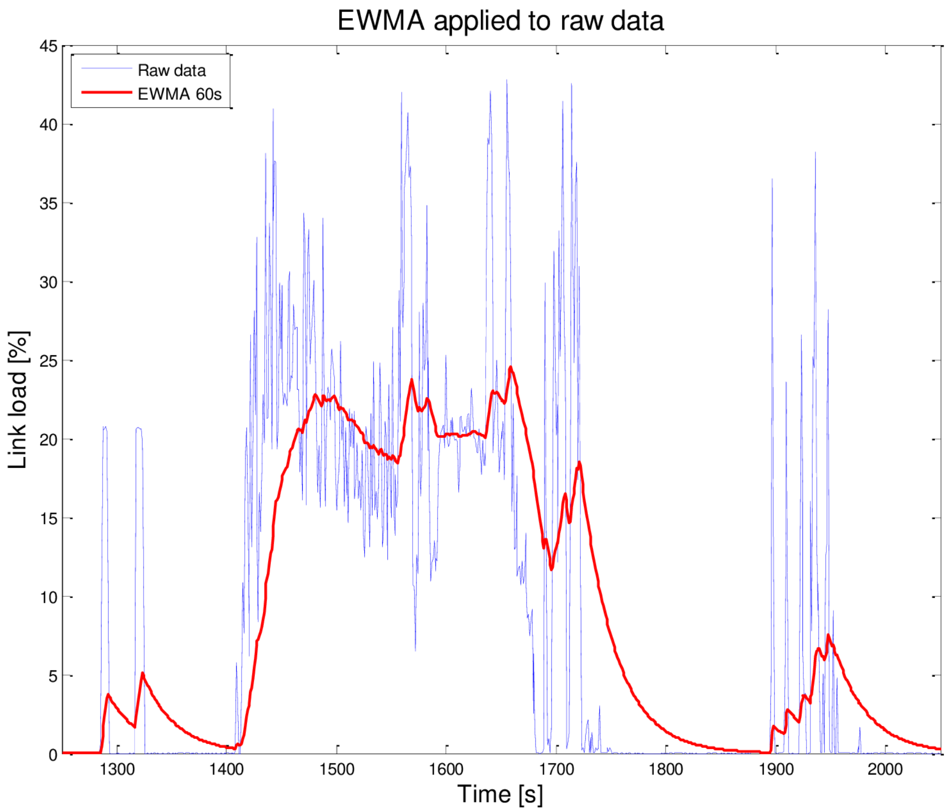


Figure 4.3 EWMA applied to raw data

The computation of the metric for DM-SPF builds on common metric calculations used by OSPF protocol mentioned in formula (4.1). Formula (4.8) is used for EWMA computation and applied onto the metric. The resulting metric including EWMA can be then expressed as:

$$DMSPF\_metric_{EWMA}(load) = \begin{cases} metric\_init & for\ EWMA(load) < 10\% \\ metric & for\ 10\% < EWMA(load) < 90\% \\ metric\_load\_balance & for\ EWMA(load) > 90\% \end{cases} \quad (4.10)$$

where the component $metric\_init$ initialize default OSPF protocol metric if the EWMA load of the affected router drops below 10%. The component $metric$ is a link metric computed according to the formula (4.1) and $metric\_load\_balance$ is triggered when 90% of utilization is reached.

The question is how to find alternative paths. The process of finding alternative paths includes following steps. The current shortest path (Link A) is not considered for further computations and is marked down. The next step is to apply Dijkstra’s algorithm to this modified topology. After new shortest paths with new composite metrics are found, it is possible to increase original metric of Link A to demanded value.

Let's assume in the following graph in Figure 4.4 that $q(a) = 1, q(b) = 1, q(c) = 1$ where $q()$ is a function converting the edge parameter to its metric calculated according to general OSPF process.

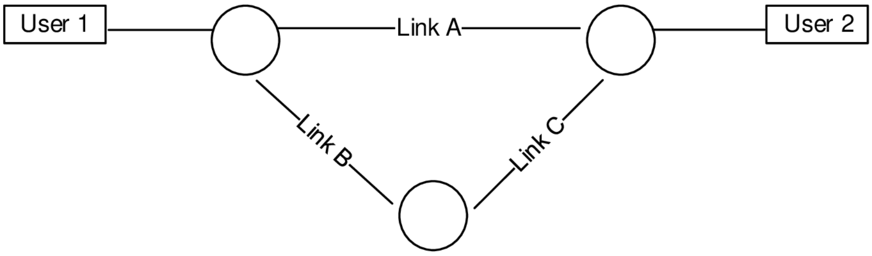


Figure 4.4 Explanatory graph (alternative path finding)

Dijkstra's algorithm is performed and chain α, which represents the shortest path to the destination, and its composite path metric q are obtained. In this case, $q = 1$ for the chain α, consisting of Link A only.

All cuts in the graph including components of the chain α have to be obtained. There are two cuts for the graph. Then for each component of the chain α, subgraphs are created. Each subgraph β_i includes the component α_i and all the edges that are part of any cut together with the component α_i and have the same metric, in case that there exist cuts consisting of not only α_i. If for any α_i there are no cuts also including other links, this component is removed from consideration. Each of the subgraphs represents possible optimization point. The next steps are performed for each subgraph. Edge from chain α has to be removed from β_i what yields δ_i. Then again Dijkstra's algorithm is ran on δ_i to get second-best alternative subpath, cost m of which will be later used as a $metric\_load\_balance$ for the original link α_i, except for the case that β_i is not a cycle. If the alternative subpath cost is equal to $q(\alpha_i)$, equal cost multipath was already used, therefore no further optimization should be done for this point and it is removed from list of possible optimization points γ. Especially, if β_i is not a cycle, $metric\_load\_balance$ equals m minus metrics of all edges from α that have to be added to β_i for it to be a cycle.

There are two simulations depicted in Figure 4.5 and Figure 4.6. The first graph shows the behavior of DM-SPF when the alternative links (Link B and C) are not used in the beginning. The blue line stands for total measured data transmitted over both original and alternative links together (Link A). This is a situation representing default OSPF behavior. The next step is to apply EWMA, displayed in red curve. When 90% of utilization is reached, the traffic is load balanced. The black line displays resulting load on Link A. The utilization was significantly decreased what is the primary goal of DM-SPF. A part of traffic is switched to the alternative path which is displayed in green line. When 10% of utilization is reached, the alternative link is not needed any more and the original link takes the load back.

Figure 4.6 describes the scenario when DM-SPF was already used while monitoring the link utilization. It can be seen that the alternative link is in use on start.

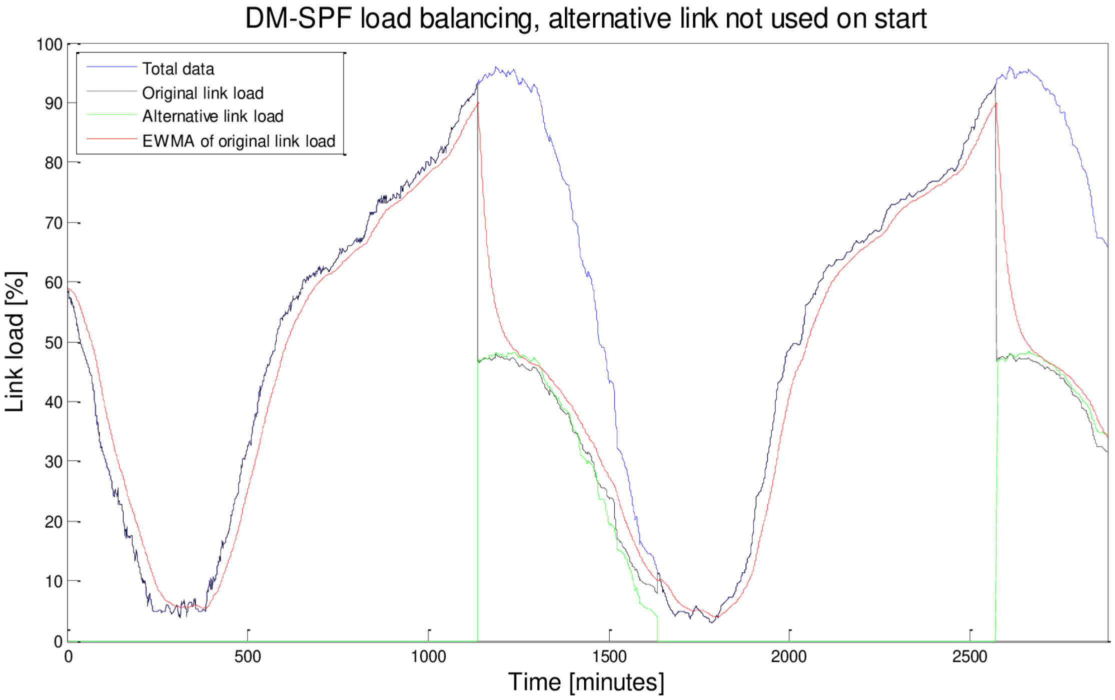


Figure 4.5 DM-SPF load balancing when alternative link is not used on start

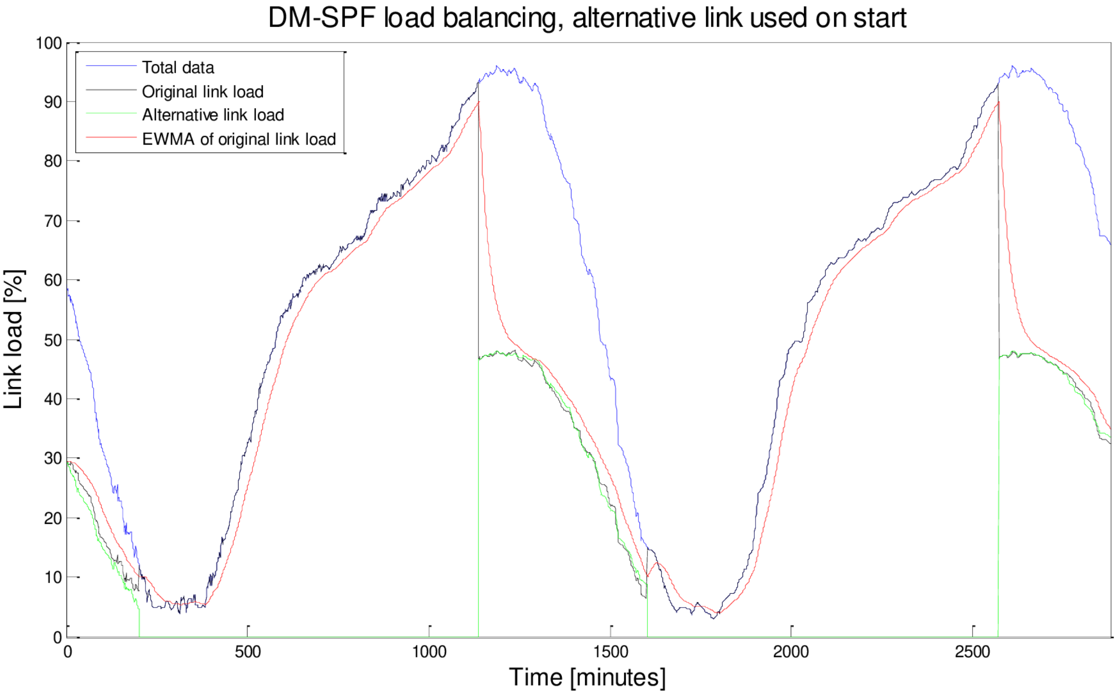


Figure 4.6 DM-SPF load balancing when alternative link is used on start

4.2 The behavior of a router running algorithm proposed

DM-SPF describes functionality in a router that distributes packets based on improved routing information. This implementation benefit does not require any special configurations in other devices. Routers learn about the possibility of load balancing through the novel mechanism and build its routing tables dynamically and accordingly. If possible, DM-SPF leads to the use of multiple parallel links without additional hardware multiplexers. This chapter will describe the process of DM-SPF routing mechanism from the routers' point of view. All routers in the topology will complete following process to reach the state of convergence:

- Router discovers its own directly connected networks.
- Router must exchange Hello packets with directly connected networks.
- Router must create Link State Packet containing the information about directly connected networks and send LSP to neighbors.
- Routers use LSPs to create a database from which the network topology and the shortest paths are computed.
- Routers use DM-SPF in order to achieve load balancing if higher load utilization threshold is reached.
- Routers use DM-SPF in order to find shortest path if lower load utilization threshold is reached.

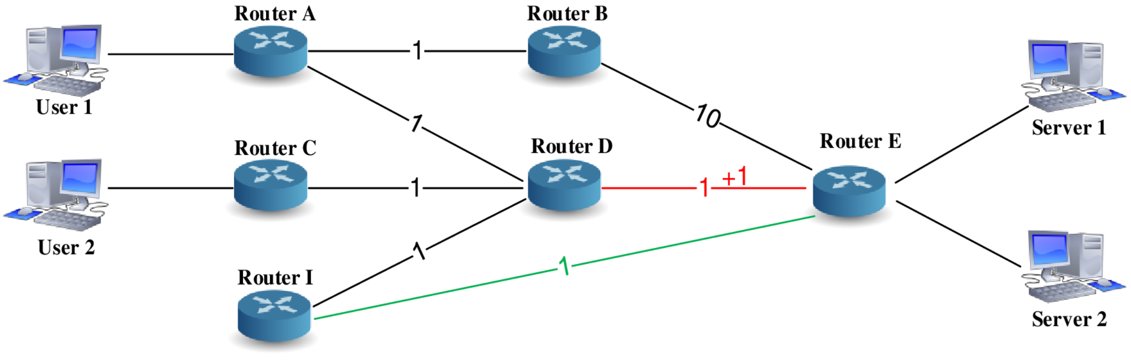


Figure 4.7 Principle of DM-SPF on testbed

Let's assume that the network described in Figure 4.7 is under moderate utilization. The topology corresponds to the network in Figure 3.2. The routing tables are shown in Table A.1. DM-SPF is implemented and controls the routing process according to equation (4.10). The $metric$ function is used under normal conditions. If the utilization situation changes and the EWMA of the link load reaches 90%, $metric\_load\_balance$ function is called. As mentioned before, the averaging period for EWMA is 60 seconds. The entire data traffic passes the link between Router D and Router E, marked as red link. Affected Router D detects the imminent danger. Among others, information about the amount of traffic is included in the interface status:

```
RouterD> show interface eth0.3
Interface eth0.3 is up, line protocol detection is disabled
index 6 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
HWaddr: 58:6d:8f:b9:6c:a2
inet 192.168.9.1/24 broadcast 192.168.9.255
input packets 2027, bytes 155746, dropped 0, multicast packets 324
input errors 0, length 0, overrun 0, CRC 0, frame 0, missed 0
output packets 1755, bytes 187086, dropped 0
output errors 0, aborted 0, carrier 0, heartbeat 0, window 0
collisions 0
```

The metric of the shortest path for Router A and C to send packets to Router E is 2. There is no traffic on links from Router B to Router E and from Router I to Router E. DM-SPF triggers the process of finding alternative paths. To do so, LSDB is used due to collected LSPs. From the database, alternative path can be found. Each record in the database contains information like an interface identifier, a link number, and metric. The router has a complete map of all the destinations in the topology and the routes to reach them. As described in Chapter 1.2.2, each router has a router ID. This identifier is used by LSDB as a method of tracking routers and their links.

Let's take a look at the current LSDB for Router D. The link state information for each router is summarized in Table 4.1.

Table 4.1 Router D's Link State Database before change

Device	Interface	Network	Neighbor	Metric
Directly connected	eth0.1	192.168.3.0/24	Router A	1
	eth0.2	192.168.7.0/24	Router C	1
	eth0.3	192.168.9.0/24	Router E	1
	eth0.4	192.168.20.0/24	Router I	1
LSPs from Router A	eth0.0	10.1.0.0/16	Directly attached	N/A
	eth0.1	192.168.1.0/24	Router B	1
	eth0.3	192.168.3.0/24	Router D	1
LSPs from Router B	eth0.1	192.168.1.0/24	Router A	1
	eth0.2	192.168.5.0/24	Router E	10
LSPs from Router C	eth0.0	10.5.0.0/16	Directly attached	N/A
	eth0.2	192.168.7.0/24	Router D	1
LSPs from Router E	eth0.0	10.7.0.0/16	Directly attached	N/A
	eth0.1	192.168.5.0/24	Router B	10
	eth0.2	192.168.9.0/24	Router D	1
	eth0.3	192.168.21.0/24	Router I	1
	eth0.4	10.6.0.0/16	Directly attached	N/A
LSPs from Router I	eth0.0	10.16.0.0/16	Directly attached	N/A
	eth0.1	192.168.20.0/24	Router D	1
	eth0.2	192.168.21.0/24	Router E	1

An example below shows records of LSDB by `sh ip ospf database` command from the Router D.

```
RouterD> sh ip ospf database
       OSPF Router with ID (0.0.0.4)
                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
```

```
0.0.0.1         0.0.0.1         1464 0x80000005 0xe884 3
0.0.0.2         0.0.0.2         1456 0x80000006 0xef88 2
0.0.0.3         0.0.0.3         1461 0x80000004 0xad98 2
0.0.0.4         0.0.0.4         1458 0x80000005 0x6609 4
0.0.0.5         0.0.0.5         1459 0x80000004 0x8788 5
0.0.0.6         0.0.0.6         1456 0x80000005 0xaf58 3
                Net Link States (Area 0.0.0.0)
Link ID         ADV Router      Age  Seq#       CkSum
192.168.1.2     0.0.0.2         1463 0x80000002 0xc81f
192.168.3.2     0.0.0.4         1469 0x80000002 0xc21f
192.168.5.2     0.0.0.5         1463 0x80000002 0xbe1e
192.168.7.2     0.0.0.4         1464 0x80000002 0xaa31
192.168.9.2     0.0.0.5         1463 0x80000002 0xa630
192.168.20.2    0.0.0.6         1466 0x80000002 0x3594
192.168.21.2    0.0.0.6         1461 0x80000002 0x3493
```

It was found that the possible link to pass the traffic is between Router I and Router E, marked as green link, so far used as a standby link. The composite metric of the whole path (Router A, D, I and Router E) equals to 3. To achieve load balancing, the metric value of the overloaded link must be increased by 1.

The process of finding alternate paths is described as follows. Dijkstra's algorithm is applied and chain α representing link between Router A and D and link between Router D and E with composite metric of 2 is obtained. Since link between Router A and D has no alternate path, it is not considered any more. Next step is to find cuts that split all paths from User 1 to Server 1 and include α_i as one, but not only one, of the components. There are two such cuts as described in Figure 4.8, marked red and green. Next, subgraph β_i is created which includes all edges that mentioned cuts consist of. Therefore, in this example $\beta_2 = (\text{Link D} - \text{E}, \text{Link D} - \text{I}, \text{Link I} - \text{E})$. It is worthy to mention that subgraph β_2 is a cycle. Subgraph δ_2 is created from β_2 by removing α_2, $\delta_2 = (\text{Link D} - \text{I}, \text{Link I} - \text{E})$. There is only one point of optimization because there are no other alternative paths for Link A-D. Dijkstra's algorithm is applied to the new subgraph γ_2 to get second-best alternative subpath. Cost $m = 2$ will be later used as $metric\_load\_balance$ for the original link, $metric\_load\_balance = 2$.

Before any change is done, all the routers from subgraph γ_2 has to be notified of this update. This can be done by sending DM-SPF LSA message. If the routers in question (Router I) subsequently detect link overload, they respond to this message what leads to reverting the metric to the original state $metric\_init$ even during high load.

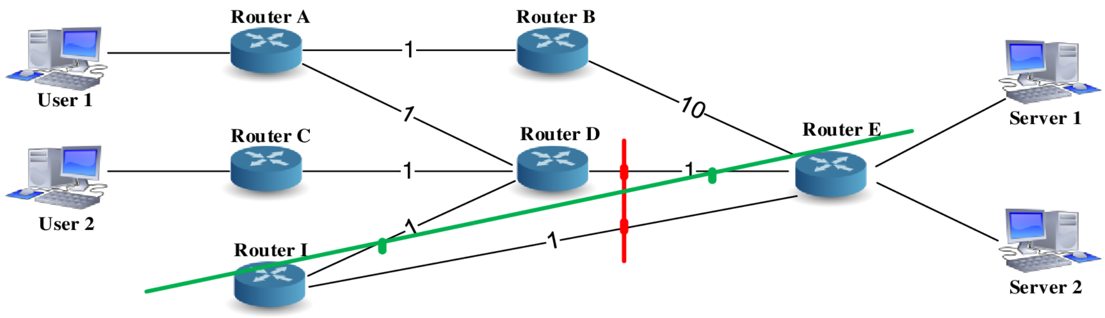


Figure 4.8 Performed cuts of the graph

DM-SPF now uses two paths to get to Router E because both paths have the same metric of 3. The path on Router A, B and E is still not in use because of high composite metric eleven.

LSPs are sent out all interfaces on Router D. The LSP flooding process causes the transmission of the messages on all interfaces except the incoming one. Neighbor routers receive the LSPs and forward the LSP packet out all interfaces excluding the interface where this LSP was received. As explained in Chapter 1.2.2, each message has a time stamp or a message number which distinguishes old and new information and prevent database pollution. Immediate flooding achieves fast convergence. If updating is not done in a timely fashion, the routing information may be incomplete or inaccurate, resulting in packet delays and possible packet loss.

The way of managing the number of adjacencies and reducing the amount of OSPF traffic in multi-access networks is to elect a Designated Router (DR) and a Backup Designated Router (BDR). The DR represents the collection and distribution point for LSPs and is responsible for updating all other routers. The BDR substitutes DR if the DR fails. All other routers are called DROthers. Routers form full adjacencies with DR and BDR send LSP packets only to DR (BDR) with multicast address 224.0.0.6. The DR is then responsible for forwarding LSPs to DROthers. DR uses multicast address 224.0.0.5. The resulting advantage is that one router performs all the LSP flooding.

After new LSP packets are sent, new LSDB (Table 4.2) with modified metrics is created.

Table 4.2 Router D's Link State Database after change

Device	Interface	Network	Neighbor	Metric
Directly connected	eth0.1	192.168.3.0/24	Router A	1
	eth0.2	192.168.7.0/24	Router C	1
	eth0.3	192.168.9.0/24	Router E	2
	eth0.4	192.168.20.0/24	Router I	1
LSPs from Router A	eth0.0	10.1.0.0/16	Directly attached	N/A
	eth0.1	192.168.1.0/24	Router B	1
	eth0.3	192.168.3.0/24	Router D	1
LSPs from Router B	eth0.1	192.168.1.0/24	Router A	1
	eth0.2	192.168.5.0/24	Router E	10
LSPs from Router C	eth0.0	10.5.0.0/16	Directly attached	N/A
	eth0.2	192.168.7.0/24	Router D	1
LSPs from Router E	eth0.0	10.7.0.0/16	Directly attached	N/A
	eth0.1	192.168.5.0/24	Router B	10
	eth0.2	192.168.9.0/24	Router D	2
	eth0.3	192.168.21.0/24	Router I	1
	eth0.4	10.6.0.0/16	Directly attached	N/A
LSPs from Router I	eth0.0	10.16.0.0/16	Directly attached	N/A
	eth0.1	192.168.20.0/24	Router D	1
	eth0.2	192.168.21.0/24	Router E	1

At this point the network is still not converged. The next step is to find the best path to each destination network. All the routers then run Dijkstra's algorithm to create an SPF tree. The process is described in Chapter 1.2.2. Once the SPF tree is completed, the routing table is created. Each router makes its decision alone, based on the information from its own routing table. For the correct functioning of DM-SPF, all the routers in an area must have complete and accurate routing information. The new Router D's routing table is described in Table 4.3. It can be seen that packets from Router D to network 10.7.0.0/16 can now use two paths with exit interfaces eth0.3 and eth0.4 allowing the load balancing of traffic.

Table 4.3 Router D's routing table after change

Network	Target	Netmask	Gateway	Metric
eth0.2	192.168.7.0	255.255.255.0	Directly attached	1
eth0.3	192.168.5.0	255.255.255.0	192.168.9.2	12
eth0.4	192.168.5.0	255.255.255.0	192.168.20.2	12
eth0.1	192.168.5.0	255.255.255.0	192.168.3.1	12
eth0.4	192.168.21.0	255.255.255.0	192.168.20.2	2
eth0.4	192.168.20.0	255.255.255.0	Directly attached	1
eth0.1	192.168.3.0	255.255.255.0	Directly attached	1
eth0.1	192.168.1.0	255.255.255.0	192.168.3.1	2
eth0.3	192.168.9.0	255.255.255.0	Directly attached	2
eth0.1	10.1.0.0	255.255.0.0	192.168.3.1	2
eth0.3	10.6.0.0	255.255.0.0	192.168.9.2	3
eth0.4	10.6.0.0	255.255.0.0	192.168.20.2	3
eth0.3	10.7.0.0	255.255.0.0	192.168.9.2	3
eth0.4	10.7.0.0	255.255.0.0	192.168.20.2	3
eth0.2	10.5.0.0	255.255.0.0	192.168.7.1	2
eth0.4	10.16.0.0	255.255.0.0	192.168.20.2	2

When the EWMA of the link load decreases to 10% of utilization, the alternative link is not needed any more and $metric\_init$ function is called. In this case default OSPF routing mechanism is used to find shortest path to desired destination and to avoid the ever-growing trend of DM-SPF metric.

The convergence is achieved, the routing tables of all the routers are at the state of consistency and the routers have complete and accurate view of the network.

5 SIMULATIONS

New method needs to be tested first in order to see if the expectations were correct and if the method performs as expected before its deployment into real networks. Therefore, simulations are preferred in the initial evaluation phases.

To ease the testing and development of projected changes to the OSPF protocol implementation, Matlab environment will be used. The testing option enables DM-SPF verification prior to deployment in a real network. Matlab offers several features, which will save time and effort and simplify measurement.

Three different scenarios are simulated. The scenarios cover the range of possible situations. The simulation of DM-SPF follows the proposal described in the previous chapter 4. The simulations of DM-SPF routing algorithm are based on the physical layout of the test network described in Figure 3.2, summarized in Table 3.1.

5.1 Scenario 1

Scenario 1 consists of three parts. The goal of the first part is to rapidly increase the load of traffic on servers for short time period (burst of heavy traffic). Such trend is shown in Figure 5.1 in the first twelve minutes of measurement. The following trend is opposite. It means that the goal is to fall the load to minimum in following twelve minutes. There is no traffic generated for next five minutes.

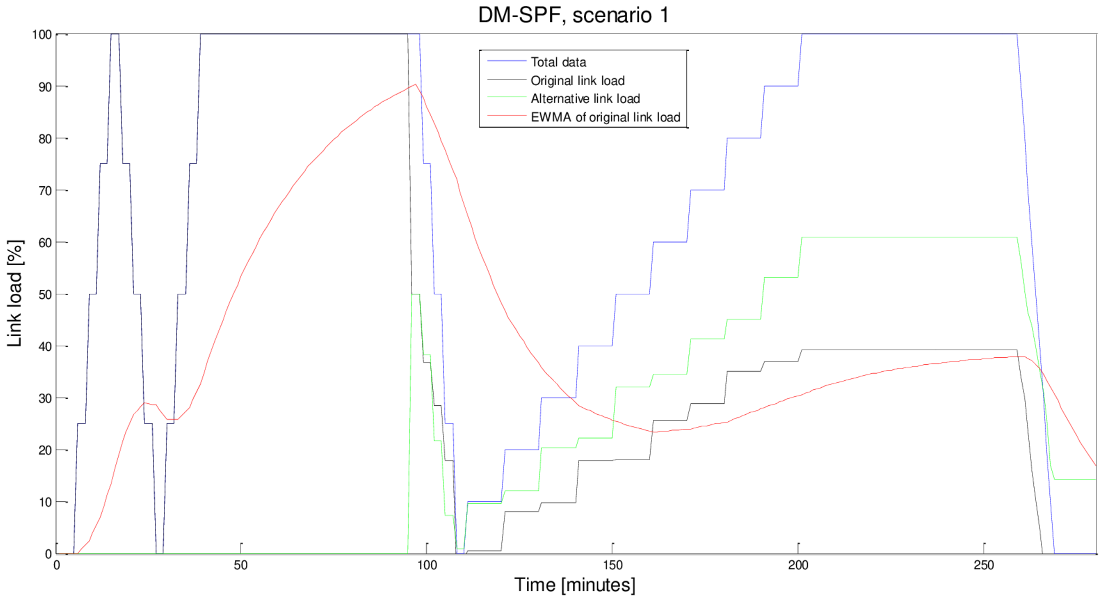


Figure 5.1 DM-SPF simulation, scenario 1

The second part follows with another rapid growing of traffic utilization. The maximal utilization remains steady over next one hour. The goal is to see if EWMA threshold of 90% is reached and load balancing process initialized. After one hour the utilization drops to zero again. There is no traffic generated for next five minutes. The third part shows a gradual increase in utilization over next 100 minutes. Over the next 60 minutes, the load remains, reaching maximal utilization. The measurement ends with final drop of utilization. The overall observation interval is set to 280 minutes.

The simulations show the influence of traffic-aware routing on the network performance. From the Figure 5.1 may be seen that the burst of heavy traffic (first part) has no impact on DM-SPF because of applied EWMA. During the second part, the higher EWMA load threshold of 90% is reached. The total data is then load balanced between original link and alternative link. The alternate link is used because lower EWMA load threshold of 10% is not reached. The last part shows the behavior of DM-SPF while alternative link is still utilized.

5.2 Scenario 2

In Scenario 2, DM-SPF was already used while monitoring the link utilization. It can be seen that the alternative link is in use on start. Scenario 2 consists of two parts as shown in Figure 5.2. The goal of the first part is to decrease the load to low value in the first fifteen minutes. The goal of the second part is to keep low utilization to reach the EWMA threshold of 10%. The overall observation interval is set to 60 minutes.

After the load drops to very low utilization and remains unchanged, the lower EWMA threshold of 10% is reached and alternative link is not used anymore. Load is switched to the original link.

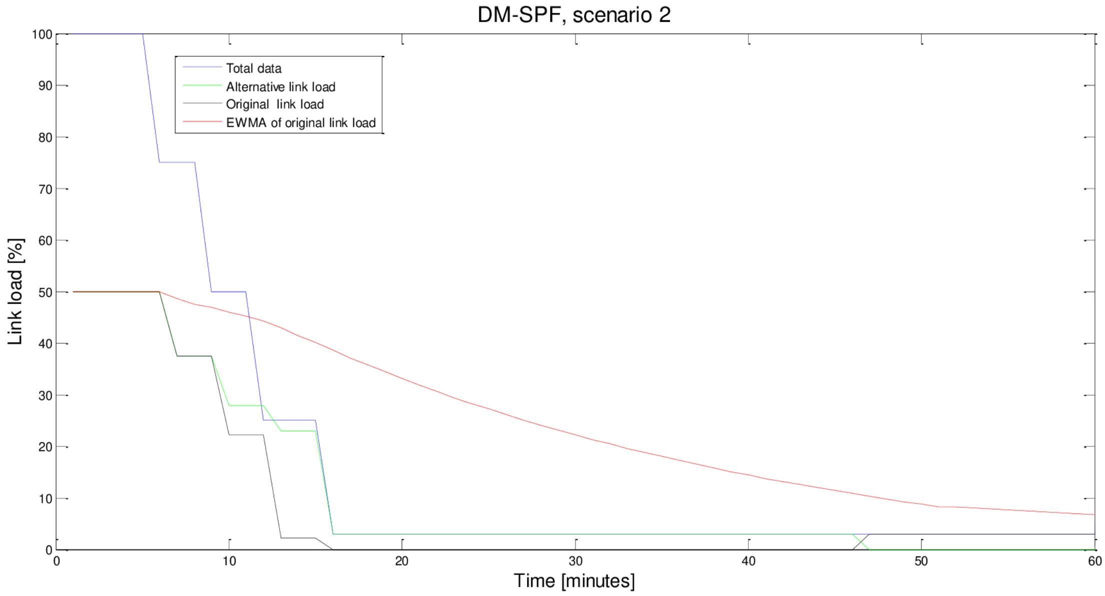


Figure 5.2 DM-SPF simulation, scenario 2

5.3 Scenario 3

The overall observation interval of scenario 3 is set to 380 minutes. The reason is to show differences between default OSPF and DM-SPF routing mechanisms in long term interval. This scenario 3, depicted in Figure 5.3, includes two parts. The first 190 minutes time period shows the behavior of default OSPF while the following 190 minutes time period shows the behavior of DM-SPF. It may be seen that the load is balanced between original and alternative link during the second period. There is no single overloaded link.

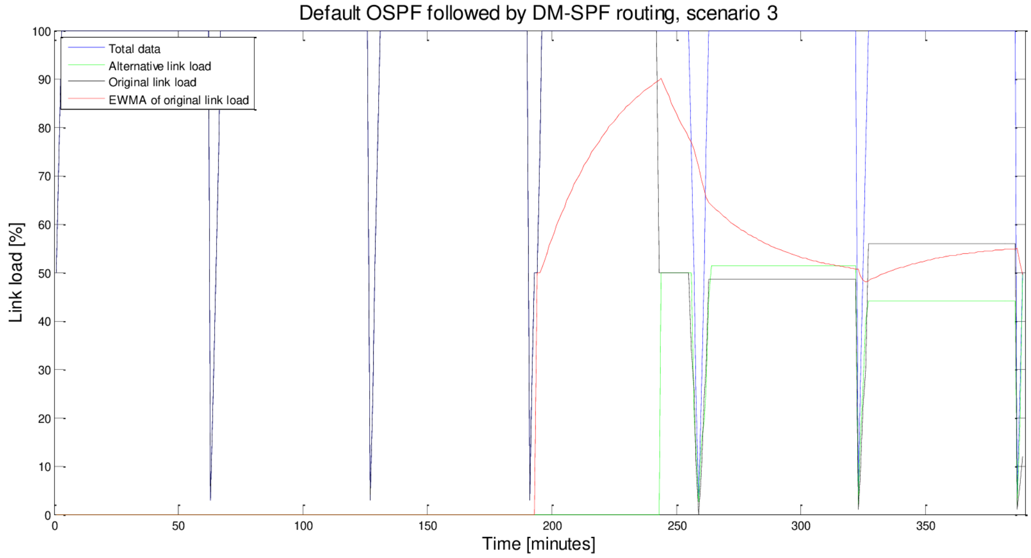


Figure 5.3 Default OSPF followed by DM-SPF simulation, scenario 3

The simulations fulfill and verify theoretical expectations and testing in real network may proceed.

6 EXPERIMENTAL EVALUATION OF PROPOSED METHOD IN REAL NETWORK

After the simulation and evaluation of DM-SPF using Matlab, testing in a network under real conditions follows. Mentioned logical network design forms the foundation for following physical design. Such network needs to be suitably designed for appropriate routing traffic. The advantages of IP-based networks are the robustness, increased reliability or the fact that the transmission of packets can be dynamically adapted to changing network conditions such as network failures.

The routers run the trial version of DM-SPF routing algorithm. This experimental evaluation of method proposed follows the network topology described in Figure 3.2. The IP addressing scheme for the topology using DM-SPF is summarized in Table 3.1. Router Linksys WRT54GL running OpenWrt is again used for testing. Three scenarios are tested as mentioned and simulated in chapter 5.

6.1 Scenario 1

This chapter brings the series of fundamental graphs. Measured data were captured with the help of PRTG monitoring tool. Figure 6.1 shows total traffic between users and servers. From the plot can be seen that the scenario 1 consists of three parts as mentioned in chapter 5.1. Detailed information of routers and end devices including routing tables and graphs are summarized in Appendix B.

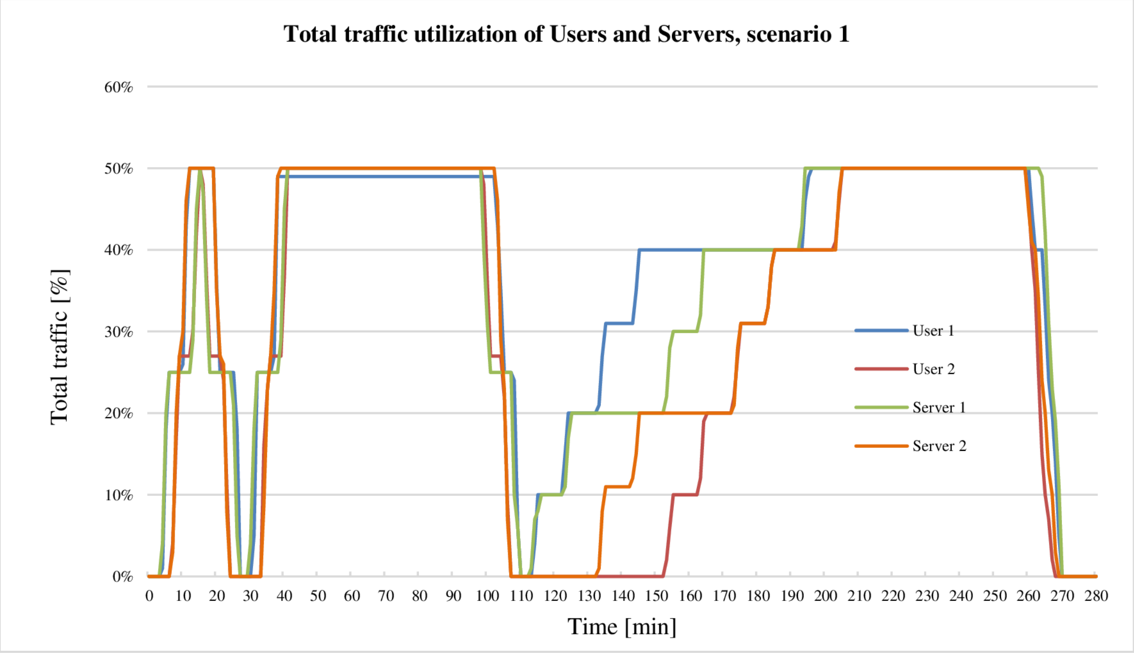


Figure 6.1 Total traffic utilization of Users and Servers, scenario 1

The most affected routers are Router D, E and I. Figure 6.2 depicts total traffic utilization on Router D. The red curve outlines the utilization on interface eth0.3. It may be seen that the utilization on this interface follows default OSPF behavior in the first 90 minutes because $metric$ function is used. Since EWMA is applied, the burst of heavy traffic has no impact on DM-SPF routing mechanism. The situation changes in minute 91. EWMA threshold of 90% is reached and the $metric\_load\_balance$ function is called. DM-SPF initializes load balancing process and Router I is involved into routing process. The traffic is balanced between eth0.3 and eth0.4. The utilization of eth0.3 interface is reduced accordingly. It may be seen that the traffic does not exceed high utilization limits.

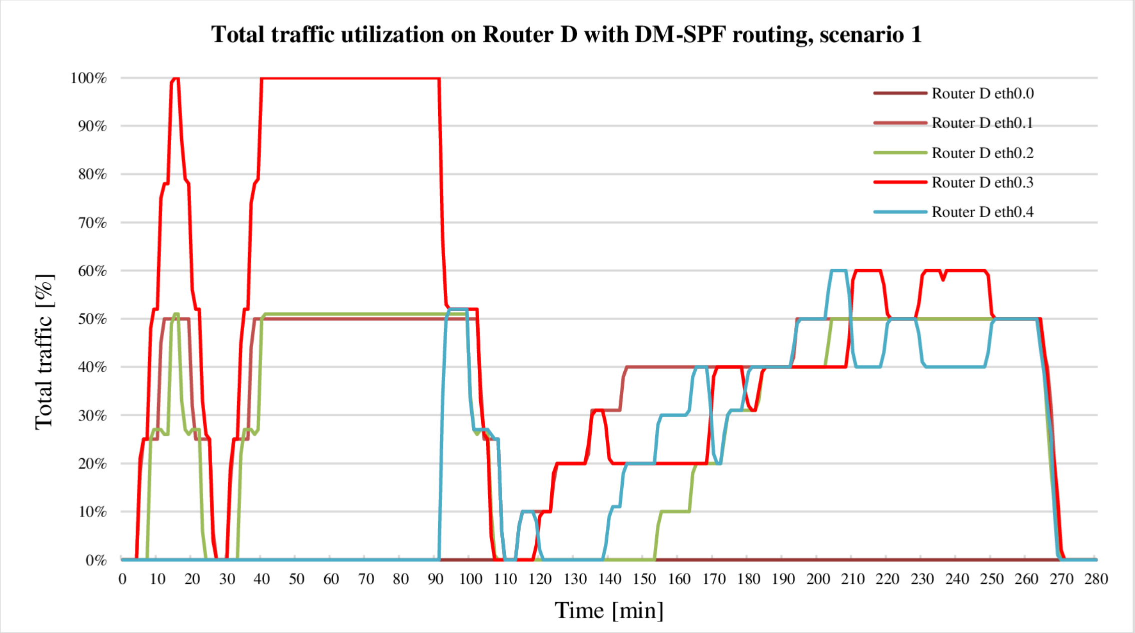


Figure 6.2 Total traffic utilization on Router D with DM-SPF routing, scenario 1

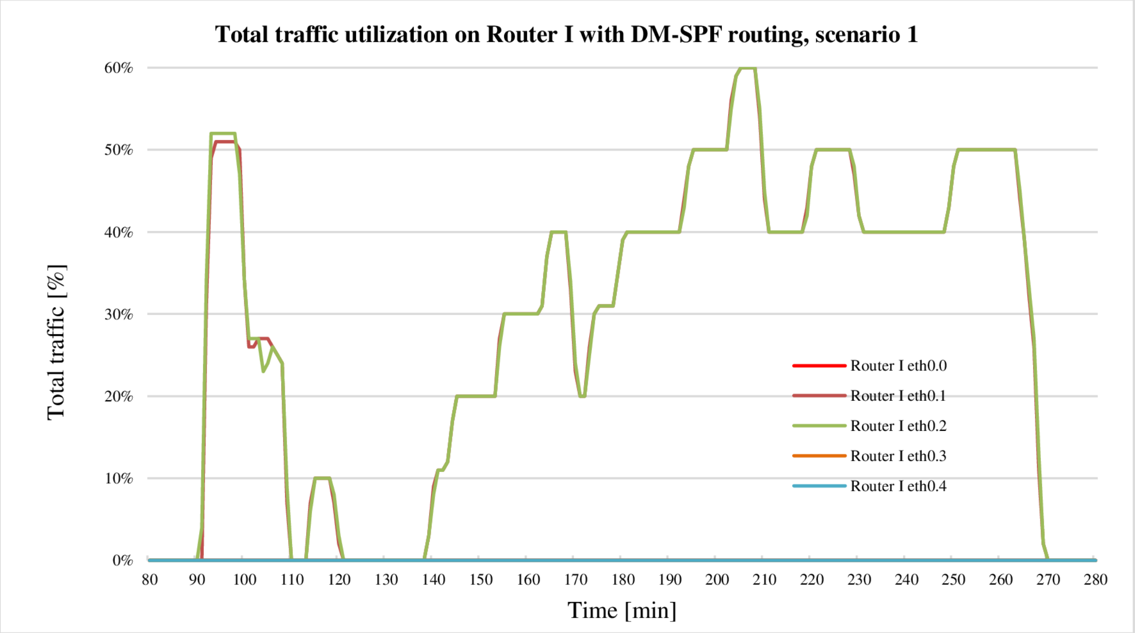


Figure 6.3 Total traffic utilization on Router I with DM-SPF routing, scenario 1

Figure 6.3 depicts the traffic utilization on Router I. There is no load on Router I before minute 91. The load is then balanced and links connected to interfaces eth0.1 and eth0.2 form the alternative path. It is important to mention that the utilization of some interfaces is almost identical and the curves are overlapping.

6.2 Scenario 2

The second scenario describes the situation when DM-SPF load balancing is already used while monitoring the link utilization. Scenario 2 consists of two parts as shown in Figure 6.4.

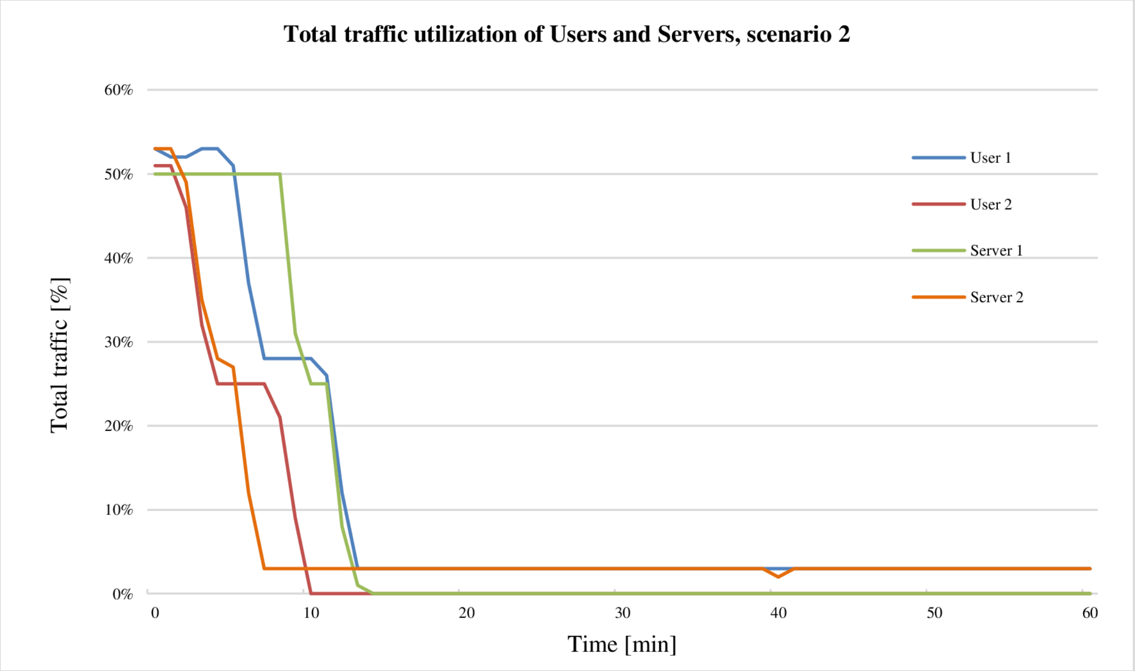


Figure 6.4 Total traffic utilization of Users and Servers, scenario 2

This figure shows the traffic generated by users and servers. The goal of the first part is to decrease the load to low value. The goal of the second part is to keep low utilization to reach the EWMA threshold of 10%.

The main noticeable fact about the graph in Figure 6.5 is the traffic switch between eth0.3 and eth0.2 on Router E. If the utilization remains low and the lower threshold is reached, *metric_init* function is called. The load is shifted from alternative to original link in minute 40.

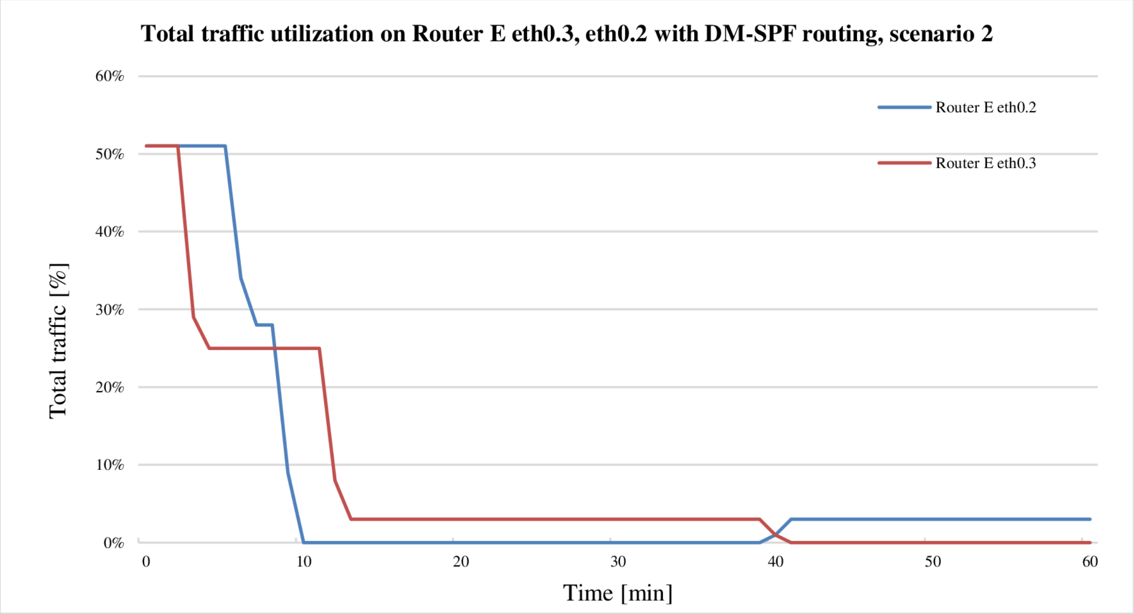


Figure 6.5 Total traffic utilization (Router E eth0.3, eth0.2) DM-SPF routing, scenario 2

Figure 6.6 shows the load of Router I. It can be seen that the alternative link is in use on start. There is no need to use alternative link after the minute 40.

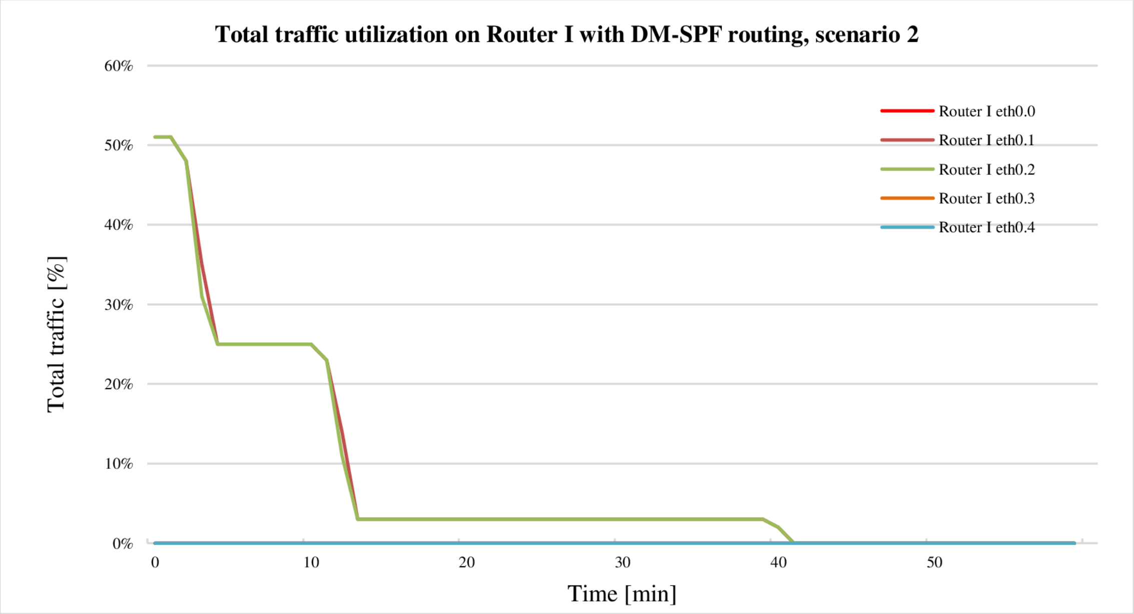


Figure 6.6 Total traffic utilization on Router I with DM-SPF routing, scenario 2

6.3 Scenario 3

The goal of scenario 3 is to highly utilize the links and compare packet loss and coverage with and without running DM-SPF load balancing. This scenario consists of two parts. The first half of the whole observation does not run DM-SPF. The algorithm is enabled for the second half of this measurement. The total utilization of users and servers is depicted in Figure 6.7.

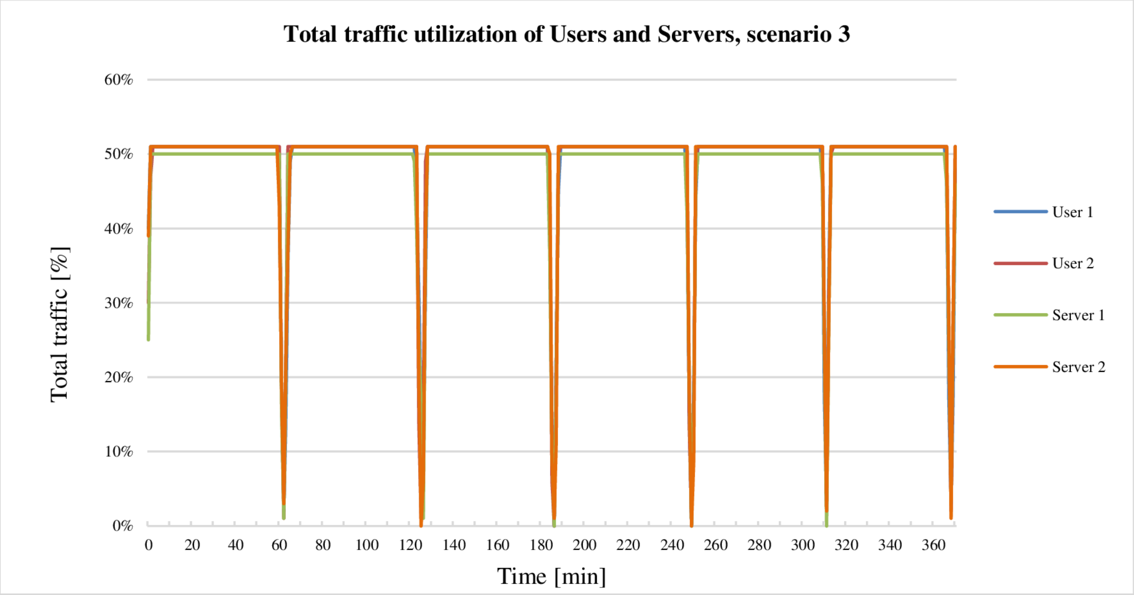


Figure 6.7 Total traffic utilization of Users and Servers, scenario 3

Figure 6.8 shows the load balancing of traffic between eth0.2 and eth0.3 on Router E. The *metric_load_balance* function is called around minute 240. Idle interface eth0.3 is utilized after the shift. It means that the traffic is then passing Router I which is shown in Figure 6.9.

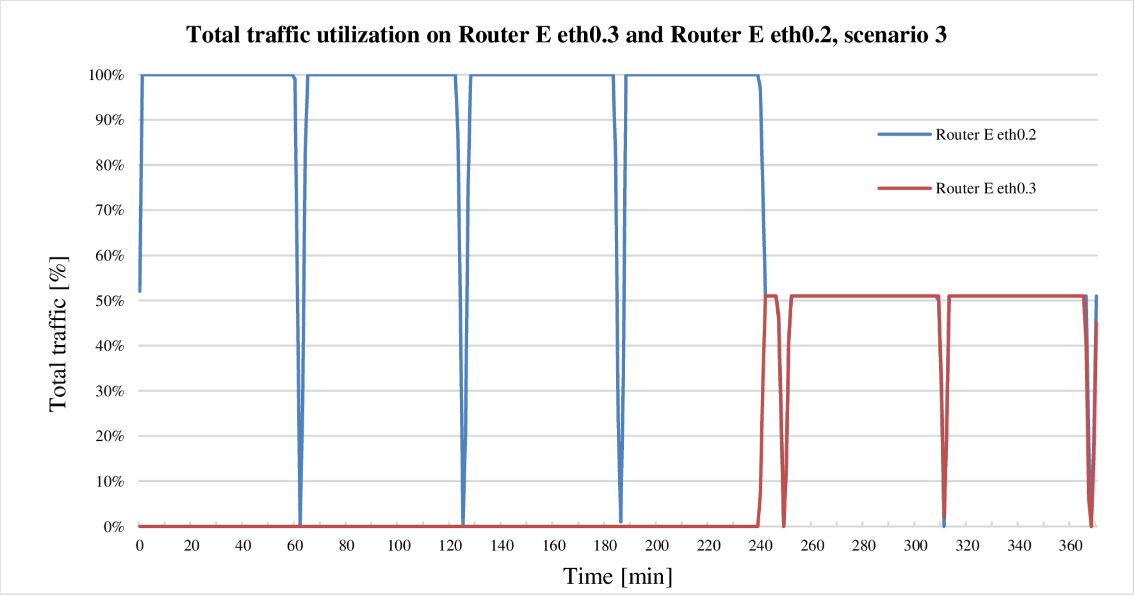


Figure 6.8 Total traffic utilization on Router E eth0.3, eth0.2, scenario 3

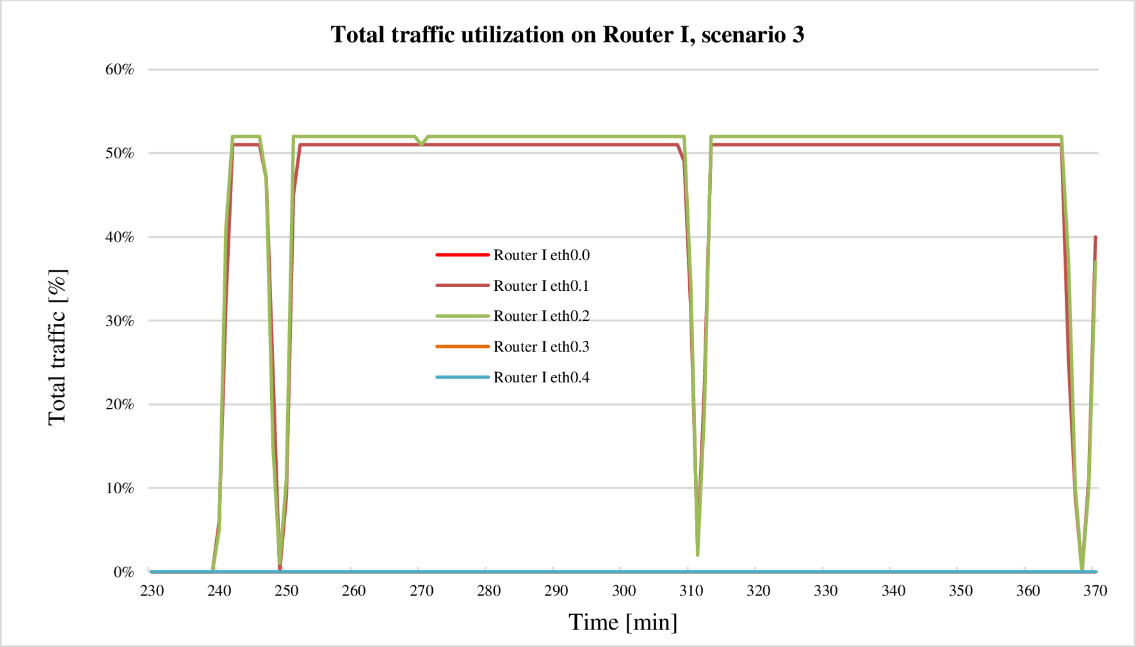


Figure 6.9 Total traffic utilization on Router I, scenario 3

All three scenarios were successfully tested in real environment. The results of monitoring packet loss and coverage are summarized in the next chapter.

7 COMPARISON OF OSPF AND DM-SPF ROUTING ALGORITHMS

This chapter focuses on DM-SPF characteristics and the comparison of default OSPF and DM-SPF method. The comparison is mainly based on values resulting from tests performed in real network environment. This chapter summarizes results from monitoring packet loss and coverage of scenario 1 and 3. There is no need to focus on scenario 2 because the load balancing is already used while monitoring the link utilization. Advantages and also disadvantages of DM-SPF are highlighted.

7.1 DM-SPF advantages

The main advantage is that the method proposed takes current link load into consideration. This approach gives an advantage of congestion avoidance and reduces its data rates in the case of congestion. The network performance is improved by supporting load balancing across parallel links. The result is that DM-SPF distributes the traffic among multiple paths to the same destination to use bandwidth efficiently.

There are some common advantages coming from OSPF. DM-SPF supports scalability and the growing of a network. It automatically adapts to topology changes and is indifferent to the network size. DM-SPF has no impact on Administrative Distance (AD). DM-SPF is expected to perform fast convergence.

The following part is focused on the mentioned comparison emphasizing DM-SPF advantages. Table 7.1 summarizes the comparison of OSPF and DM-SPF routing in scenario 1. Two main parameters called coverage and packet loss are observed. The term coverage stands for the ability of monitoring system to get reliable and complete data from all the monitored devices. Packet loss is chosen as an indicator of congestion and highly utilized devices. Default OSPF routing under high utilization caused packet loss, low throughput and no response to traffic congestion. Table 7.1 demonstrates that DM-SPF ensures maximal coverage and brings no packet loss compared to default OSPF.

Table 7.1 Comparison of OSPF and DM-SPF routing algorithm, scenario 1

Device	Scenario 1			
	Coverage		Packet Loss	
	Default	DM-SPF	Default	DM-SPF
Router A	92.31 %	100.00 %	3.89 %	0.00 %
Router B	93.41 %	100.00 %	6.74 %	0.00 %
Router C	96.70 %	100.00 %	7.54 %	0.00 %
Router D	96.70 %	100.00 %	7.33 %	0.00 %
Router E	98.90 %	100.00 %	0.55 %	0.00 %
Router I	97.80 %	100.00 %	7.85 %	0.00 %
User 1	95.60 %	100.00 %	2.24 %	0.00 %
User 2	98.90 %	100.00 %	1.88 %	0.00 %
Server 1	98.90 %	100.00 %	0.33 %	0.00 %
Server 2	96.70 %	100.00 %	1.32 %	0.00 %

Figure 7.1 depicts packet loss in scenario 1 in the first ninety minutes. There is no packet loss after the $metric\_load\_balance$ function is called.

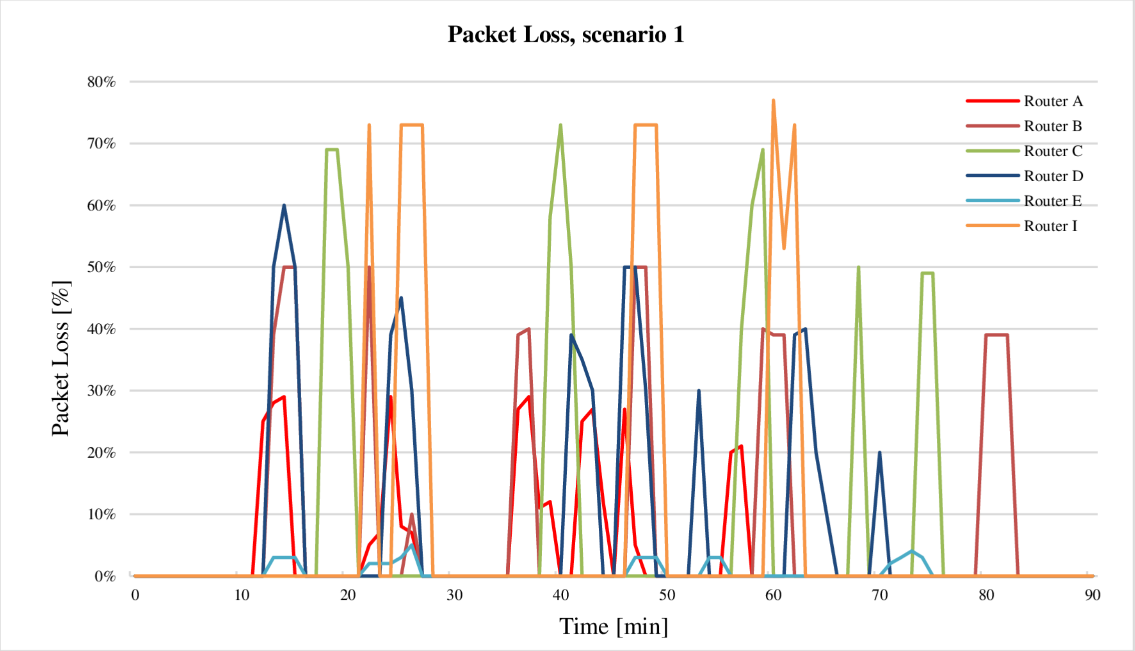


Figure 7.1 Packet Loss on the routers, scenario 1

Figure 7.2 shows the coverage of all the devices in scenario 1. The coverage drops to 0% if the network is overloaded. There is no coverage drop after minute 90 because the traffic is balanced.

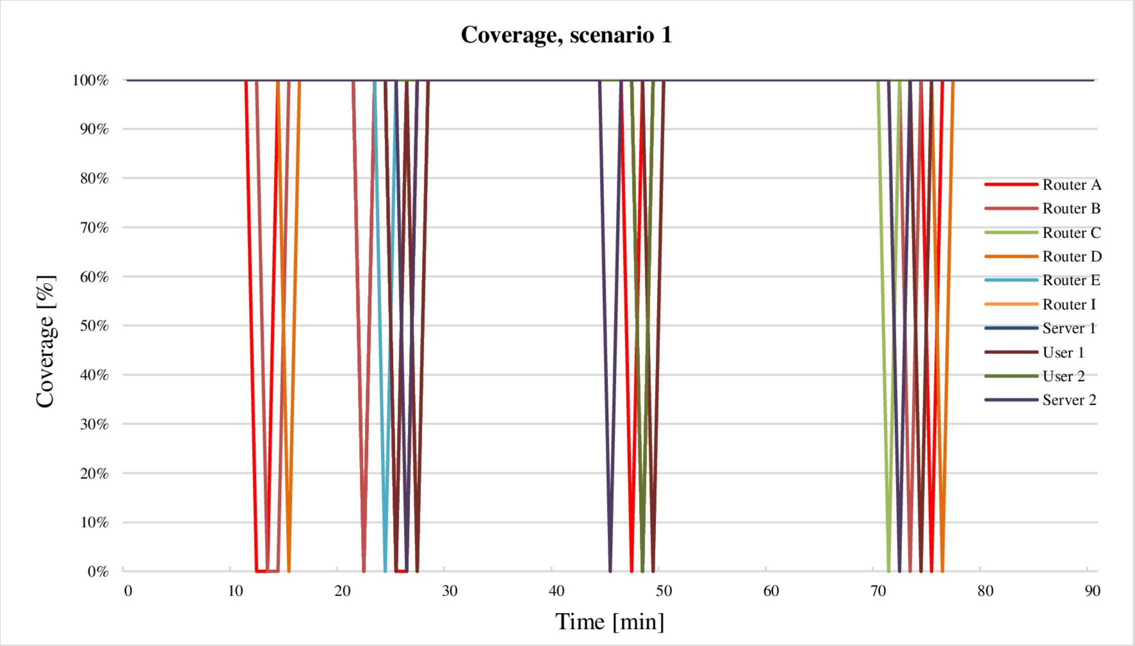


Figure 7.2 Coverage of devices, scenario 1

From the graphs in Figure 7.3 can be seen that the CPU of Router D and Router E is slightly decreased during high utilization when comparing before and after the $metric\_load\_balance$ function is called. Packet loss and coverage measurements of scenario 3 are summarized in Table 7.2.

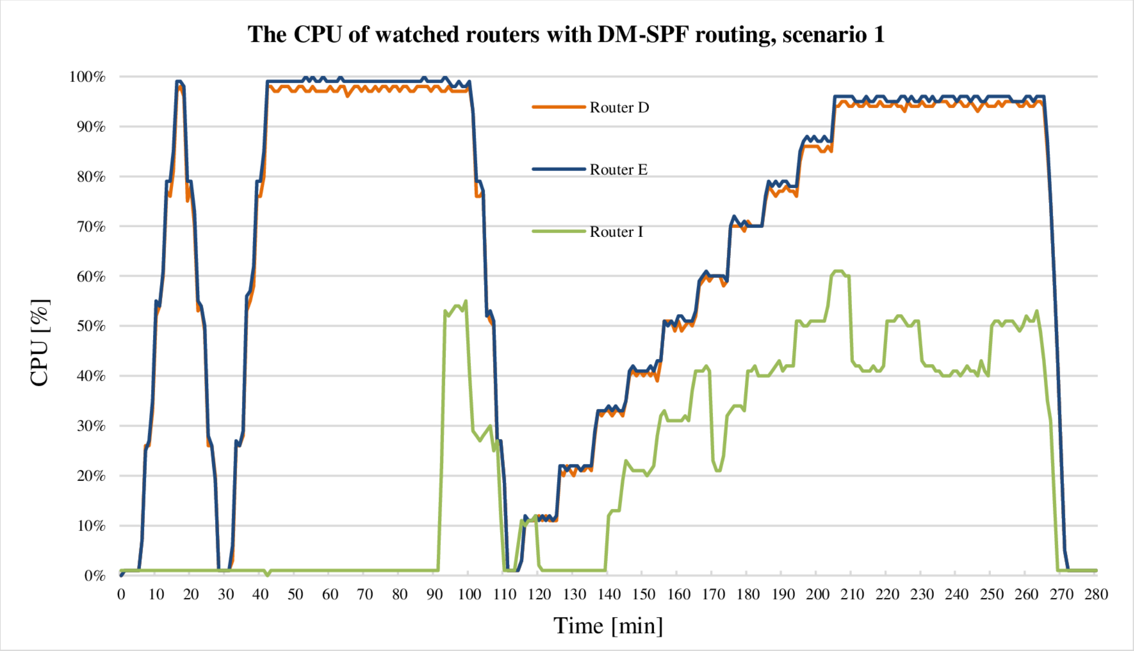


Figure 7.3 The CPU of watched routers with DM-SPF routing, scenario 1

Table 7.2 Comparison of default OSPF and DM-SPF routing algorithm, scenario 3

Device	Scenario 3			
	Coverage		Packet Loss	
	Default	DM-SPF	Default	DM-SPF
Router A	98.38 %	100.00 %	2.07 %	0.00 %
Router B	98.38 %	100.00 %	0.91 %	0.00 %
Router C	97.57 %	100.00 %	1.99 %	0.00 %
Router D	98.65 %	100.00 %	2.78 %	0.00 %
Router E	97.84 %	100.00 %	2.95 %	0.00 %
Router I	98.65 %	100.00 %	0.54 %	0.00 %
User 1	98.92 %	100.00 %	1.74 %	0.00 %
User 2	98.38 %	100.00 %	1.95 %	0.00 %
Server 1	99.59 %	100.00 %	0.17 %	0.00 %
Server 2	98.65 %	100.00 %	1.00 %	0.00 %

It can be concluded from the conducted measurements that in specific cases not addressed by original OSPF, the DM-SPF approach decreases packet loss, increases coverage values and brings improved performance. The load balancing process starts around minute 240. Figure 7.4 illustrates the packet loss before and after the load balancing. It can be seen that routers running DM-SPF are not heavily utilized which leads to no packet loss.

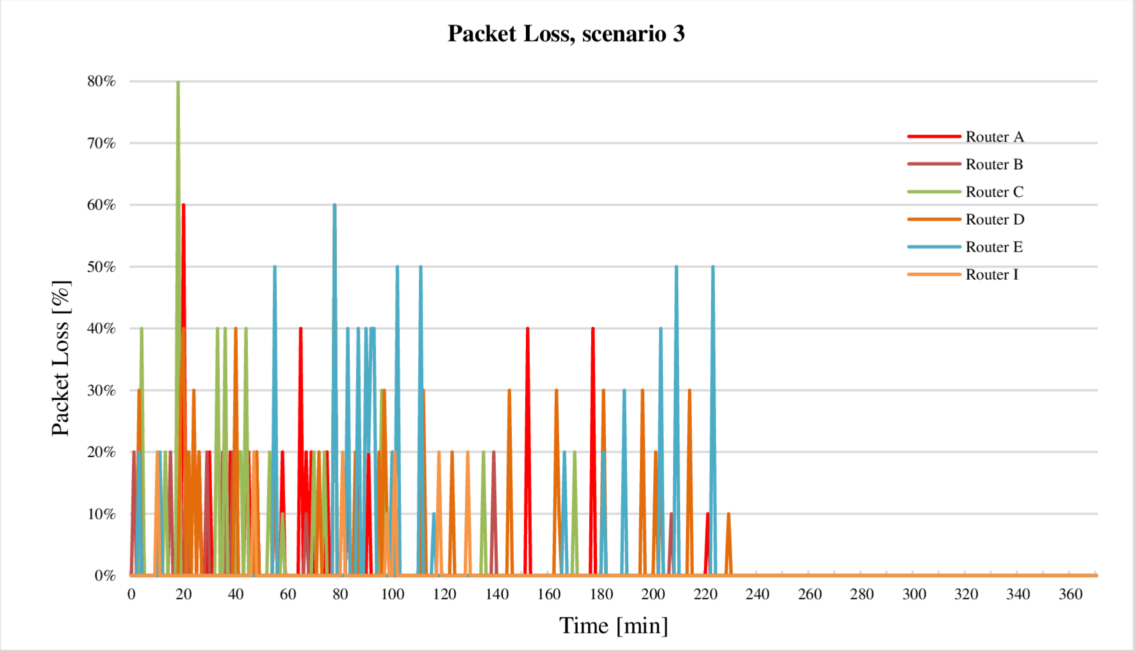


Figure 7.4 Packet Loss, scenario 3

Figure 7.5 shows the coverage of the devices. Again, in the first part of measurement, the coverage drops to minimal value. On the other hand, when load balancing is in place, the coverage is 100%.

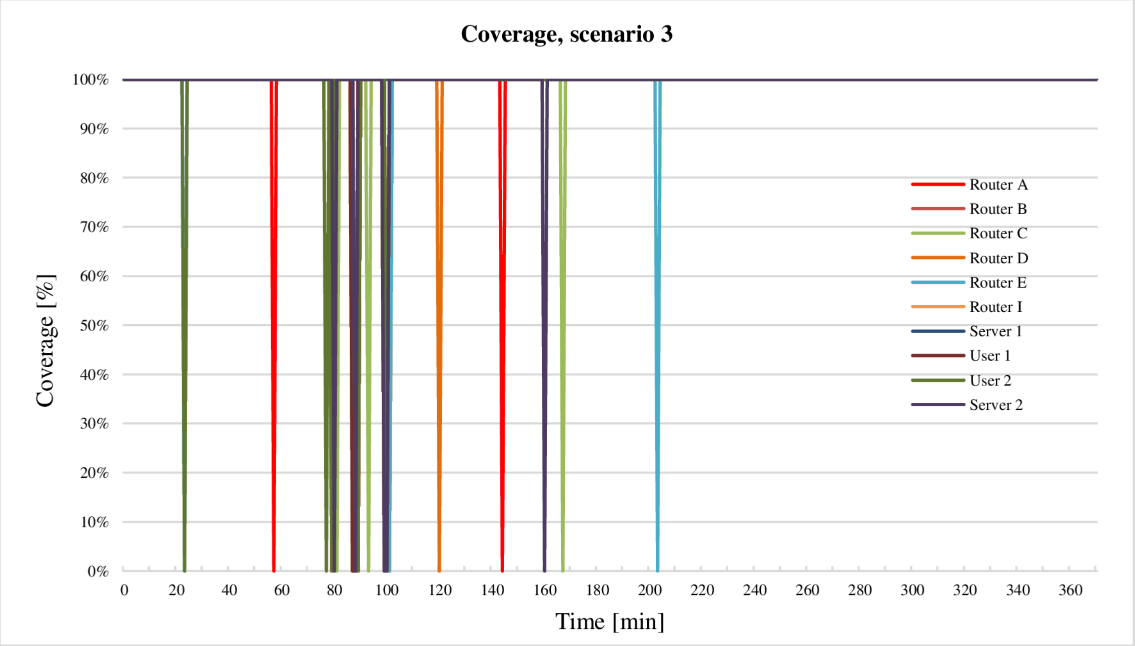


Figure 7.5 Coverage, scenario 3

Decreasing of router's utilization slightly reduces CPU utilization. Figure 7.6 depicts three CPU graphs (Router D, E and I). The CPU utilization of Router D and E is less than 100% while load balancing.

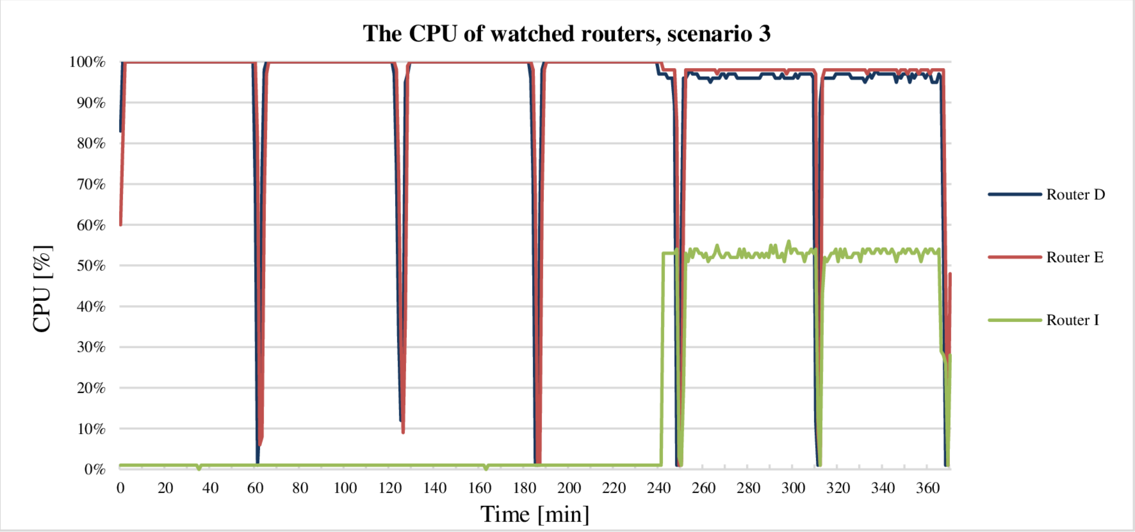


Figure 7.6 The CPU of watched routers, scenario 3

Any protocol that changes routes quickly can become unstable. The instability arises because IP traffic can change dynamically. Two-stage oscillation effect can occur. In this case, the traffic switches between two paths back and forth. An example of routing oscillation is shown in Figure 7.7. Let's assume that Dijkstra's algorithm was processed, minimal total metrics were determined and routing tables were created. There are two paths between Router1 and Router 2. In this simple example, according to the minimal total metric, only path A is used. If impropriate routing mechanism is deployed and the routing changes, path B takes the load of the path A. Let's assume that after some time path B becomes heavily loaded. The routing situation can change again and the traffic is switched back onto the path A. Such oscillations have undesirable consequences like inefficient bandwidth usage and frequent route computation. The benefit of DM-SPF is to quickly converge without the risk of oscillations because the idea is to achieve traffic load balancing. The convergence time depends on traffic, smoothing factor (60 seconds) and the time of the routing table creation process.

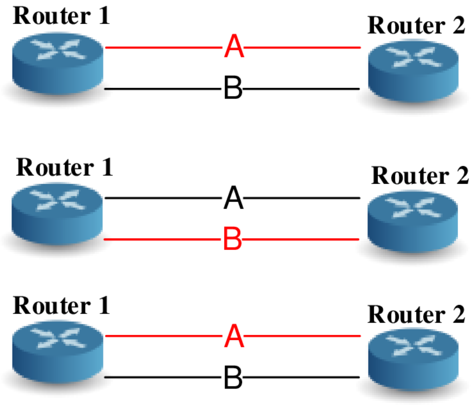


Figure 7.7 Routing oscillation

7.2 DM-SPF disadvantages

The method is an improvement but it is not a final solution to the traffic optimization problem. As is usually the case, the method has a possible cost. The traffic load caused by DM-SPF itself must be taken into consideration. The question is, how much bandwidth is used to send routing updates. It was found that the algorithm uses little bandwidth. Increasing number of transmitted control traffic causes short-term memory usage and CPU consumption due to the use of link state databases and the creation of the SPF tree.

A change in the topology is represented as a change in one or more of the LSPs. LSPs flooding process may degrade performance. The change causes new LSP flooding and new routing tables are computed. However, even this amount of necessary traffic is undesirable. Regardless of how well the router can handle the packets, the packet's overhead will have some impact on performance. When a router starts looking into a packet deeper and deeper, its performance will suffer. After the LSPs are flooded and stored in a database, each router is responsible for calculating the SPF tree for each known destination. On the other hand, LSP flooding is limited inside the affected area.

CONCLUSION

Being able to reliably communicate to anyone and anywhere plays an important role in our personal and business lives. In order to support immediate delivery of information being exchanged between people, users rely on effectively working interconnected networks. These networks are dependent on proper routing. Routing is a sophisticated network function that requires cooperation between routers and routing protocols.

The thesis offered the comparison of all routing protocols currently used in the Internet. There is a number of routing protocols from different manufacturers, but this thesis is focused on OSPF protocol, as it is commonly used. OSPF was discussed in depth. With gained pieces of knowledge it was revealed that there are still challenges to solve in OSPF routing process. The OSPF routing is limited by static metrics. OSPF may dynamically route around link failures but not around congestions. Such limitation was described in detail in chapter 3.

A simple testbed was designed and created. This testbed represented a network testing environment that enabled to validate default OSPF routing behavior. From the results, it was obvious how OSPF drawbacks affected network performance. Default OSPF routing under high utilization caused packet loss, low throughput and no response to traffic congestion. Therefore, the main focus of this thesis was on a new method for better route selection while traffic load was also taken into account.

This thesis offered an approach of network congestion avoidance caused by individual links. The primary objective of the routing protocol was to determine the best paths for each route to include in the routing table. The routing algorithm generated a metric for each path through the network. This metric was based on two characteristics of a path - bandwidth and link load. It means that the novel method based route selection on multiple characteristics, combining them into a single metric. In general, the smaller the value of the metric led to the better path. The new method supported multipath routing. When there were multiple paths between DM-SPF routers, the paths shared the load.

When load became a variable used in determining a route and was used as a part of composite metric, DM-SPF was designed to avoid the impact of unexpected and accidental peak utilization. Such a burst of heavy traffic could destabilize the network. To ensure dynamic control of link load observations of routers, appropriate technique had to be applied. Exponentially Weighted Moving Average represented such tool for monitoring process variability. EWMA was more sensitive to small shifts in the process as compared to other techniques mentioned in the thesis.

The routing software was made as a process program providing all of the functionalities of a routing protocol. Not only when one of the links became utilized but also if there was a change in the topology, DM-SPF invoked the affected router to update the corresponding records in its database and warned the other nodes in that area. The goal was to split the traffic across multiple paths. To prevent frequent metric changes, load calculation was influenced by EWMA. The routers updated their link state databases, rerun the SPF algorithm, created a new SPF tree, and updated their routing tables. After a change, all routers had to agree on the network topology and achieve convergence.

The method was tested in a simulator and under real conditions. For the simulation of the method, Matlab was chosen as the most appropriate simulation tool since it offered flexibility and all the required functionalities. Performed simulations proved the proper function of the method and testing in the network under real conditions could follow.

There were three scenarios tested. These scenarios covered all possible situations that could occur. The results were discussed and described in graphs and tables. The advantages and disadvantages of DM-SPF were summarized in chapter 7. The method measurably increased the effectiveness and performance of the network in mentioned cases. This method not only made a best path determination but also a new best path when the initial path became unusable or if the topology changed. For these reasons, the method had an advantage over default OSPF routing protocol with its key feature - traffic awareness. DM-SPF made the most efficient use of the available bandwidth and distributed traffic across multiple paths.

As with every advance in communication technology, the creation of new methods had its drawbacks. Increased number of transmitted control traffic caused short-term memory usage and CPU consumption.

It was verified that DM-SPF met predefined goals. The proper functions and benefits were proved, examined and described. It was proved that the new method is able to support robust, reliable and efficient network.

Different people have different approaches to designing OSPF networks. The important thing to consider was that any protocol can fail under pressure. The idea was not to change the OSPF protocol but to adapt it in order to get the best performance.

REFERENCES

[1] BRADEN, R. RFC 1122: Requirements for Internet Hosts -- Communication Layers. Internet Engineering Task Force. 1989.

[2] BRADEN, R. RFC 1123: Requirements for Internet Hosts -- Application and Support. Internet Engineering Task Force. 1989.

[3] POSTEL, J. RFC 791: Internet Protocol DARPA Internet Program Protocol Specification. Internet Engineering Task Force. 1981.

[4] POSTEL, J. RFC 793: Transmission Control Protocol. Internet Engineering Task Force. 1981.

[5] POSTEL, J. RFC 768: User Datagram Protocol. Internet Engineering Task Force. 1980.

[6] ISO/EIC 7498-1. Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model. 1994.

[7] ITU-T. Open System Interconnection. Series X.2000.

[8] MOY, J. RFC 1584: Multicast Extensions to OSPF. Internet Engineering Task Force. 1994.

[9] COLTUN, R., FULLER, V. RFC 1587: The OSPF NSSA Option. Internet Engineering Task Force. 1994.

[10] MOY, J. RFC 1131: The OSPF Specification. Internet Engineering Task Force. 1989.

[11] MOY, J. RFC 2328: OSPF Version 2. Internet Engineering Task Force. 1998.

[12] COLTUN, R., FERGUSON, D., MOY, J. RFC 2740: OSPF for IPv6. Internet Engineering Task Force. 1999.

[13] ALMQUIST, P. RFC 1349: Type of Service in the Internet Protocol Suite. Internet Engineering Task Force. 1992.

[14] COLTUN, R. RFC 2370: The OSPF Opaque LSA Option. Internet Engineering Task Force. 1998.

[15] ROSEN, E., PSENAK, P., PILLAY-ESNAULT, R. RFC 4576: Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs). Internet Engineering Task Force. 2006.

[16] PSENAK, P., MIRTORABI, S., ROY, A., NGUYEN, L., PILLAY-ESNAULT, R. RFC 4915: Multi-Topology (MT) Routing in OSPF. Internet Engineering Task Force. 2007.

[17] DIJKSTRA, E., W. A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1. pp. 269-271.1959.

[18] Muniswamy, V.V. Design and Analysis of Algorithms. I K International. 272 p. 2009. ISBN: 978-9380026732.

[19] MEHLHORN, K., SANDERS, P. Algorithms and Data Structures: The Basic Toolbox. Springer. 300 p. 2008. ISBN: 978-3540779773.

[20] CHANG, S-K. Data structures and algorithms. World Scientific Publishing Company. 360 p. 2003. ISBN: 978-9812383488.

[21] TEARE, D. Implementing Cisco IP Routing (ROUTE). Foundation Learning Guide. Foundation Learning for the ROUTE 642-902 Exam. Cisco Press. 945 p. 2010. ISBN: 978-1-58705-882-0.

[22] BLANK, A., G. TCP/IP Foundations. John Wiley & Sons. 304 p. 2004. ISBN: 978-078214306.

[23] HEDRICK. C. RFC 1058: Routing Information Protocol. Internet Engineering Task Force. 1988.

[24] MALKIN, G. RFC 2453: RIP Version 2. Internet Engineering Task Force. 1998.

[25] KOZIEROK, CH. M. The TCP/IP guide: a comprehensive, illustrated Internet protocols reference. No Starch Press. 1616 p. 2005. ISBN: 978-1593270476.

[26] HUITEMA, CH. Routing in the Internet (2nd Edition). 385 p. 1999. ISBN-13: 978-0130226471.

[27] MALKIN, G., MINNEAR. R. RFC 2080: RIPng for IPv6. Internet Engineering Task Force. 1997.

[28] CALLON, R. RFC 1195: Use of OSI IS-IS for Routing in TCP/IP and Dual Environments. Internet Engineering Task Force. 1990.

[29] GREDLER, H., GORALSKI, W. The Complete IS-IS Routing Protocol. Springer. 540 p. 2005. ISBN: 978-1852338220.

[30] MARTEY, A. IS-IS Network Design Solutions (Cisco Core). Cisco Press, 416 p. 2002. ISBN: 978-1578702206.

[31] HENDRICK, C. L., BOSACK, L. An Introduction to IGRP. 1989.

[32] MEDHI, D., RAMASAMY, K. Network Routing: Algorithms, Protocols, and Architectures (The Morgan Kaufmann Series in Networking). Morgan Kaufman. 824 p. 2007. ISBN: 978-0120885886.

[33] AZIZ, Z. et al. Troubleshooting IP Routing Protocols (CCIE Professional Development). Cisco Press. 912 p. 2002. ISBN: 978-1587050190.

[34] MILLS, D. L. RFC 904: Exterior Gateway Protocol Formal Specification. Internet Engineering Task Force. 1984.

[35] LOUGHEED, K. A. RFC 1105: Border Gateway Protocol (BGP) Internet Engineering Task Force. 1989.

[36] REKHTER, Y., LI, T. RFC 1771: A Border Gateway Protocol 4 (BGP-4). Internet Engineering Task Force. 1995.

[37] REKHTER, Y., LI, T., HARES, S. RFC 4271: A Border Gateway Protocol 4 (BGP-4). Internet Engineering Task Force. 2006.

[38] RIVEST, R. RFC 1321: The MD5 Message-Digest Algorithm. Internet Engineering Task Force. 1992.

[39] ITU-T. Recommendation Y. 2001. SERIES Y: GLOBAL INFORMATION. INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS. Next Generation Networks – Frameworks and functional architecture models. General overview of NGN. 2004.

[40] POIKSELKA, M.; MAYER, G.; KHARTABIL, H.; NIEMI, A. The IMS: IP Multimedia Concepts and Services in the Mobile Domain. WILEY, 2004. 448 p. ISBN-10 0-470-87113-X.

[41] XU, J. Practical WPF Charts and Graphics (Expert's Voice in .NET). APRESS. 2009. 684 p. ISBN: 978-1430224815.

[42] ROBERTS, S. W. Control chart tests based on geometric moving averages. Technometrics.

[43] HUNTER, J. S. The Exponential Weighted Moving Average. Journal of Quality Technology.

[44] ZHOU, H., PAN, J., SHEN, P. Cost Adaptive OSPF. Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications. 2003.

[45] GUERIN, R., ORDA, A., WILLIAMS, D. QoS routing mechanisms and OSPF extensions. In Proceedings of IEEE GLOBECOM. Phoenix. AZ. pp. 1903–1908. November 1997.

[46] WANG, Z., CROWCROFT, J. Quality-of-service routing for supporting multimedia applications. IEEE Journal on Selected Areas of Communication. vol. 14, pp. 1288– 1234. September 1996.

[47] MA, Q., STEENKISTE, P., ZHANG, H. Routing high-bandwidth traffic in max-min fair share network. In Proceedings of ACM SIGCOMM. Stanford. CA. pp. 206–217. August 1996.

[48] WANG, J., NAHRSTEDT, K. Hop-by-hop routing algorithms for premium class traffic in DiffServ networks. In Proceedings IEEE INFOCOM. pp. 705–714. June 2002.

[49] APOSTOLOPOULOS, G., KAMAT, S. GUERIN, R. Implementation and Performance Measurements of QoS Routing Extensions to OSPF. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. pp. 680-688. 1999.

[50] FORTZ, B., THORUP. M. Increasing Internet capacity using local search. Computational Optimization and Applications. 2004. pp.13-48.

[51] FORTZ, B., REXFORD, J., THORUP. M. Traffic engineering with traditional IP routing protocols. IEEE Communication Magazine, vol. 40, pp. 118-124. October 2002.

[52] ISHIGURO, K., et al. Quagga. A routing software package for TCP/IP networks. 2011.

[53] Cisco. Wireless-G Linux Broadband router, Model: WRT54GL v1.1. Datasheet. Linksys by Cisco.

[54] CHERRY, N. Linux Smart Home for Dummies. WILEY. 364 p. 2006. ISBN: 978-0764598234.

[55] SCHOFFSTALL, M., et al. A Simple Network Management Protocol (SNMP). RFC 1157. 1990.

[56] MACHA, T.; KRKOS, R.; NOVOTNY, V. Proposal of load aware routing for OSPF routing protocol. Communications, 2013, 15, n. 2a/ 2013, s. 139-144. ISSN: 1335- 4205.

[57] MACHA, T.; KRKOS, R.; NOVOTNY, V. OSPF Alternate Costing Strategy. In 2012 International Conference on Telecommunication Systems, Modeling and

Analysis (ICTSM2012). Prague: Czech Technical University in Prague, 2012.s. 136-140. ISBN: 978-0-9820958-6- 7.

[58] MACHA, T.; KRKOS, R. A novel approach to OSPF metric calculation. In Research in Telecommunication Technologies 14th International Conference Proceedings. 2012. s. 85-91. ISBN: 978-80-554-0570- 4.

[59] MACHA, T.; NOVOTNY, V.; MARTINASEK, Z.; NAGY, L. DiffServ and IntServ Mapping in IMS. In 33nd International Conference on Telecommunication and Signal Processing - TSP' 2010. 2010. s. 325-330. ISBN: 978-963-88981-0- 4.

[60] MACHA, T.; NAGY, L.; MARTINASEK, Z.; NOVOTNY, V. IMS Mapping of QoS Requirements on the Network Level. Elektrorevue. 2010, n. 11, s. 1-6. ISSN: 1213- 1539.

[61] MACHA, T.; NAGY, L.; NOVOTNY, V. Support of IP Multimedia Subsystem in the development tool SDS Ericsson 4.1. Elektrorevue, 2010, n. 21, s. 1-9. ISSN: 1213- 1539.

[62] MACHA, T.; NAGY, L. The Proposal of an IMS Client. In Proceedings of the 16th Conference Student EEICT 2010 Volume 4. 2010. s. 32-37. ISBN: 978-80-214-4079- 1.

[63] MACHA, T.; NAGY, L.; NOVOTNY, V. QoS Management in OS Windows XP for VoIP IMS Application. In New Information and Multimedia Technologies - NIMT 2010. 2010. s. 4-8. ISBN: 978-80-214-4126- 2.

[64] MACHA, T. Analysis of communication in the implementation of a VoIP connection. Elektrorevue. 2008, n. 12, s. 1-11. ISSN: 1213- 1539.

LIST OF APPENDICES

A Default OSPF testbed

Appendix A contains full routing tables and configuration files of all the routers. Also several graphs showing the process of OSPF routing algorithm mentioned in chapter 3.

Table A.1 Default OSPF routing tables

Router A

Network	Target	Netmask	Gateway	Metric
eth0.3	192.168.7.0	255.255.255.0	192.168.3.2	2
eth0.1	192.168.5.0	255.255.255.0	192.168.1.2	11
eth0.3	192.168.21.0	255.255.255.0	192.168.3.2	3
eth0.3	192.168.20.0	255.255.255.0	192.168.3.2	2
eth0.3	192.168.3.0	255.255.255.0	Directly attached	1
eth0.1	192.168.1.0	255.255.255.0	Directly attached	1
eth0.3	192.168.9.0	255.255.255.0	192.168.3.2	2
eth0.0	10.1.0.0	255.255.0.0	Directly attached	1
eth0.3	10.6.0.0	255.255.0.0	192.168.3.2	3
eth0.3	10.7.0.0	255.255.0.0	192.168.3.2	3
eth0.3	10.5.0.0	255.255.0.0	192.168.3.2	3
eth0.3	10.16.0.0	255.255.0.0	192.168.3.2	3

Router B

Network	Target	Netmask	Gateway	Metric
eth0.1	192.168.7.0	255.255.255.0	192.168.1.1	3
eth0.2	192.168.5.0	255.255.255.0	Directly attached	10
eth0.1	192.168.21.0	255.255.255.0	192.168.1.1	4
eth0.1	192.168.20.0	255.255.255.0	192.168.1.1	3
eth0.1	192.168.3.0	255.255.255.0	192.168.1.1	2
eth0.1	192.168.1.0	255.255.255.0	Directly attached	1
eth0.1	192.168.9.0	255.255.255.0	192.168.1.1	3
eth0.1	10.16.0.0	255.255.0.0	192.168.1.1	4
eth0.1	10.1.0.0	255.255.0.0	192.168.1.1	2
eth0.1	10.6.0.0	255.255.0.0	192.168.1.1	4
eth0.1	10.7.0.0	255.255.0.0	192.168.1.1	4
eth0.1	10.5.0.0	255.255.0.0	192.168.1.1	4

Router C

Network	Target	Netmask	Gateway	Metric
eth0.2	192.168.7.0	255.255.255.0	Directly attached	1
eth0.2	192.168.5.0	255.255.255.0	192.168.7.2	12
eth0.2	192.168.21.0	255.255.255.0	192.168.7.2	3
eth0.2	192.168.20.0	255.255.255.0	192.168.7.2	2
eth0.2	192.168.3.0	255.255.255.0	192.168.7.2	2
eth0.2	192.168.1.0	255.255.255.0	192.168.7.2	3
eth0.2	192.168.9.0	255.255.255.0	192.168.7.2	2
eth0.2	10.1.0.0	255.255.0.0	192.168.7.2	3
eth0.2	10.6.0.0	255.255.0.0	192.168.7.2	3
eth0.2	10.7.0.0	255.255.0.0	192.168.7.2	3
eth0.0	10.5.0.0	255.255.0.0	Directly attached	1
eth0.2	10.16.0.0	255.255.0.0	192.168.7.2	3

Router D

Network	Target	Netmask	Gateway	Metric
eth0.2	192.168.7.0	255.255.255.0	Directly attached	1
eth0.3	192.168.5.0	255.255.255.0	192.168.9.2	11
eth0.3	192.168.21.0	255.255.255.0	192.168.9.2	2

eth0.4	192.168.21.0	255.255.255.0	192.168.20.2	2
eth0.4	192.168.20.0	255.255.255.0	Directly attached	1
eth0.1	192.168.3.0	255.255.255.0	Directly attached	1
eth0.1	192.168.1.0	255.255.255.0	192.168.3.1	2
eth0.3	192.168.9.0	255.255.255.0	Directly attached	1
eth0.1	10.1.0.0	255.255.0.0	192.168.3.1	2
eth0.3	10.6.0.0	255.255.0.0	192.168.9.2	2
eth0.3	10.7.0.0	255.255.0.0	192.168.9.2	2
eth0.2	10.5.0.0	255.255.0.0	192.168.7.1	2
eth0.4	10.16.0.0	255.255.0.0	192.168.20.2	2

Router E

Network	Target	Netmask	Gateway	Metric
eth0.2	192.168.7.0	255.255.255.0	192.168.9.1	2
eth0.1	192.168.5.0	255.255.255.0	Directly attached	10
eth0.3	192.168.21.0	255.255.255.0	Directly attached	1
eth0.2	192.168.20.0	255.255.255.0	192.168.9.1	2
eth0.3	192.168.20.0	255.255.255.0	192.168.21.2	2
eth0.2	192.168.3.0	255.255.255.0	192.168.9.1	2
eth0.2	192.168.1.0	255.255.255.0	192.168.9.1	3
eth0.2	192.168.9.0	255.255.255.0	Directly attached	1
eth0.3	10.16.0.0	255.255.0.0	192.168.21.2	2
eth0.2	10.1.0.0	255.255.0.0	192.168.9.1	3
eth0.4	10.6.0.0	255.255.0.0	Directly attached	1
eth0.0	10.7.0.0	255.255.0.0	Directly attached	1
eth0.2	10.5.0.0	255.255.0.0	192.168.9.1	3

Router I

Network	Target	Netmask	Gateway	Metric
eth0.1	192.168.7.0	255.255.255.0	192.168.20.1	2
eth0.2	192.168.5.0	255.255.255.0	192.168.21.1	11
eth0.2	192.168.21.0	255.255.255.0	Directly attached	1
eth0.1	192.168.20.0	255.255.255.0	Directly attached	1
eth0.1	192.168.3.0	255.255.255.0	192.168.20.1	2
eth0.1	192.168.1.0	255.255.255.0	192.168.20.1	3
eth0.1	192.168.9.0	255.255.255.0	192.168.20.1	2
eth0.2	192.168.9.0	255.255.255.0	192.168.21.1	2
eth0.0	10.16.0.0	255.255.0.0	Directly attached	1
eth0.1	10.1.0.0	255.255.0.0	192.168.20.1	3
eth0.2	10.6.0.0	255.255.0.0	192.168.21.1	2
eth0.2	10.7.0.0	255.255.0.0	192.168.21.1	2
eth0.1	10.5.0.0	255.255.0.0	192.168.20.1	3

ospfd.conf of Router A:

```
hostname RouterA
password zebra
interface eth0.0
 description to_User1
 ip ospf cost 1
interface eth0.1
 description to_RouterB
 ip ospf cost 1
interface eth0.3
 description to_RouterD
 ip ospf cost 1
router ospf
 ospf router-id 0.0.0.1
  network 192.168.1.0/24 area 0
  network 192.168.3.0/24 area 0
  network 10.1.0.0/16 area 0
```

ospfd.conf of Router B:

```
hostname RouterB
password zebra
interface eth0.1
 description to_RouterA
 ip ospf cost 1
interface eth0.2
 description to_RouterE
 ip ospf cost 10
router ospf
 ospf router-id 0.0.0.2
 network 192.168.1.0/24 area 0
 network 192.168.5.0/24 area 0
```

ospfd.conf of Router C:

```
hostname RouterC
password zebra
interface eth0.0
 description to_User2
 ip ospf cost 1
interface eth0.2
 description to_RouterD
 ip ospf cost 1
router ospf
 ospf router-id 0.0.0.3
 network 192.168.7.0/24 area 0
 network 10.5.0.0/16 area 0
```

ospfd.conf of Router D:

```
hostname RouterD
password zebra
interface eth0.1
 description to_RouterA
 ip ospf cost 1
interface eth0.2
 description to_RouterC
 ip ospf cost 1
interface eth0.3
 description to_RouterE
 ip ospf cost 1
interface eth0.4
 description to_RouterI
 ip ospf cost 1
router ospf
 ospf router-id 0.0.0.4
 network 192.168.3.0/24 area 0
 network 192.168.7.0/24 area 0
 network 192.168.9.0/24 area 0
 network 192.168.20.0/24 area 0
```

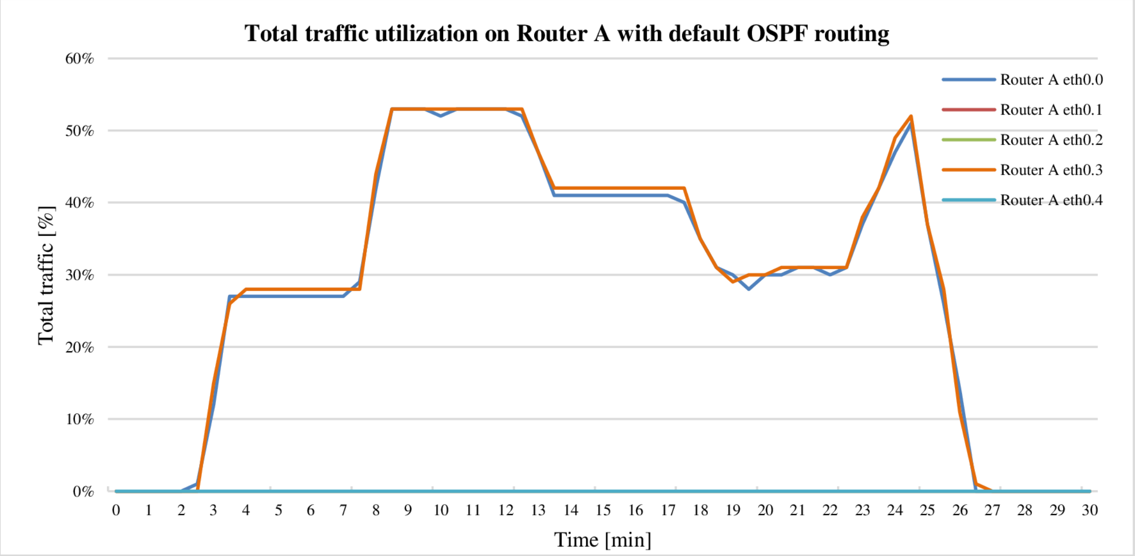
ospfd.conf of Router E:

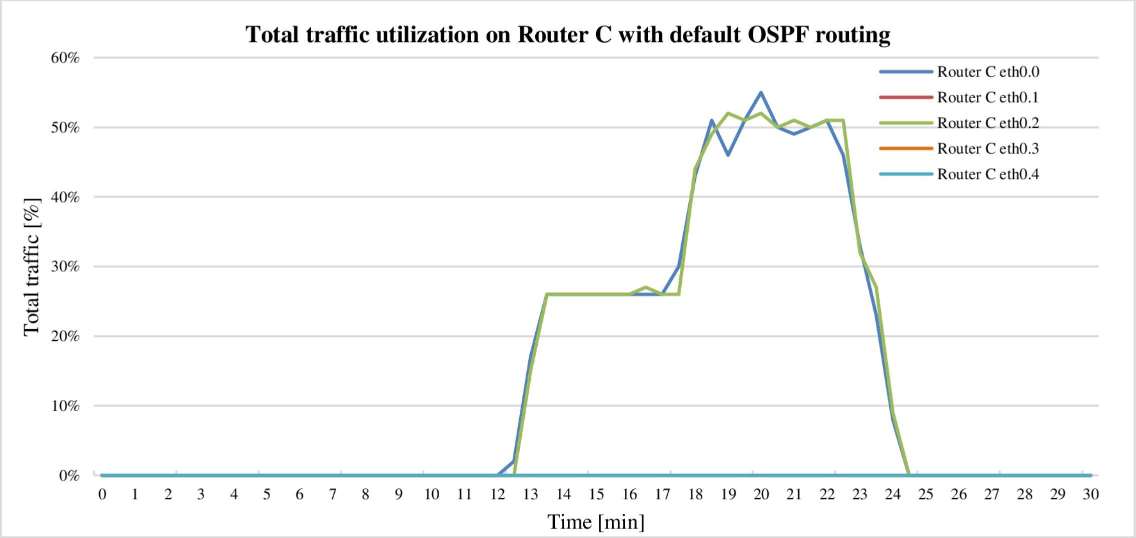
```
hostname RouterE
password zebra
interface eth0.0
 description to_Server1
 ip ospf cost 1
```

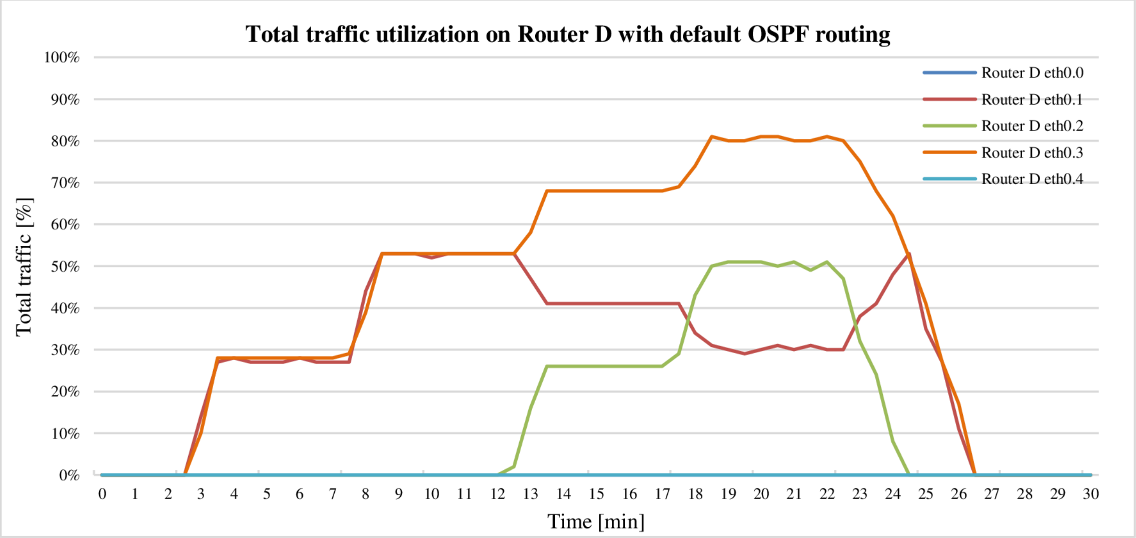
```
interface eth0.1
 description to_RouterB
 ip ospf cost 10
interface eth0.2
 description to_RouterD
 ip ospf cost 1
interface eth0.3
 description to_RouterI
 ip ospf cost 1
interface eth0.4
 description to_Server2
 ip ospf cost 1
router ospf
 ospf router-id 0.0.0.5
 network 192.168.5.0/24 area 0
 network 192.168.9.0/24 area 0
 network 192.168.21.0/24 area 0
 network 10.7.0.0/16 area 0
 network 10.6.0.0/16 area 0
```

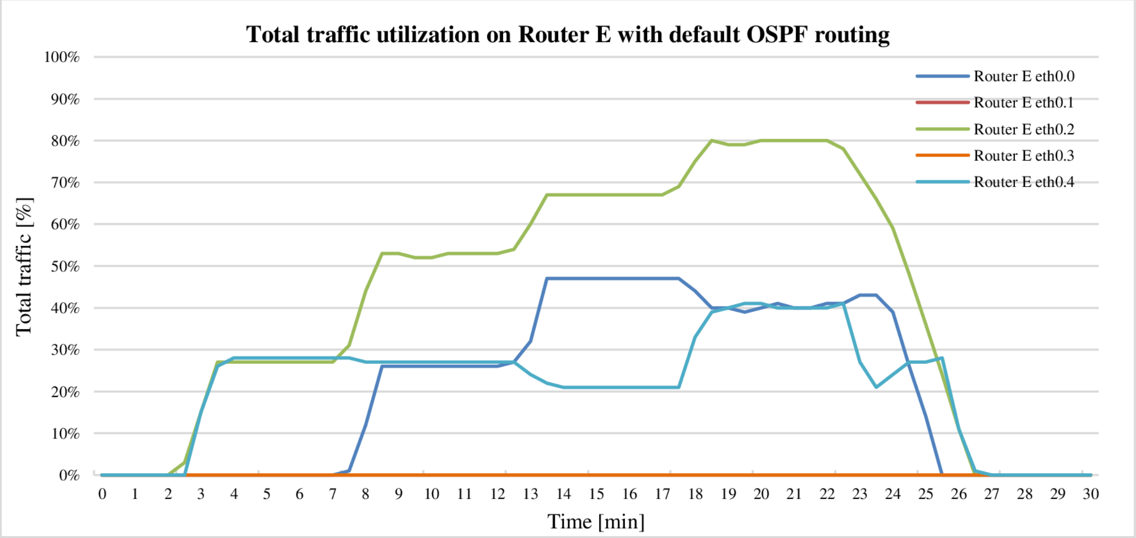
ospfd.conf of Router I:

```
hostname RouterI
password zebra
interface eth0.0
 description to_User3
 ip ospf cost 1
interface eth0.1
 description to_RouterD
 ip ospf cost 1
interface eth0.2
 description to_RouterE
 ip ospf cost 1
router ospf
 ospf router-id 0.0.0.6
 network 192.168.20.0/24 area 0
 network 192.168.21.0/24 area 0
 network 10.16.0.0/16 area 0
```









There is no need to show traffic utilization on Router B and I because there was no traffic.

B DM-SPF testbed

Appendix B contains full routing tables of all the DM-SPF routers. Also several graphs showing the process of DM-SPF routing algorithm tested in chapter 6.

Table B.2 DM-SPF testbed routing tables

Router A

Network	Target	Netmask	Gateway	Metric
eth0.3	192.168.7.0	255.255.255.0	192.168.3.2	2
eth0.1	192.168.5.0	255.255.255.0	192.168.1.2	11
eth0.3	192.168.21.0	255.255.255.0	192.168.3.2	3
eth0.3	192.168.20.0	255.255.255.0	192.168.3.2	2
eth0.3	192.168.3.0	255.255.255.0	Directly attached	1
eth0.1	192.168.1.0	255.255.255.0	Directly attached	1
eth0.3	192.168.9.0	255.255.255.0	192.168.3.2	3
eth0.0	10.1.0.0	255.255.0.0	Directly attached	1
eth0.3	10.6.0.0	255.255.0.0	192.168.3.2	4
eth0.3	10.7.0.0	255.255.0.0	192.168.3.2	4
eth0.3	10.5.0.0	255.255.0.0	192.168.3.2	3
eth0.3	10.16.0.0	255.255.0.0	192.168.3.2	3

Router B

Network	Target	Netmask	Gateway	Metric
eth0.1	192.168.7.0	255.255.255.0	192.168.1.1	3
eth0.2	192.168.5.0	255.255.255.0	Directly attached	10
eth0.1	192.168.21.0	255.255.255.0	192.168.1.1	4
eth0.1	192.168.20.0	255.255.255.0	192.168.1.1	3
eth0.1	192.168.3.0	255.255.255.0	192.168.1.1	2
eth0.1	192.168.1.0	255.255.255.0	Directly attached	1
eth0.1	192.168.9.0	255.255.255.0	192.168.1.1	4
eth0.1	10.16.0.0	255.255.0.0	192.168.1.1	4
eth0.1	10.1.0.0	255.255.0.0	192.168.1.1	2
eth0.1	10.6.0.0	255.255.0.0	192.168.1.1	5
eth0.1	10.7.0.0	255.255.0.0	192.168.1.1	5
eth0.1	10.5.0.0	255.255.0.0	192.168.1.1	4

Router C

Network	Target	Netmask	Gateway	Metric
eth0.2	192.168.7.0	255.255.255.0	Directly attached	1
eth0.2	192.168.5.0	255.255.255.0	192.168.7.2	13
eth0.2	192.168.21.0	255.255.255.0	192.168.7.2	3
eth0.2	192.168.20.0	255.255.255.0	192.168.7.2	2
eth0.2	192.168.3.0	255.255.255.0	192.168.7.2	2
eth0.2	192.168.1.0	255.255.255.0	192.168.7.2	3
eth0.2	192.168.9.0	255.255.255.0	192.168.7.2	3
eth0.2	10.1.0.0	255.255.0.0	192.168.7.2	3
eth0.2	10.6.0.0	255.255.0.0	192.168.7.2	4
eth0.2	10.7.0.0	255.255.0.0	192.168.7.2	4
eth0.0	10.5.0.0	255.255.0.0	Directly attached	1
eth0.2	10.16.0.0	255.255.0.0	192.168.7.2	3

Router D

Network	Target	Netmask	Gateway	Metric
eth0.2	192.168.7.0	255.255.255.0	Directly attached	1
eth0.3	192.168.5.0	255.255.255.0	192.168.9.2	12
eth0.4	192.168.5.0	255.255.255.0	192.168.20.2	12

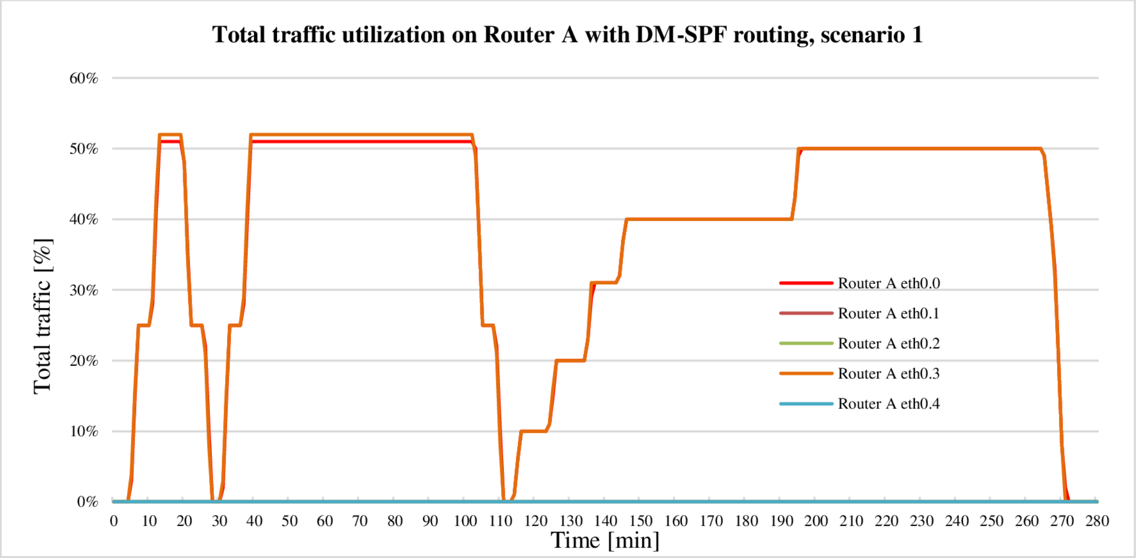
eth0.1	192.168.5.0	255.255.255.0	192.168.3.1	12
eth0.4	192.168.21.0	255.255.255.0	192.168.20.2	2
eth0.4	192.168.20.0	255.255.255.0	Directly attached	1
eth0.1	192.168.3.0	255.255.255.0	Directly attached	1
eth0.1	192.168.1.0	255.255.255.0	192.168.3.1	2
eth0.3	192.168.9.0	255.255.255.0	Directly attached	2
eth0.1	10.1.0.0	255.255.0.0	192.168.3.1	2
eth0.3	10.6.0.0	255.255.0.0	192.168.9.2	3
eth0.4	10.6.0.0	255.255.0.0	192.168.20.2	3
eth0.3	10.7.0.0	255.255.0.0	192.168.9.2	3
eth0.4	10.7.0.0	255.255.0.0	192.168.20.2	3
eth0.2	10.5.0.0	255.255.0.0	192.168.7.1	2
eth0.4	10.16.0.0	255.255.0.0	192.168.20.2	2

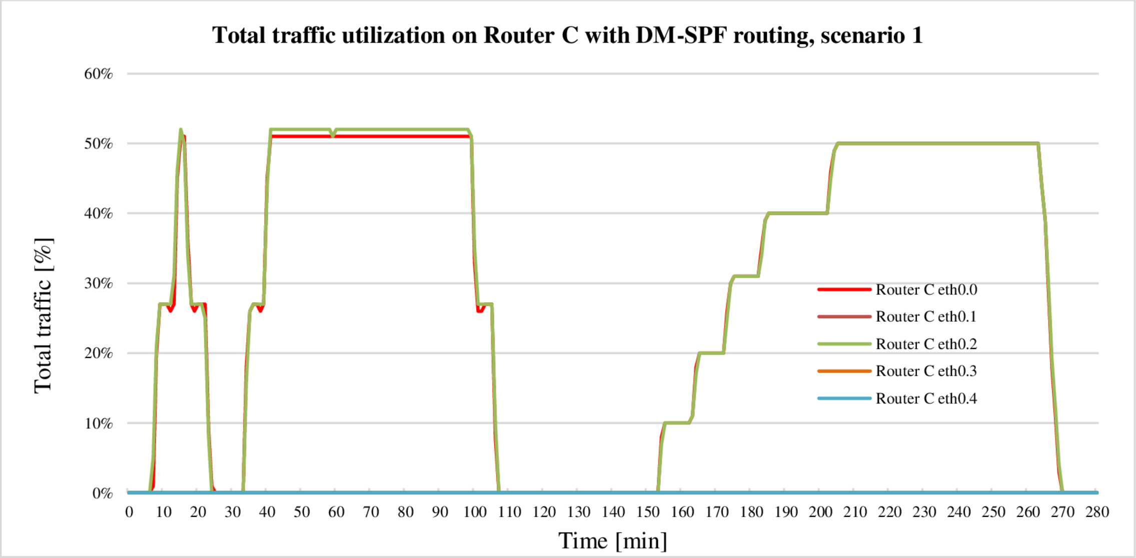
Router E

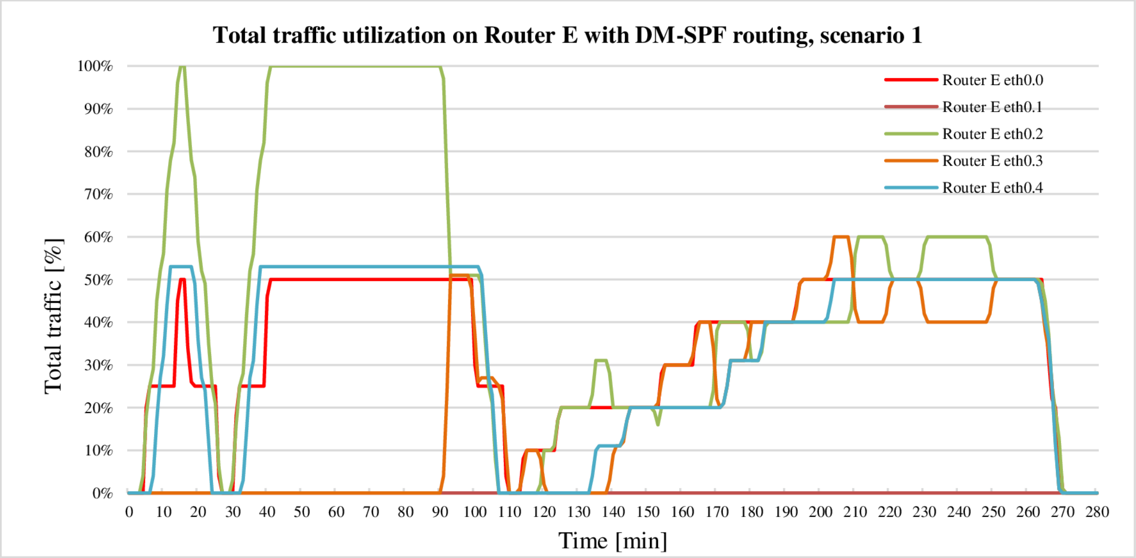
Network	Target	Netmask	Gateway	Metric
eth0.2	192.168.7.0	255.255.255.0	192.168.9.1	3
eth0.3	192.168.7.0	255.255.255.0	192.168.21.2	3
eth0.1	192.168.5.0	255.255.255.0	Directly attached	10
eth0.3	192.168.21.0	255.255.255.0	Directly attached	1
eth0.3	192.168.20.0	255.255.255.0	192.168.21.2	2
eth0.2	192.168.3.0	255.255.255.0	192.168.9.1	3
eth0.3	192.168.3.0	255.255.255.0	192.168.21.2	3
eth0.2	192.168.1.0	255.255.255.0	192.168.9.1	4
eth0.3	192.168.1.0	255.255.255.0	192.168.21.2	4
eth0.2	192.168.9.0	255.255.255.0	Directly attached	1
eth0.3	10.16.0.0	255.255.0.0	192.168.21.2	2
eth0.2	10.1.0.0	255.255.0.0	192.168.9.1	4
eth0.3	10.1.0.0	255.255.0.0	192.168.21.2	4
eth0.4	10.6.0.0	255.255.0.0	Directly attached	1
eth0.0	10.7.0.0	255.255.0.0	Directly attached	1
eth0.2	10.5.0.0	255.255.0.0	192.168.9.1	4
eth0.3	10.5.0.0	255.255.0.0	192.168.21.2	4

Router I

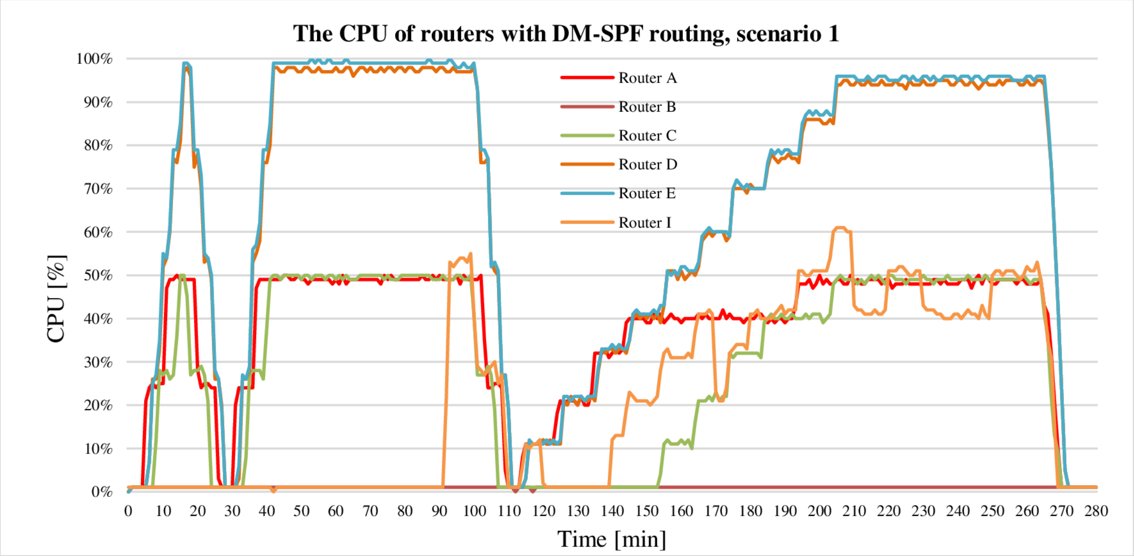
Network	Target	Netmask	Gateway	Metric
eth0.1	192.168.7.0	255.255.255.0	192.168.20.1	2
eth0.2	192.168.5.0	255.255.255.0	192.168.21.1	11
eth0.2	192.168.21.0	255.255.255.0	Directly attached	1
eth0.1	192.168.20.0	255.255.255.0	Directly attached	1
eth0.1	192.168.3.0	255.255.255.0	192.168.20.1	2
eth0.1	192.168.1.0	255.255.255.0	192.168.20.1	3
eth0.1	192.168.9.0	255.255.255.0	192.168.20.1	3
eth0.2	192.168.9.0	255.255.255.0	192.168.21.1	3
eth0.0	10.16.0.0	255.255.0.0	Directly attached	1
eth0.1	10.1.0.0	255.255.0.0	192.168.20.1	3
eth0.2	10.6.0.0	255.255.0.0	192.168.21.1	2
eth0.2	10.7.0.0	255.255.0.0	192.168.21.1	2
eth0.1	10.5.0.0	255.255.0.0	192.168.20.1	3



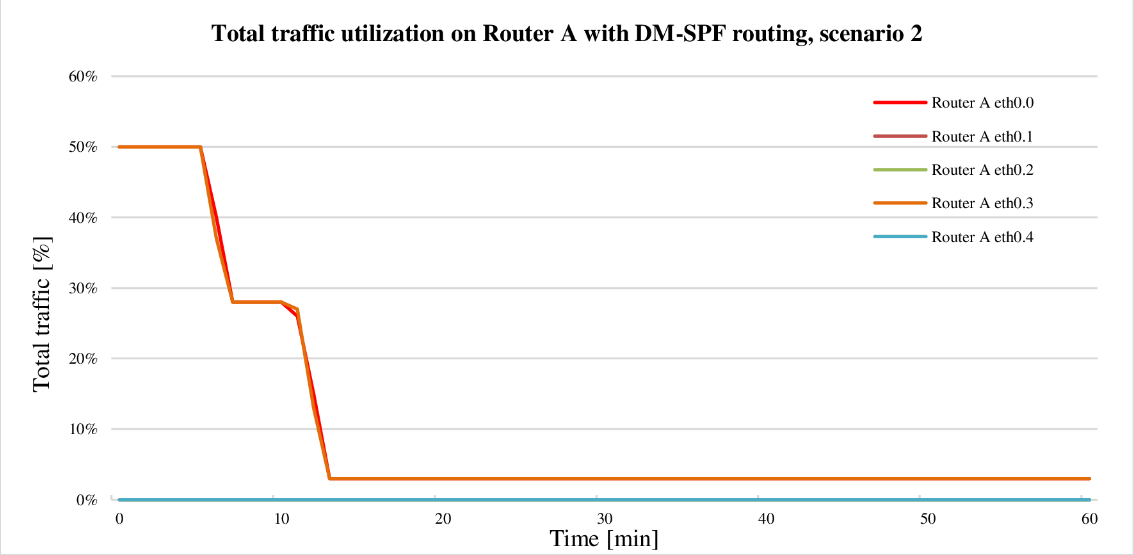


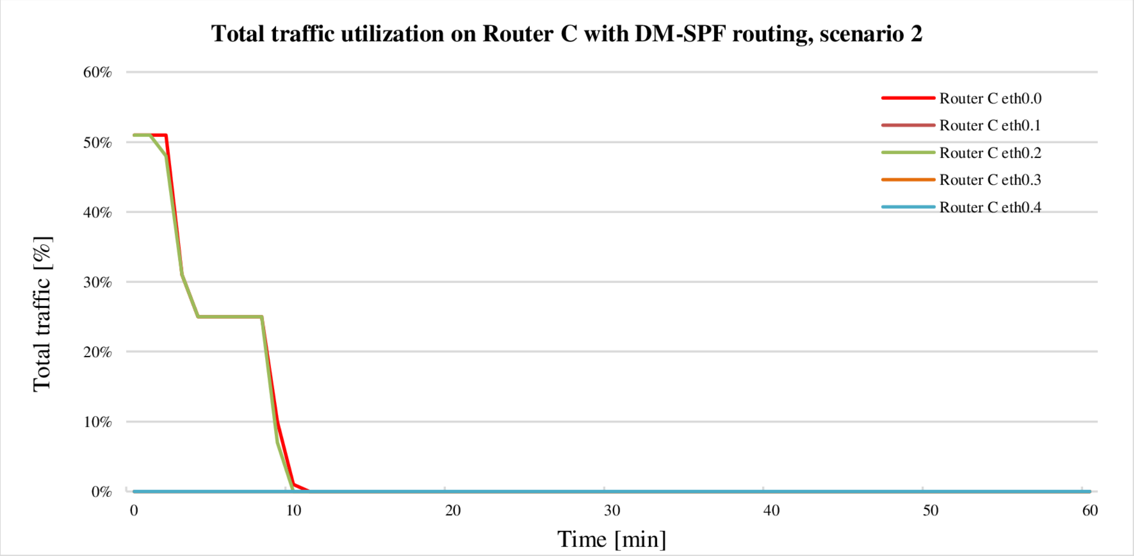


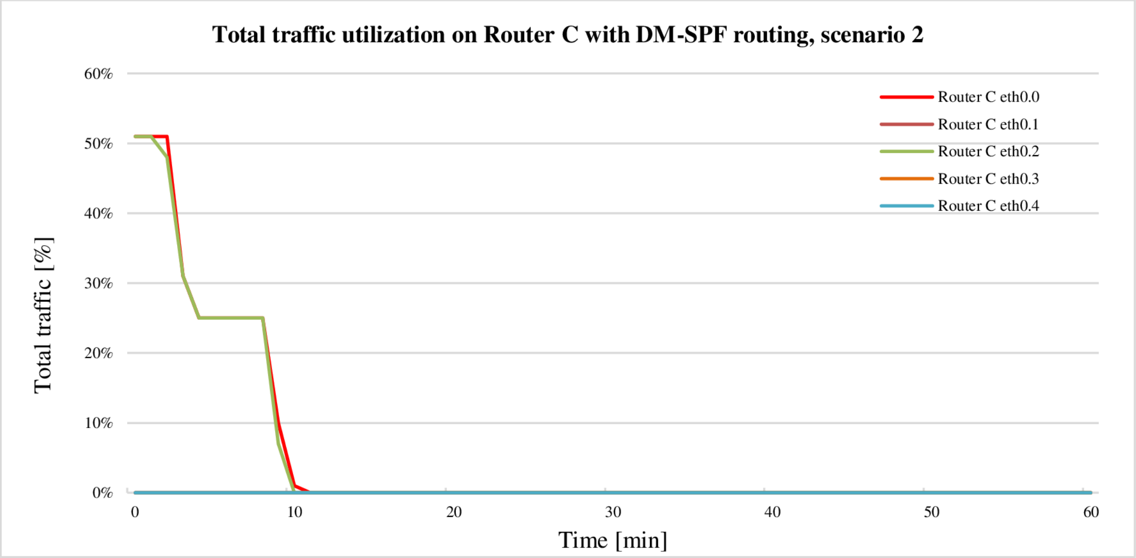
There is no need to show traffic utilization on Router B because there was no traffic.


The CPU of routers with DM-SPF routing, scenario 1
CPU [%]
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280
Time [min]
Router A
Router B
Router C
Router D
Router E
Router I

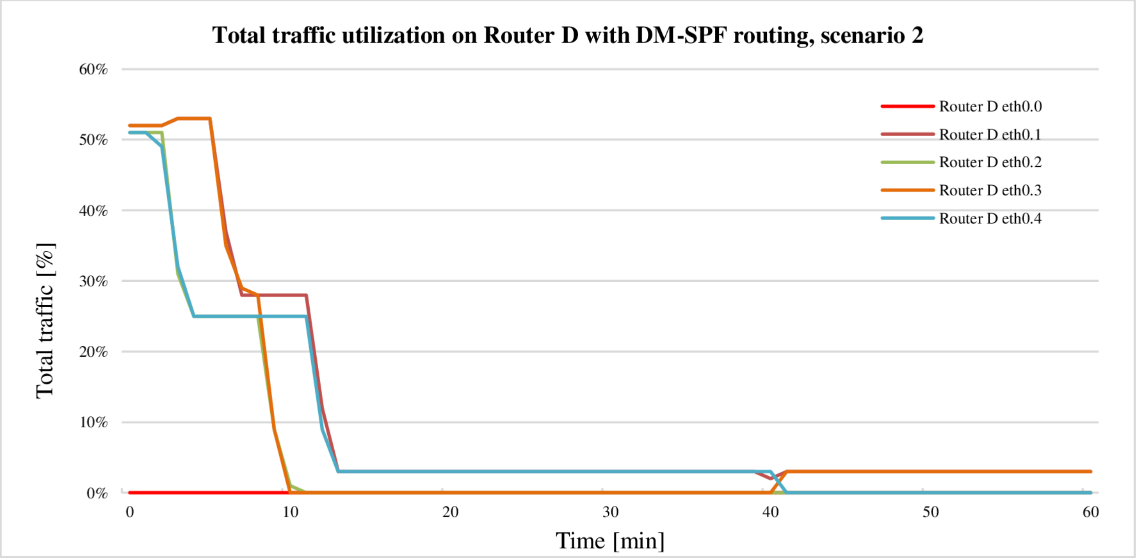
Scenario 2


Total traffic utilization on Router A with DM-SPF routing, scenario 2
Total traffic [%]
60%
50%
40%
30%
20%
10%
0%
0 10 20 30 40 50 60
Time [min]
Router A eth0.0
Router A eth0.1
Router A eth0.2
Router A eth0.3
Router A eth0.4

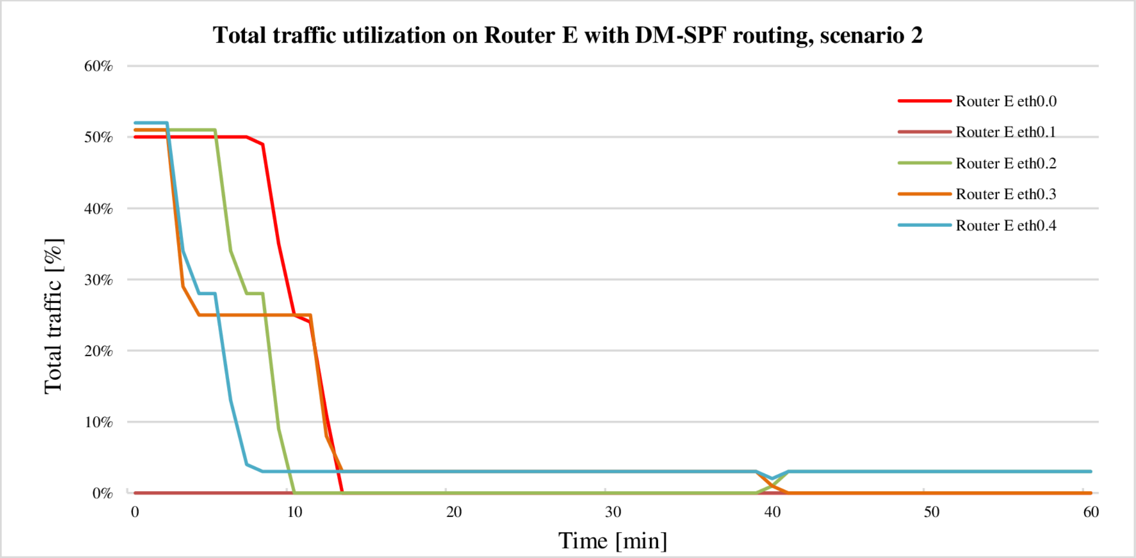

Total traffic utilization on Router C with DM-SPF routing, scenario 2
Total traffic [%]
60%
50%
40%
30%
20%
10%
0%
0 10 20 30 40 50 60
Time [min]
Router C eth0.0
Router C eth0.1
Router C eth0.2
Router C eth0.3
Router C eth0.4



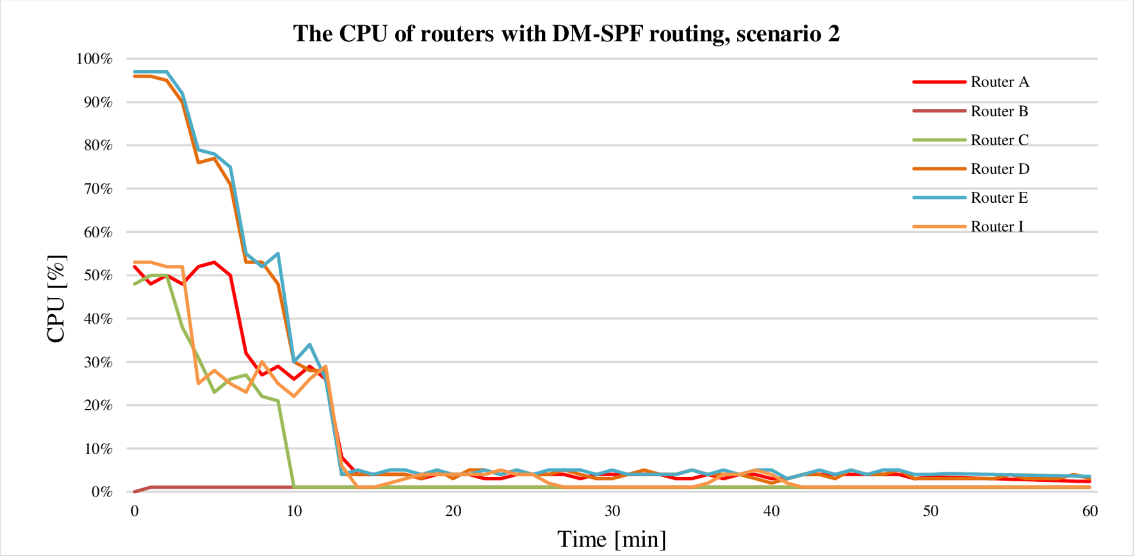
Total traffic utilization on Router C with DM-SPF routing, scenario 2



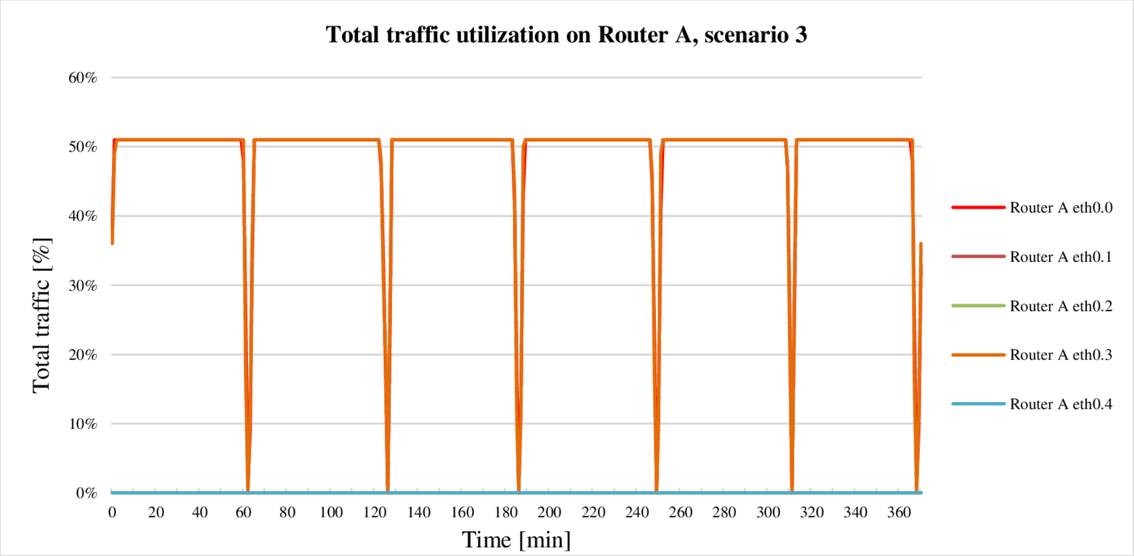
Total traffic utilization on Router D with DM-SPF routing, scenario 2

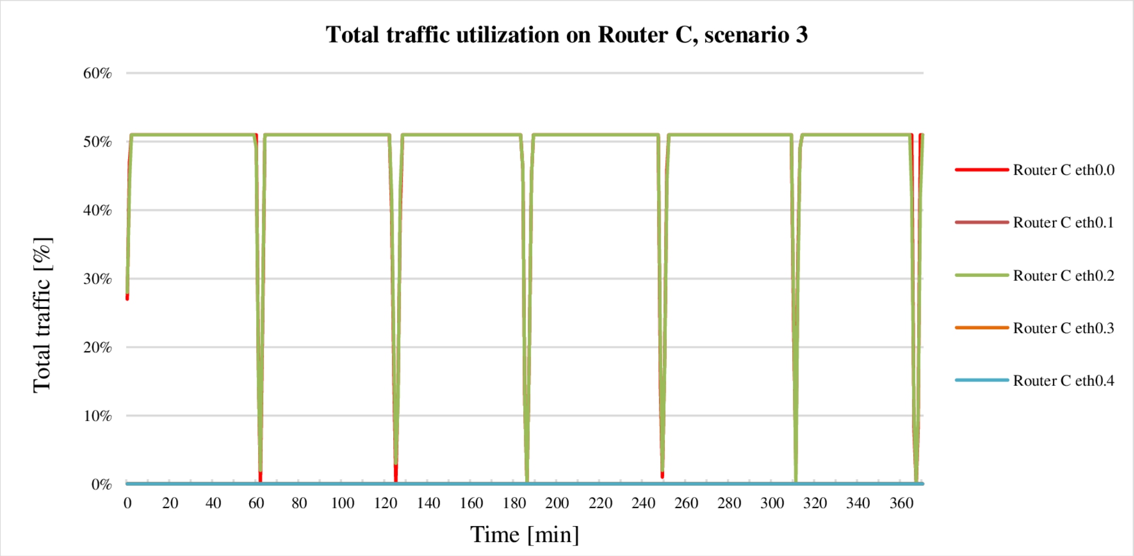


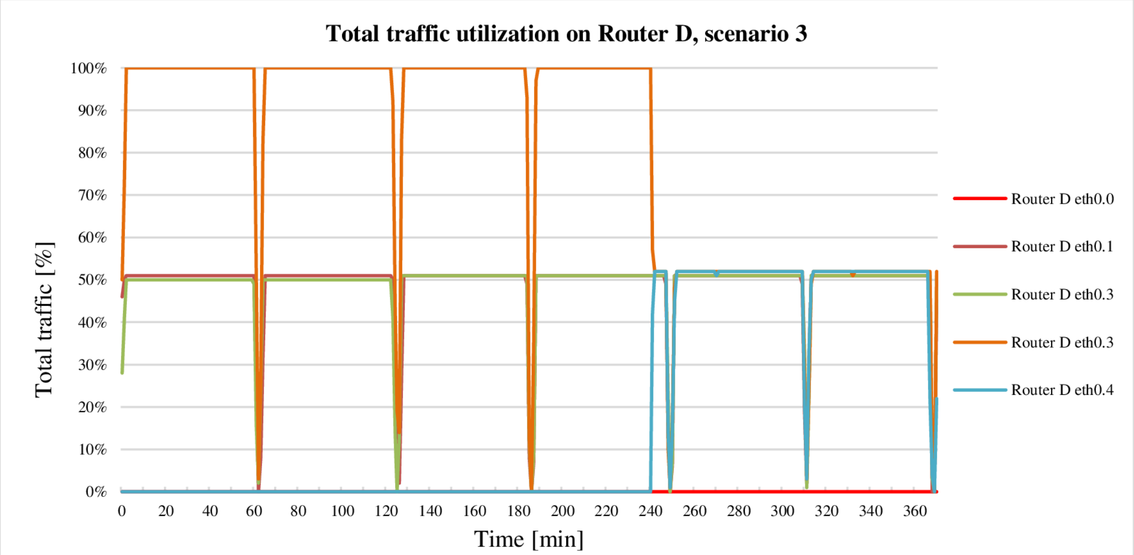
Total traffic utilization on Router E with DM-SPF routing, scenario 2

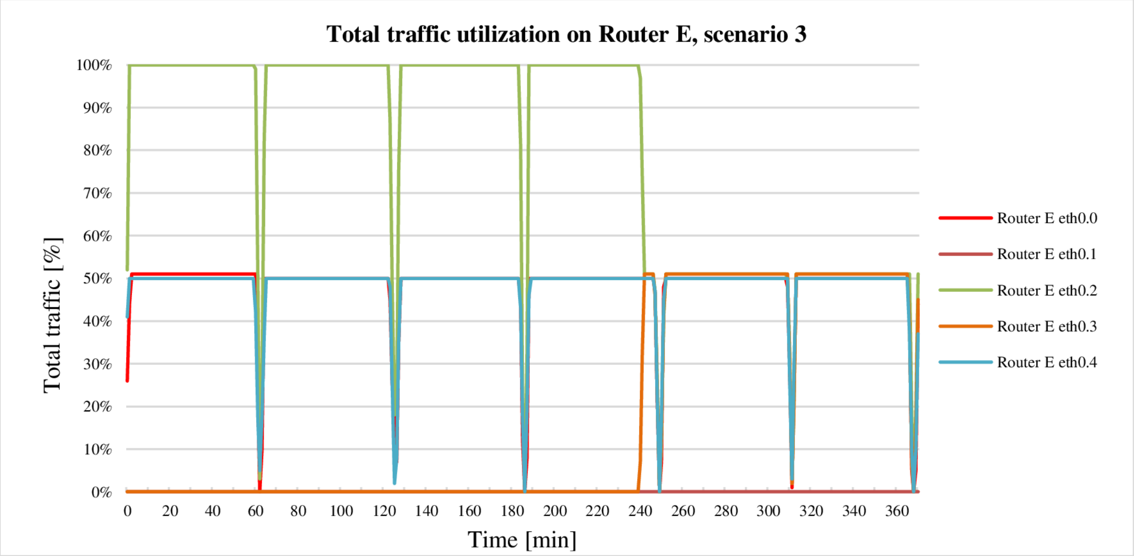

The CPU of routers with DM-SPF routing, scenario 2
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
CPU [%]
0
10
20
30
40
50
60
Time [min]
Router A
Router B
Router C
Router D
Router E
Router I

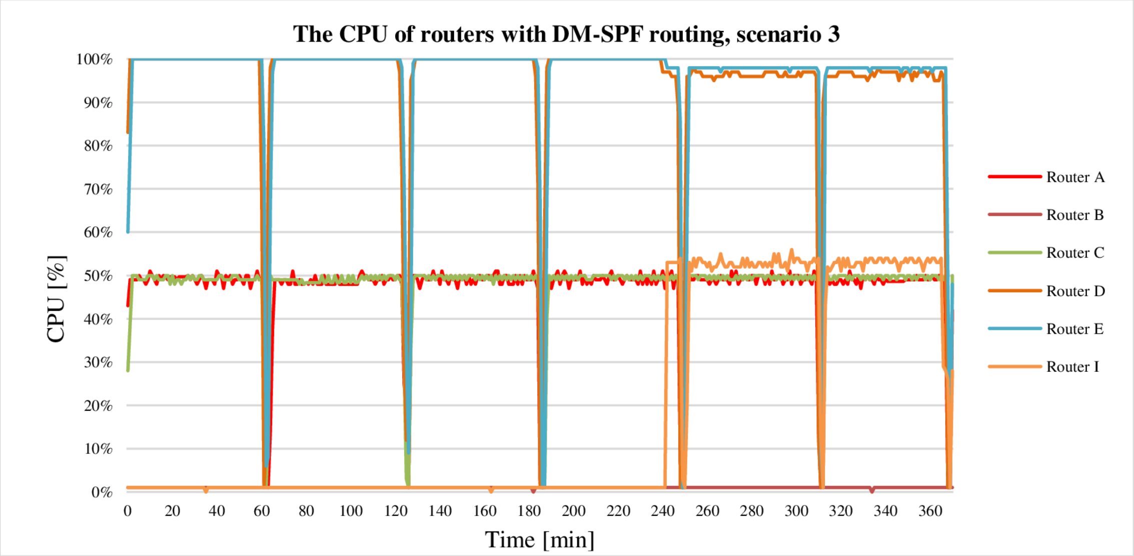
Scenario 3


Total traffic utilization on Router A, scenario 3
60%
50%
40%
30%
20%
10%
0%
Total traffic [%]
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
Time [min]
Router A eth0.0
Router A eth0.1
Router A eth0.2
Router A eth0.3
Router A eth0.4


Total traffic utilization on Router C, scenario 3
60%
50%
40%
30%
20%
10%
0%
Total traffic [%]
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
Time [min]
Router C eth0.0
Router C eth0.1
Router C eth0.2
Router C eth0.3
Router C eth0.4


Total traffic utilization on Router D, scenario 3
Total traffic [%]
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360
Time [min]
Router D eth0.0
Router D eth0.1
Router D eth0.3
Router D eth0.3
Router D eth0.4


Total traffic utilization on Router E, scenario 3
Total traffic [%]
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360
Time [min]
Router E eth0.0
Router E eth0.1
Router E eth0.2
Router E eth0.3
Router E eth0.4


The CPU of routers with DM-SPF routing, scenario 3
CPU [%]
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360
Time [min]
Router A
Router B
Router C
Router D
Router E
Router I

CURRICULUM VITAE

Name:	Tomáš Mácha
Date of birth:	29.7.1984
Address:	Na Rybníčku 421, Krmelín, 739 24
Tel.:	+420 777 094 295
E-mail:	tomas.macha@phd.feec.vutbr.cz

Education

2008	**Brno University of Technology** Faculty of Electrical Engineering and Communication Teleinformatics *(Doctoral Study Programme)*
2006-2008	**Brno University of Technology** Faculty of Electrical Engineering and Communication Communications and Informatics *(Master Study Programme)* Thesis title: Converged solutions of speech services
2003-2006	**Brno University of Technology** Faculty of Electrical Engineering and Communication Teleinformatics *(Bachelor Study Programme)* Thesis title: Audio services over IP networks
1999-2003	**Grammar school Ostrava-Hrabůvka**

International study experience

2011	**Molde University College**, Norway – one semester Faculty of Informatics
2010	**Molde University College**, Norway – one semester Faculty of Informatics
2007	**Aalborg University**, Denmark – one semester Faculty of Engineering, Science and Medicine Mobile Communication Project: Connectivity in a sensor network

Participation in Projects

FEKT-S-11-15: Research of electronic communication systems

FRVS 2986/2010/G1: Introduction to tutorial support of multimedia services in a converged environment of mobile and wireless networks

FRVS 2954/2010/F1a: Tutorial support of multimedia services in IP networks

FEKT-S-10-16: Research of communication systems and networks

MSM21630513: Electronic communication systems and technologies of new generation (ELKOM)
FRVS 1589/2008/F1a: Innovation of education process of modern mobile and wireless technologies

Publications and Products

International journals with impact factor: 1
Proceedings of international conferences: 19
Other journals: 9
Product: 1

Awards and Certificates

First place in EEICT student competition (In Proceedings of the 17th Conference Student EEICT 2011) in doctoral program

Third place in EEICT student competition (In Proceedings of the 15th Conference Student EEICT 2009) in doctoral program

CCNA Routing and Switching (2013)

CCNA Voice (2013)

Work experience

Since 2012	**Tieto Czech s.r.o.** Network Specialist, VoIP specialist, Problem manager
2008-2012	**Brno University of Technology** Educational activity: Exercise leading: Communication Means of Mobile networks, Subscriber Terminal Equipment, Network Architecture Supervisor: Leadership of defended diploma and bachelor theses

Folders in Enclosed CD