BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

COMPARISON OF LINK-STATE ROUTING PROTOCOLS
POROVNÁNÍ LINK-STATE SMĚROVACÍCH PROTOKOLŮ

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR   **JAN ZAVŘEL**
AUTOR PRÁCE

SUPERVISOR   **Ing. VLADIMÍR VESELÝ, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2019

Ústav informačních systémů (UIFS)   Akademický rok 2018/2019

Zadání bakalářské práce

21559

Student: **Zavřel Jan**

Program: Informační technologie

Název: **Porovnání link-state směrovacích protokolů**

Comparison of Link-State Routing Protocols

Kategorie: Počítačové sítě

Zadání:

1. Analyzujte směrovací protokoly pracující na principu link-state, konkrétně protokoly OSPF a IS-IS.
2. Zjistěte aktuální stav a nasazení těchto protokolů v simulátoru OMNeT++ a na reálných zařízeních.
3. Dle doporučení vedoucího navrhněte simulační scénáře pro porovnávání těchto protokolů mezi sebou, zaměřte se např. na rychlost konvergence, množství vyměněných zpráv (a jejich pořadí a velikosti).
4. Proveďte měření a porovnání, diskutujte a analyzujte dosažené výsledky.

Literatura:

- MOY, J. OSPF version 2. STD 54. *RFC 2178*, 1998.
- ORAN, D. OSI ISIS Intradomain routing protocol. *RFC 1142*, 1990.
- COCUĽA, M. *Porovnání distance-vector směrovacích protokolů*. Brno, 2017. Dostupné z: http://www.fit.vutbr.cz/study/DP/BP.php?id=19977. Bakalářská práce

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Veselý Vladimír, Ing., Ph.D.**

Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 15. května 2019

Datum schválení: 30. října 2018

Abstract

This thesis deals with a comparison of two dynamic link-state routing protocols OSPF and IS-IS. These protocols are used to route within one autonomous system under control of one administrative entity. They are similar in theory but IS-IS seems to be much more popular in large topologies in practice. Goal of this thesis is to figure out how their properties and features manifest on given topologies on real routers and in simulations. Simulations are realized in discrete simulator OMNeT++ and real topologies are built with Cisco devices. Primary measured parameters are speed of convergence, bandwidth usage and message order. The simulation result may be affected by their implementation in INET and AN-SAINET frameworks. Implementation properties of OSPF and IS-IS on Cisco devices are compared to their simulated counterparts in order to verify functionality of the simulation models.

Abstrakt

Tato práce se zabývá porovnáním dynamických link-state směrovacích protokolů OSPF a IS-IS. Tyto protokoly se používají při směrování v jednom autonomním systému pod jednou správou. Oba jsou v teorii podobné, avšak v praxi se ve velkých sítích zdá preferovaný IS-IS. Cílem práce je zjistit, jak se vlastnosti protokolů projeví na testovaných topologiích na reálných síťových prvcích a v simulacích. Simulace jsou uskutečněny v diskrétním simulátoru OMNeT++ a reálné topologie jsou sestaveny na Cisco směrovačích. Hlavními sledovanými parametry jsou rychlost konvergence a množství přenesených dat. Výsledky ze simulací jsou porovnány s výsledky naměřenými na Cisco zařízeních. Cílem práce je také prověřit funkčí vlastnosti těchto modelů.

Keywords

Comparison, link-state, OSPF, IS-IS, configuration, routing, OMNeT, INET

Klíčová slova

Porovnání, link-state, OSPF, IS-IS, konfigurace, směrování, OMNeT, INET

Reference

Rozšířený abstrakt

S narůstající velikostí internetu vzniká požadavek na používání co nejefektivnějších protokolů. Mezi populární směrovací protokoly patří OSPF a IS-IS. Oba tyto protokoly jsou navrženy ke směrování v jednom autonomním systému pod jednou správní entitou. Cílem této práce je objektivně porovnat oba protokoly jak na úrovni teoretické, tak na úrovni praktické a poodhalit, jak se rozdílné vlastnosti protokolů projevují na reálných zařízeních a v simulacích na příslušných modelech. Praktické porovnání probíhá v diskrétním simulátoru OMNeT++ a na Cisco zařízeních. Hlavními parametry pro praktické porovnání je rychlost konvergence a množství přenesených dat jak během synchronizace, tak v klidovém stavu.

I když se v rámci směrování mohou tyto protokoly zdát velice podobné, tak při bližším pohledu jsou jejich rozdílnosti velice markantní. Základní principy obou protokolů jsou uvedeny v teoretické kapitole. Text v této kapitole shrnuje obecné vlastnoti protokolů a také uvádí v jakých standartech jsou definovány. Kromě popisu použitých komunikačních struktur se kapitola také zabývá celkovým pohledem na proces navazování sousedctví a na rozdílosti v chování na point-to-point a broadcastových linkách. Sekce o protokolech také obsahují shrnutí stavu jejich modelu v simulátoru OMNeT++.

Konfigurační kapitola se zabývá zprovozněním protokolů na jednoduché topologii. V topologii jsou použitá Cisco zařízení a tato sekce obsahuje hlavně příkazy pro samotné zprovoznění protokolů a pro upravení základních parametrů. Jsou zde také uvedeny Cisco konfigurační manuály, které pokrývají všechny příkazy pro daný protokol. Samotná topologie je také popsána pomocí obrázků, ve kterých je naznačena i rozdílná filozofie při vytváření oblastí.

Cisco je aktuálně největší výrobce síťových prvků a právě jejich výrobky se používají v páteřních sítích. Na těchto zařízeních se předpokládá perfektní implementace měřených protokolů. Tyto implementace také mohou obsahovat některé vedlejší funkce, které mohou potencionálně měření ovlivnit.

OMNeT++ je rozšiřitelné, open source simulační prostředí používáné převážně pro simulaci sítí. Právě rozšiřitelnost simulátoru o moduly umožnila vznik několika desítek aplikačních rámců, které simulaci rozšiřují o uživately vytvořené modely. Praktická kapitola v této práci využívá rámec INET a jeho rozšíření ANSAINET. Samotný INET obsahuje například implementaci základních TCP/IP protokolů jako TPC, UDP, IPv4, IPv6 a nebo také implementaci některých směrovacích protokolů. Mezi tyto patří například OSPFv2 či BGP. Rozšíření ANSAINET obsahuje modely protokolů mezi které patří například IS-IS, HSRP, STP či OSPFv3. Tento projekt je provozován zaměstnanci a studenty Fakuly Informačních Technologií Vysokého Učení Technické v Brně. Oba rámce mají plně otevřený zdrojový kód a oba měřené protokoly jsou s každou verzí těchto rámců zdokonalovány. Oba protokoly jsou testovány v jejich posledních implementacích, které jsou maximálně několik týdnů staré. U tohoto měření jsou použity operační systémy Windows 10 a Ubuntu 18.04.

Samotné testování probíhá nejdříve na Cisco prvcích a následně v OMNeTu. Oba protokoly jsou měřeny na 2 topologiích a každá topologie obsahuje 2 scénáře, které topologii ovlivňují. Topologie a scénáře jsou identické pro Cisco i OMNeT. Toto vede k vyprodukování porovnatelných hodnot. Hlavními měřenými hodnotami jsou časy výměn informací o topologii, časy zavedení informací do směrovacích tabulek a množství vyměněných dat. Obě měřené sekce obsahují celkové shrnutí naměřených hodnot a porovnání OSPF a IS-IS implementací na dané platformě. V části o OMNeTu jsou také uvedeny pozorované chyby v komunikaci obou protokolů a chyby při práci s virtuálními směrovači.

V rámci měření v OMNeTu jsou také popsány tzv. 'Fingerprints'. Jedná se o hashe, které se počítají během simulace z událostí a zpráv a které po jejim dokončení mohou ověřit, zda se simulace změnila oproti například referenční simulaci. Toho se velice hodí například během refaktorizace kódu nebo při opravování drobných chyb. Nutno zmínit, že hashe ze stejné simulace se na procesorech Intel a AMD mohou lišit. Toto je způsobeno rozdílem počítání s plovoucí řádovou čárkou.

V záverečné části práce je cílem porovnat naměřené hodnoty na obou platformách mezi sebou. OSPFv2, který je implementován v rámci INETu, a IS-IS, implementovaný v rámci rozšíření ANSAINET, jsou zde porovnány s jejich implentacemi na Cisco zařízeních. Kapitola také obsahuje výčet nejdůležitějších rozdílů a případných chyb a celkově shrnuje měření.

Comparison of Link-State Routing Protocols

Declaration

Hereby I declare that this bachelor's thesis was prepared as an original author's work under the supervision of Ing. Vladimír Veselý Ph.D. The supplementary information was provided by Ing. Marcel Marek and Bc. Lukáš Galbička. All the relevant information sources, which were used during preparation of this thesis, are properly cited and included in the list of references.

<div align="right">

. . . .

Jan Zavřel

May 15, 2019

</div>

Acknowledgements

I would like to thank my family for providing me with excellent study environment and also my supervisor Ing. Vladimír Veselý, Ph.D. for guiding me along the way of writing this thesis. I would also like to thank Ing. Marcel Marek for his help during some simulation difficulties and Bc. Lukáš Galbička for providing some insights and a new perspective.

Contents

Chapter 1

Introduction

Usage of dynamic routing protocols is absolutely crucial when building a large network. Main goal of these protocols is to route data down the optimal path to the destination when multiple paths are available. Routers exchange information about changes in the topology and update their routing tables accordingly. Using only static routes would render route management totally impossible in larger networks. Main parameters of routing protocols are scalability, speed of convergence, security and efficiency.

This thesis is focused on comparing two very popular dynamic routing protocols OSPF and IS-IS. Both protocols are used for routing under single administrative entity in one autonomous system. Main asset of this thesis is the comparison of differences and similarities between the two and impact of these differences on different topologies. These topologies are tested in a simulation and on Cisco routers. Both protocols are similar in design as they for example use the same Dijkstra's algorithm to calculate best path to the destination, both support authentication and division of topology into areas. There is a number of less or more significant differences between the two. OSPF works on layer 3 and natively supports Internet Protocol (IP) and IS-IS operates on layer 2 of ISO/OSI model and had to be extended in order to support IP.

Brief theoretical overview of both protocols is included in the second chapter. Short sections provide brief description of each protocol, describe the way it discovers neighbors and explain usage of different messages in the communication.

Third chapter establishes simple topology and showcases basic configuration on Cisco devices for both protocols.

Topologies are built on Cisco devices, properly configured, analyzed and measured. Results of there measurements also serve as referential outcome.

Simulations are realized in discreet event simulator OMNeT++ which implements both OSPF and IS-IS. OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily used for realizing network simulations. INET framework for OMNeT contains models for the Internet stack and many other protocols and components. Used extension of this framework is called ANSAINET and is developed by employees and students at Brno University of Technology at Faculty of Information Technology.

Chapter 2

Theory

2.1 OSPF

2.1.1 Introduction

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP)[1] used to distribute routing information to all routers in Autonomous System in TCP/IP Networks. It is based on link-state technology and dynamically adapts to network changes. Each router is collecting information about the whole topology and Dijkstra's algorithm is used for finding the shortest path to the destination. Each link has its own cost (metric) which is derived from the speed of the link. If more routes with equal cost exist, traffic is equally distributed. OSPF is designed for TCP/IP, is classless and responds quickly to network changes yet generates a small amount of traffic. Packet authentication is available via a simple password or MD-5. Authentication type can differ in each area.

OSPF exists in two versions: OSPFv2 is standardized in RFC 2328 [14] for IPv4 networks and OSPFv3 is standardized in RFC 5340 for IPv6 [9]. Support of multiple address families in OSPFv3 is standartized in RFC 5838 [12]. This chapter is based on information provided in those documents and also in [15], [3], [17].

2.1.2 Overview

OSPF clusters networks into areas. Each area is identified by unique ID and divides routers into groups in order to control traffic. Also topology of an area is hidden to the rest of the autonomous system (AS). Routers in any area type with the same Area ID will always form adjacencies. Individual areas are connected via Area Border Router (ABR) to the backbone. Each interface of a router can belong to a different area. Routers connecting one AS to another are called Autonomous System Boundary Routers (ASBR). Routes learned from Exterior Gateway Protocol (EGP)[2] are distributed to the whole AS. OSPF defines area types as standard, backbone and stub. Cisco devices support extensions for stub area. Area types differ in policy of distributing internal and external routes.

Designated Router and Backup Designated Router

Every multi-access segment must have its Designated Router (DR) and Backup Designated Router (BDR). This solution greatly limits the amount of traffic OSPF generates and thus

---

[1]IGP - Used to exchange routing information within one autonomous system.
[2]EGP - used to exchange data between Autonomous Systems.

saves bandwidth. In order to do this, two routers are elected to establish adjacency with everyone else on the segment. Once the database is complete, routers will share their Link-State Database (LSDB) with everyone else. This reduces the number of exchanges from $O(n * n)$ to $O(n)$. BDR is elected in case of failure on DR. Election is done by interface priority `0 - 255`. Router with the highest interface priority becomes DR and router with second highest becomes BDR. If DR or BDR cannot be elected because priority is not set or is the same, Router-ID is used instead. Once DR and BDR are elected, they keep their role until failure and even if router with higher priority is connected to the segment.

2.1.3 Packet types

Both OSPFv2 and OSPFv3 are using multicast to communicate with each other.

`224.0.0.5` or `FF02::5` to address other OSPF routers on segment.
`224.0.0.6` or `FF02::6` to address DR and BDR router on segment.

Hello packet

Hello packet is used to discover other OSPF enabled routers and to establish neighborhood with them. This means that routers will agree on communication parameters. They are also used to elect DR and BDR on multi-access segments. These packets are sent periodically. This is called **Hello interval**. **Dead Interval** is four times larger than Hello interval. If no Hello packets are received in this interval, the neighborhood is considered down. Implicit Cisco Hello interval on point-to-point links and on broadcast is **10 seconds**, so dead interval is **40 seconds**. These timer values must match between neighboring routers. Routers can only exchange a routing information if a adjacency is formed.

Database Description packet

This type of packet is used to describe the LSDB. This database describes a directed graph with itself as a root. Each link is identified by its Link-state ID (LID). These packets are exchanged between two routers when an adjacency is formed. Router with higher `Router ID` is selected as a master, the other one is a slave. Master begins communication, and multiple packets can be sent. Database Description packet (DDP or DBD) contains a set of LSAs. If a received DBD contains new information, then router generates Link-State Request (LSR).

Link-State packets

LSR contains LID of the desired Link-state Advertisement (LSA). Response to the LSR is a Link-State Update (LSU) which contains one or more desired LSAs. Link state Acknowledgement (LSAck) is used to acknowledge correction of received LSAs.

2.1.4 OSPF Neighbor Process

OSPF neighbor process can be decomposed into 8 general states.

State 1 - Down State

Router is not receiving any OSPF Hello packets but it is still trying to discover neighbors on the link by sending its own hello packets. When Hello packet is received, then OSPF enters Init State.

State 2 - Init State

- Received Hello packet does not contain receivers router ID (RID)

If received packet does not contain receivers RID, then router should contain sender's RID in next Hello packet.

State 3 - Attempt

The Attempt state is only valid for Non-Broadcast Multi-Acess (NBMA) networks. Because Hello packet has not been received, router will send unicast Hello packet to the configured IP address.

State 4 - Two-Way State

- Received Hello packet contains receivers RID

If parameters inside received Hello packet are as expected, then neighborhood is established. There are several parameters that must be the same between the neighbors.

- Authentication

- Hello and Dead Interval

- Area number

- Different Router-ID

- Network

DR and BDR are also elected in this state.

State 5 - ExStart State

Routers exchange empty DBD packets and router with higher RID is elected as a master. These DBD packets are numbered by sequence number. Master is increasing this number while slave is repeating received number.

State 6 - Exchange State

If master and slave relationship is established, then routers start to exchange DBD packets. Each router filters and collects missing LSAs in a LSR list. Missing LSAs are requested by LSRs in the next state.

State 7 - Loading State

Finally routing information is exchanged in this state. LSAs received in Exchange State are requested with LSR, received LSUs are then used to update the LSDB. LSAcks are used to acknowledge correction of the received information.

State 8 - Full State

Neighboring databases are synchronized. Router will remain in this state forever or until failure. If any change in the network occurs, then routers will go through this process again. Router still sends Hello packets and floods changes to the neighbors. On multiaccess segments routers will only achieve this state with DR and BDR as they do not share routing information directly.

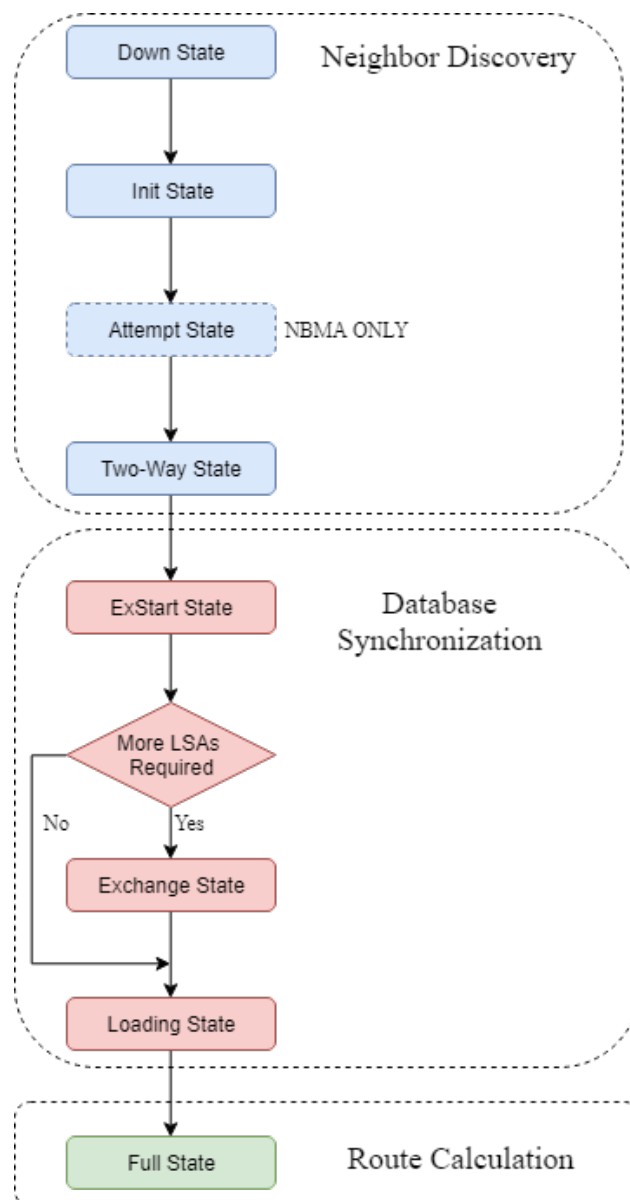### 2.1.5 Diagram of OSPF process

Figure 2.1: OSPF State Diagram

2.1.6 OSPF in OMNeT++

Model of OSPFv2 is a part of the INET framework and is considered fully functional. This model was created by *Andras Babos* and *Andras Varga* in 2006 and is updated in every new version of INET. *Model Catolog* also states that more documentation and testing of this model is required. OSPFv3 is not yet part of INET, but is available in the extension ANSAINET. This model is not yet complete. It is missing some crucial parts such as correct generation of LSAs or the SPF calculation. A lot of people have contributed to the creation of this model such as Marek Černý, Jakub Mrázek, Michal Ruprich, Lukáš Galbička and Vladimír Veselý.

2.2 IS-IS

2.2.1 Introduction

IS-IS (Intermediate System to Intermediate System) is an IGP link-state routing protocol standardized back in 1992 within Open Systems Interconnection (OSI). Like OSPF, each Intermediate System (IS) is collecting information from other ISs in order to build a map of the topology and uses the Dijkstra's algorithm (same as OSPF) to calculate the shortest path to the destination. IS-IS defines two categories of devices in a network: Intermediate System and End System (ES). IS can be identified as a level-3 router, and ES identifies end-host (e.g., computer, server). IS-IS uses Connectionless-mode Network Protocol (CLNP) for addressing, unlike OSPF, which uses IP. Support for IP is defined in RFC 1195 and IETF extensions defined support for IPv6. Integration of these services is referred to as the Integrated IS-IS.

IS-IS is standardized in ISO/IEC 10589:2002 [11]. This chapter is based on information provided in [10], [13], and [2].

2.2.2 Overview

IS-IS allows topology division into areas. Routing within these areas is called Level-1 routing and routing between these areas is called Level-2 routing. Each IS can be either Level-1 only, Level-2 only or Level-1-2 and each IS communicates only on assigned level. Unlike OSPF, IS-IS assigns whole router to just one area with just one address. Each IS has assigned Network Entity Title (NET).

NET

NET refers to the Network Service Access Point (NSAP) which indentifies an instance of the IS-IS protocol. NET has length between 8 to 20 bytes and has 3 parts:

- **Area ID** (**1 - 13 Bytes**) - First byte is referred to as the Address Family Identifier (AFI) byte and defines the format of an address. Only relevant AFI for IS-IS has value of **49**, which defines private-addressing. Rest of the bytes contains area identification number.

- **System ID** (**6 Bytes**) - Unique identification of a router usually derived from the interface loopback address.

- **NSEL** - (**1 Byte**) - Defines service on a router. On Cisco devices this number is 0.

Net address is shown on figure 2.2.

$$\underbrace{\overset{\overline{\text{AFI}}}{49.0001}}_{\text{Area ID}}.\underbrace{1921.6800.0001}_{\text{System ID}}.\underbrace{00}_{\text{NSEL}}$$

Figure 2.2: NET Address

Designated IS

A designated IS is elected on all multi-access segments of the network. This effectively reduces traffic and thus saves bandwidth on those segments. Priority is a crucial parameter when electing the designated IS. Routers advertise their priority (0 - 127) which is configured on the interface. Router with the highest priority is elected as the Designated IS and is responsible for sending link-state advertisements which describe all routers attached to the network. If routers have the same priority, then router with highest Subnetwork Point of Attachment (SNPA) is elected instead. SNPA is the MAC address on a LAN and the local data link connection identifier (DLCI) on a Frame Relay network. Unlike OSPF, IS-IS does not use the backup designated router.

2.2.3 Packet Types

In IS-IS packets are called Protocol Data Units (PDUs). There are 3 categories and each category also has different PDU types as shown in table 2.1. Each PDU also has Type-Length-Value (TLV) field. Level-1 communication uses MAC address 0180:c200:0014 and level-2 0180:c200:0015.

Category	PDU Type	Type number
Hello	LAN Level-1 Hello	15
	LAN Level-2 Hello	16
	Point-to-point Hello	17
LSP	Level-1 LSP	18
	Level-2 LSP	20
SNP	Level-1 Complete SNP	24
	Level-2 Complete SNP	25
	Level-1 Partial SNP	26
	Level-2 Partial SNP	27

Table 2.1: IS-IS PDU Types

Hello PDU

IS uses Hello PDU to establish and maintain adjacencies with another IS. When establishing Level-1 adjacency, Area-ID has to match and this is not true for level-2. Hello PDUs differ

in type number for different levels. Hello packets also differ in sender system and intended receiver system. Three types exist.

- IIH (IS-IS Hello) - Between ISes to discover neighbors.

- ESH (ES Hello) - From ES to discover IS.

- ISH (IS Hello) - From IS to announce its presence to ES.

Hello PDU also advertises the Holding Time, which is basically a timeout or a dead interval. Unlike OSPF, Hello interval and Dead interval does not have to match. It can also be changed when IS-IS process is already running. Priority is included in these PDUs on multi-access segments.

Link State PDU

These PDUs are generally used to exchange link state information between IS-IS nodes. Each IS floods an LSP throughout an area to identify its adjacencies. There are two basic types.

- Level-1 LSP

- Level-2 LSP

Level-1 LSP describes connected links/prefixes on a router and is generated on routers that support level-1. All these level-1 LSPs generated by every IS in the same area create a level-1 LSP Database (LSPDB). Needless to say that all ISs in a same area have identical LSPDB. Level-2 describes all link/prefixes within the area and these are generated on routers that support level-2. This LSP is again flooded on level-2 subdomain. All level-2 ISs will again have identical level-2 LSPDB. Both types have the same message format. Each LSP has a sequence number and this number is increased every time IS advertises topology change.

Sequence Number PDU

Sequence number PDUs are used to synchronize LSDBs. These messages contain a summary of one or more LSPs. There are 2 basic types:

- Complete Sequence Number PDU (CSNP)

- Partial Sequence Number PDU (PSNP)

Usage of these types depends on the type of network. On point-to-point networks CSNP is used for first synchronization. PSNP is only used as acknowledgement when missing LSPs are requested. On broadcast networks CSNP is generated and sent by the designated IS. Other ISs compare advertised links in the CSNP to their own database. PSNP is used when LSPs are requested. Requested LSPs are indicated in the TLV field.

2.2.4   IS-IS Adjacency Process

There are 3 types of adjacencies in IS-IS.

- L1 adjacency - Between ISs supporting Level-1.

- L2 adjacency - Between ISs supporting Level-2.

- L1-2 adjacency - Between ISs supporting both Level-1 and Level-2.

L1-only IS and L2-only IS do not form any adjacency. When forming adjacencies, each IS goes through the 3-way handshake. This process is slightly different on point-to-point and broadcast circuits. The 3-way handshake process between router A and router B is following.

- **Down** - Initial state, router A sends his Hello PDU.

- **Initializing** - Router B receives Hello PDU from A and replies with his own Hello PDU.

- **Up** - Router A receives B's Hello PDU. At this point, bidirectional connection is established. Both routers have the MAC address of their neighbor.
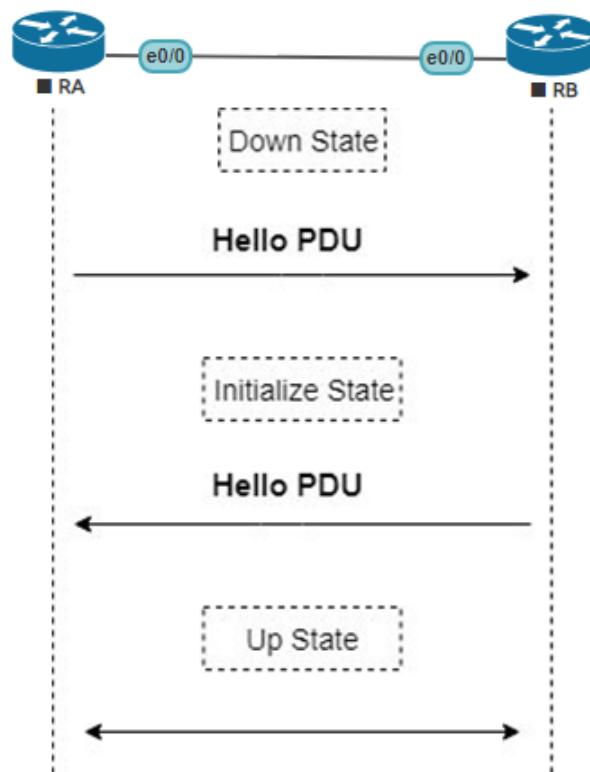

Figure 2.3: Adjacency states of IS-IS

2.2.5 IS-IS process

Process of synchronizing databases between each ISes differ by link type. Election of the designated IS takes place only on multi-access segment. Routers during initial synchronization exchange mainly LSP PDUs. There also exists a safety mechanism in a form of CSNP generation.

Point-to-point links

- Each IS builds and sends a set of CSNPs which contain a summary of connected links. One set is generated for each level the IS supports.

- Each IS compares received CSNPs with local LSPDB and missing LSPs are requested.

- IS Generates PSNPs as acknowledgement of received LSPs.

Multiaccess links

- DIS has 3 times shorter Hello Interval.

- DIS sends CSNP at regular intervals.

- Non DIS uses PSNP only if received CSNP contains unknown links.

2.2.6   IS-IS in OMNeT++

IS-IS is not yet part of the INET framework (as of January 2019), but is available in AN-SAINET extension. Authors of this protocol model are Matěj Hrnčiřík, Marcel Marek and Vladimír Veselý. Model was created as a part of Master's thesis in 2013 [13]. Implementation of this protocol is fully operational and is available since `ansainet-2.0`. This model does not support IP.

Chapter 3

Referential Configuration

3.1 Topology

In this section demonstrates basic OSPF an IS-IS configuration on topology shown on figure 3.1. All Cisco routers are running on version 15.4 with *ADVANCE ENTERPRISE SERVICES* feature set.
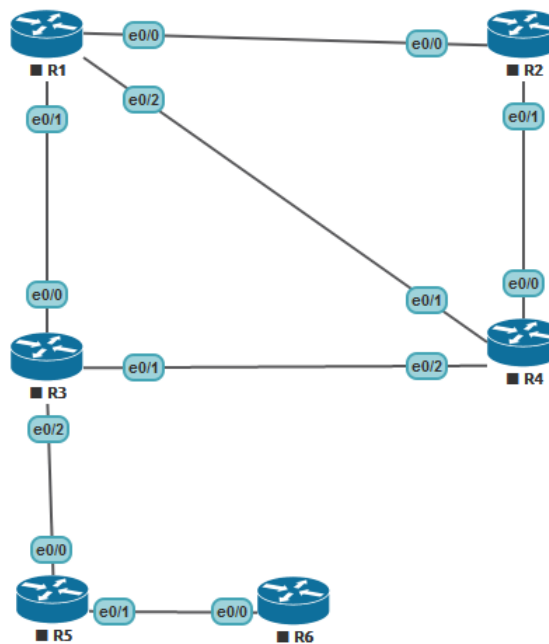

Figure 3.1: Referential topology

3.2 OSPF

All commands for configuring OSPF can be found in Cisco OSPF configuration guide [4].

3.2.1 OSPFv2

After logging onto a router we need to enter the configuration mode.

```
R> enable
R# configure terminal
```

When entering privileged and configuration mode, additional authentication may be required. Now we can enable OSPF routing with the following command.

```
R(config)# router ospf <process_id>
```

Where `process_id` is a number, specifically positive integer which has to be unique among other OSPF processes on the same router. Also at least one ip-assigned interface has to be `up`.

Now we need to define advertised networks and assign them an area ID.

```
R(config-router)# network <ip_address> <wildcard_mask> area <area_id>
```

`wildcard_mask` can be easily calculated by reversing all bits in the standard network mask.

LSA Throttling can be adjusted with following command. This is used to delay LSA generation or SPF calculation during change of the topology.

```
R(config-router)# timers throttle lsa all <start> <hold> <max>
```

SPF Throttling can also be adjusted.

```
R(config-router)# timers throttle spf <start> <hold> <max>
```

Where in both cases `start` represents initial delay, `hold` represents minimum hold time between two calculations or generations and `max` represents maximum delay.

Interface parameters

Each OSPF enabled interface can also be configured manually to override implicit values.

```
ip ospf cost <cost>
```

Sets cost on the OSPF interface.

```
ip ospf retransmit-interval <seconds>
```

Sets time between LSA retransmissions.

```
ip ospf transmit-delay <seconds>
```

Sets delay for transmission of LSU.

```
ip ospf hello-interval <seconds>
```

Sets Hello interval.

```
ip ospf dead-interval <seconds>
```

Sets Dead interval.

Configuration on different physical networks

```
R#(config-if) ip ospf network point-to-point
```

For point-to-point networks.

```
R#(config-if) ip ospf network point-to-multipoint
```

For point-to-multipoint broadcast networks.

```
R#(config-if) ip ospf network point-to-multipoint non-broadcast
```

For nonbroadcast networks.

3.2.2 OSPFv3

The legacy OSPFv3 configuration (i.e. without multiaddress family support) is very similar to the OSPFv2 configuration, but `network` command is absent.

```
R> enable
R# configure terminal
R(config)# ipv6 router <process-id>
```

Enters a configuration for OSPFv3 process.

```
R(config)# interface <interface>
R(config-if)# ipv6 ospf <process-id> area <area_number>
```

Enables legacy OSPFv3 on interface.

OSPFv3 Address families adds support of both IPv4 and IPv6 and both can be configured in same process. Configuration of OSPF Address families is as folows.

```
R> enable
R# configure terminal
R(config)# router ospfv3 <process-id>
```

Enters global configuration mode for IPv4 or IPv6 address family.

```
R(config-router)# area <cost | nssa | stub>
```

Configures the OSPFv3 area.

```
R(config-router)# auto-cost reference-bandwidth <Mbps>
```

Sets a reference value when calculating a metric for interfaces in a IPv4 OSPFv3 process.

```
R(config-router)# passive-interface <default | interface-type
interface-number>
```

Suppresses distribution of routing updates on selected interface when using IPv4 OSPFv3.

```
R(config-router)# router-id <router_id>
```

Sets custom Router ID.

```
R(config-router)# address family <ipv4 | ipv6>
```

Enters specific configuration for IPv4 or IPv6. These override global settings. Because the network command is missing again, OSPF has to be enabled manually on the interface.

```
R(config)# interface <interface>
R(config-if)# ospfv3 <process_id> <ipv4 | ipv6> area <area_number>
```

3.2.3 Configuration

Final topology for OSPF is shown on figure 3.2. This section will showcase basic OSPFv2
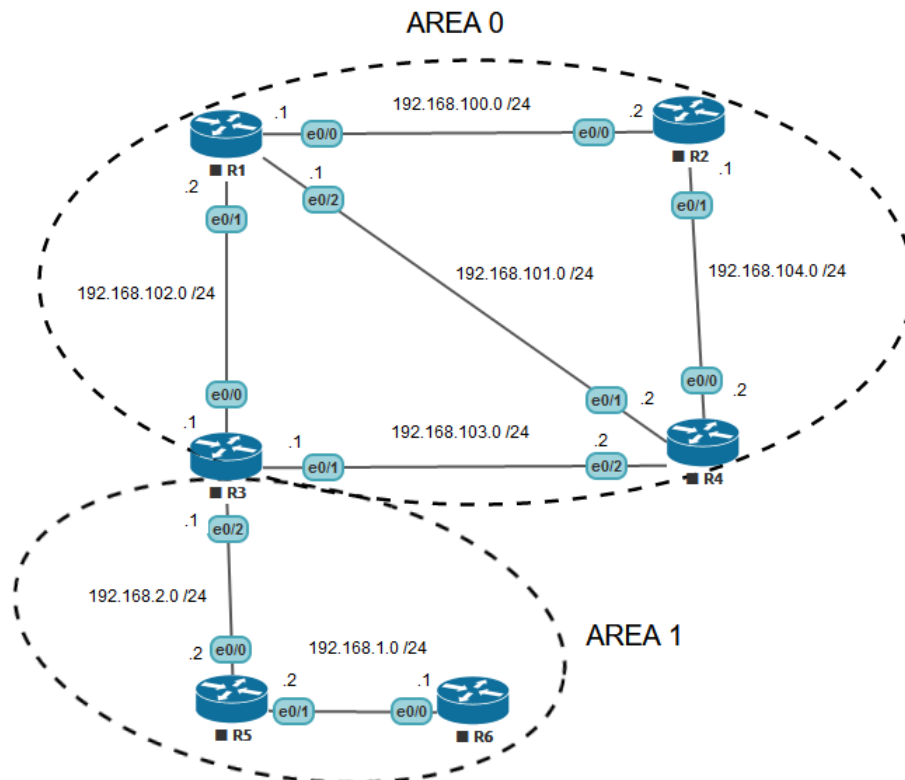

Figure 3.2: OSPF topology

configuration. It is also good practise to assign proper Router ID. Because implicit Router

ID is derived from interface IP address, simple way of setting this is to assign Loopback0 interface with IP address to better identify specific router in routing tables. Router R3 is ABR in this topology and has the following Configuration:

Configuration of interface `e0/0` is as follows.

```
R3(config)# interface ethernet 0/0
R3(config-if)# ip address 192.168.102.1 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# exit
```

Configuration of other interfaces is equivalent. Configuration of OSPF process on this ABR router is following.

```
R3(config)# router ospf 1
R3(config-router)# network 192.168.102.0 0.0.0.255 area 0
R3(config-router)# network 192.168.103.0 0.0.0.255 area 0
R3(config-router)# network 192.168.2.0 0.0.0.255 area 1
R3(config-router)# end
```

This configuration process has to be done for all routers in topology with correct addresses. Configuration for other routers is included in the attachment. Functionality of the OSPF process can be verified with command `show ip ospf database`.

3.3 IS-IS

This section will demonstrate basic IS-IS configuration on Cisco devices. All commands can be found in Cisco ISIS Command Reference [1]. Levels of routers are shown on figure 3.3. After logging onto a router we need to enter the configuration mode.

```
R> enable
R# configure terminal
```

When entering privileged and configuration mode, additional authentication may be required. First we have to assign a tag to an IS-IS process. This tag has to be unique among all IP router processes on that device.

```
R(config)# router isis <area_tag>
```

This will also enter router configuration mode. Now we have to set a `NET` address. This will identify device for IS-IS.

```
R(config-router)# net <network_entity_title>
```

It is also possible to set IS-IS level.

```
R(config-router)# is-type <level-1 | level-1-2 | level-2-only>
```

LSP Throttling can be adjusted. This is used to delay LSP generation or SPF calculation in order to reduce traffic.
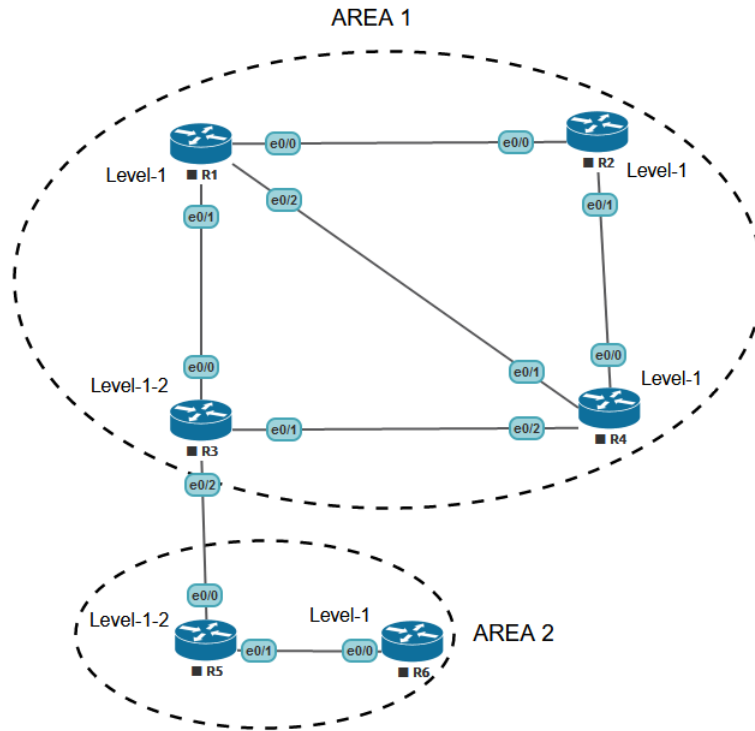
Figure 3.3: ISIS levels

```
R(config-router)# lsp-gen-interval <level-1 | level-2> <max-wait>
[<initial-wait> <second-wait>]
```

SPF Throttling can also be adjusted.

```
R(config-router)# spf-interval <level-1 | level-2> <max-wait>
[<initial-wait> <second-wait>]
```

In both cases `max-wait` represents maximum delay between two consecutive LSP generations or SPF calculations, `initial-wait` represents initial delay and `second-wait` represents delay between the first and second LSP generation or SPF calculation.

Now we have to enable IS-IS on the interface. So enter a interface configuration mode.

```
R(config)# interface <interface>
```

And enable IS-IS on interface.

```
R(config-if)#ip router isis <area_tag>
```

Where `area_tag` identifies to which IS-IS process this interface belongs.

```
R(config-if)#isis network point-to-point
```

To configure interface as point-to-point.
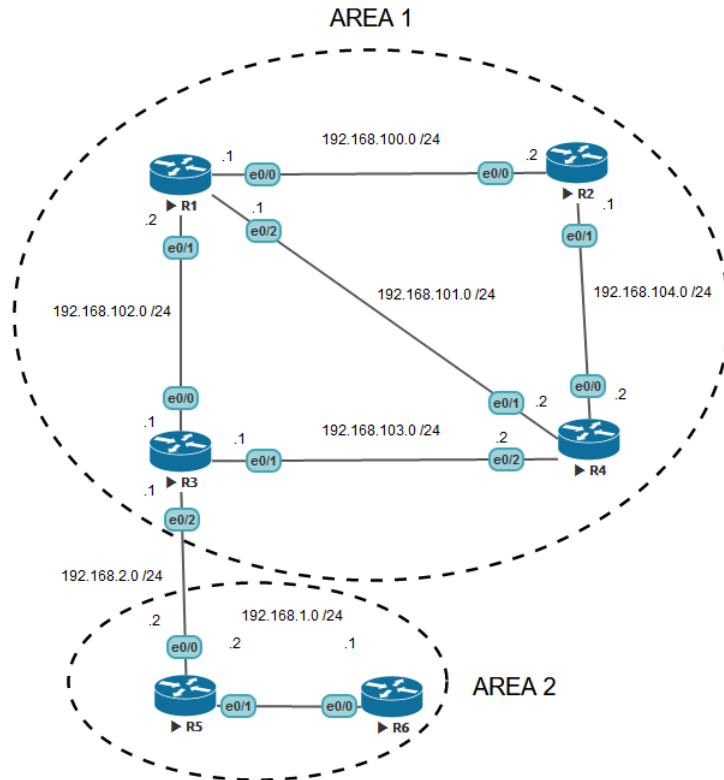
Figure 3.4: ISIS topology

Final topology is shown on figure 3.4. Configuration of Level-1 router R1 is following. To enable IS-IS we have to start a new process, assign NET address and level.

```
R1(config)# router isis 0
R1(config-router)# net 49.0001.0000.0000.1111.00
R1(config-router)# is-type level-1
R1(config-router)# exit
```

No we have to enable assign IS-IS process to interfaces.

```
R1(config)# interface ethernet 0/0
R1(config-if)# ip address 192.168.100.1 255.255.255.0
R1(config-if)# ip router isis 0
R1(config-if)# no shutdown
R1(config-if)# exit
```

Configuration for other interfaces is equivalent. IS-IS is now running on R1. Configuration for other router is included in the attachment.

Proper configuration of IS-IS can be verified for example by checking routing table with `show ip route` or with `show isis database`.

Chapter 4

Practice

This chapter focuses on measuring properties of OSPF and IS-IS on different topologies on Cisco devices and also in simulator OMNeT++ on their most recent implementations. Main parameters for these measurements are the convergence time, as it plays major role in network availability, a bandwidth usage and a message order. Traffic between nodes is recorded and stored mostly as `.pcap` files. Even though point-to-point configuration for router-to-router connection is preferred and has faster convergence, multiaccess medium generates more complex communication and thus is more interesting to test. Protocols settings on Cisco routers and in OMNeT++ are equivalent in order to produce comparable values.

4.1 Cisco

Cisco Systems is the biggest manufacturer of network hardware at the moment. Cisco routers are used in the most crucial networks. Cisco may have some proprietary features build into OSPF and IS-IS but I do not expect to see any violation of the official RFC. I expect to find Cisco implementation of these protocols much more efficient than their counterparts in OMNeT++. There are multiple measurements of the same scenario because race condition makes same scenarios differ in measured values quite a lot.

All Cisco routers are running on version `15.4` with *ADVANCE ENTERPRISE SER-VICES* feature set. All timers are set to default values.

4.1.1 Metodology

Each protocol is measured on two different topologies with two scenarios. First scenario is accomplished by delaying `no shutdown` command on particular interface until all other routers are synchronized. This simulates basic topology extension. Second scenario is accomplished by shutting down interfaces on all ends of a link at the same time. This simulates basic interface or communication medium error as is accomplished by simple applet created with the Embedded Event Manager. These applets use the `shutdown` command on particular interface on different routers at the same time. Both scenarios and topologies are aimed to be equivalent with their OMNeT++ counterparts.

All traffic is monitored with Wireshark and saved in `.pcap` files. All stated times defined in tables are calculated from these files and from logs of routing tables. Interface startup or shutdown is always used as the starting reference points. Routing table updates are monitored with the `debug ip routing` command. OSPF LSAcks are ignored as IS-IS uses

implicit acknowledgements. All routers have same settings.

OSPF

All routers are set is in the same area - `0.0.0.0`. All interfaces are set as broadcast and their metric or priority has not been altered. All addresses are IPv4 Class C. All timers are also set to default. Throttling values are as follows:

- SPF-start: 5000ms

- SPF-hold: 10000ms

- SPF-max-wait: 10000ms

- LSA-start: 0ms

- LSA-hold: 5000ms

- LSA-max: 5000ms

IS-IS

All routers are set in the same area - `49.0001`. All System IDs are of the same format - last 4 digits repeat routers number. All routers are configured to level-1 only communication and all used addresses are IPv4. No metric, timer or priority have been altered. Throttling values are as follows:

- SPF-max-wait: 10000ms

- SPF-initial-wait: 5500ms

- SPF-second-wait: 5500ms

- LSP-max-wait: 5000ms

- LSP-initial-wait: 50ms

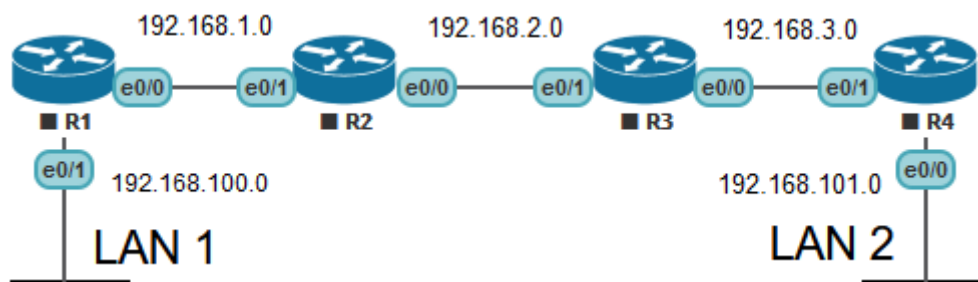- LSP-second-wait: 5000ms

4.1.2   Topology 1

Figure 4.1: Topology 1

This topology consists of four routers connected in a straight line. Critical error on any link would result in total communication interruption between LAN1 and LAN2. All address masks are `255.255.255.0`. Topology is shown on figure 4.1.

- **Scenario 1** - All routers are configured for specific routing protocol. Start of interface *e0/0* on *R1* is delayed until topology is fully converged. All other routers are synchronized and designated routers are elected. Start of the interface is the starting reference point.

- **Scenario 2** - All routers are configured for specific routing protocol and all interfaces are enabled. Starting reference point for this measurement is the time of a `shutdown` command on interface *e0/0* on *R2* or *e0/1* on *R3*.

OSPF

Scenario 1

Attached files: `cot1s1R1m0/1/2.pcap`, `cot1s1R4m0/1/2.pcap`

This scenario is repeated and measured three times. When interface *e0/0* on *R1* starts, its first Hello is delayed by 3 seconds. The time between send and received Hello differs across all 3 measurements as expected and can potentially delay convergence by up to *Hello interval* in seconds. *R1* is elected as the BDR before synchronization even starts. Communication continues with exchange of DBDs, LSRs, LSUs and finally LSAacks. Elapsed time between first DBD and last LSAck is between 5 and 7 seconds. LSUs usually contains multiple LSAs, making the communication very clean and efficient. Total number of data transmitted is **1746 Bytes**, excluding Hello Packets. Communication between *R3* and *R4* happens in a time frame of 5 second with the total of **620 Bytes** being transmitted. Recorded values are displayed in table 4.1.

	Measurement no.	1 [s]	2 [s]	3 [s]
	Interface startup	0.000	0.000	0.000
	First Hello	2.997	2.999	3.003
	Received Hello	5.962	4.115	5.322
R1	Last LSU	8.863	8.507	8.509
	Routing table update	8.863	8.507	8.509
	Last LSU	8.504	8.508	8.509
R4	Routing table update	10.971	9.123	10.334

Table 4.1: Cisco OSPF - Topology 1: Scenario 1

Scenario 2

Attached files: `cot1s2R1m0/1.pcap`, `cot1s2R4m0/1.pcap`

When the `shutdown` command is executed on interface *e0/0* on *R2*, there is a delay of approximately 0.5 second until LSU is generated and send to *R1*. This LSU is acknowledged and routing table is updated 5 second after receiving the LSU. This is caused by SPF Throttling. This feature delays actual calculation of shortest path by default for given number of seconds during network instability. Delay will actually grow when router receives another LSU in this delay time and delays calculation even further. This feature is fully explained in OSPF Shortest Path First Throttling guide by Cisco [6], [5] but is not unique to OSPF. *R3* in this scenario waits for the Dead interval before generating any LSUs. Communication between *R1* and *R2* consists of **176 Bytes**, communication between *R3* and *R4* of **302 Bytes**. Measurement is repeated with the `shutdown` command on interface *e0/1* on

R3 to measure LSUs for *R4*. First LSU is again generated 0.5 seconds after the interface shutdown and routing table update is 5 seconds after that. In this case, *R2* waits for the Dead interval before updating *R1* but results are equivalent. Measured values are shown on figure 4.2.

	Measurement no.	1 [s]	2 [s]
	Shutdown	0.000	0.000
R1	Last LSU	0.505	37.125
	Routing table update	5.512	42.132
R4	Last LSU	32.169	0.540
	Routing table update	37.136	5.511

Table 4.2: Cisco OSPF - Topology 1: Scenario 2

IS-IS

Scenario 1

Attached files: `cit1s1R1m0/1/2.pcap`, `cit1s1R4m0/1/2.pcap`

This scenario is repeated three times. When interface *e0/0* on *R1* starts, there is a 3 second delay before its first Hello PDU is send. However, responding Hello is received almost instantly across all measurements. Another set of Hello PDUs are exchanged before synchronization starts. *R2* is the DS. Synchronization starts with brief exchange of LSPs and continues with CSNP, which describe DSs whole database. *R1* requests LSPs by sending PSNP and synchronization ends by receiving these LSPs. No acknowledgements are used on multi-access networks. Whole communication between *R1* and *R2* excluding Hello PDUs contains **939 Bytes**. Communication between *R3* and *R4* captures only a set of LSPs by *R3* and consists of **300 Bytes**. Recorded values are listed in table 4.3.

	Measurement no.	1 [s]	2 [s]	3 [s]
	Interface Startup	0.000	0.000	0.000
R1	First Hello	3.010	2.998	3.005
	Received Hello	3.019	3.006	3.013
	Last LSP	8.947	6.590	8.773
	Routing table update	9.590	9.565	9.578
R4	Last LSP	5.578	5.068	5.075
	Routing table update	9.637	9.626	9.637

Table 4.3: Cisco IS-IS - Topology 1: Scenario 1

Scenario 2

Attached files: `cit1s2R1m0/1.pcap`, `cit1s2R4m0/1.pcap`

Measurement is repeated two times. When `shutdown` command is executed on *e0/1* on *R2*m first LPS is generated and send to *R1* within 0.1 seconds. Because of SPF Throttling, the routing table update is delayed by 5 seconds. LSP exchange consists of **95 Bytes**. Communication between *R3* and *R4* consists only of LSPs of quick succession with total of **227 Bytes** being transmitted. Generation of these LSPs by *R3* is delayed by the Hold timer. Recorded values are shown in table 4.4.

Measurement no.		1 [s]	2 [s]
	Shutdown	0.000	0.000
R1	Last LSP	0.083	0.075
	Routing table update	5.589	5.582
R4	Last LSP	23.161	22.186
	Routing table update	28.589	27.621

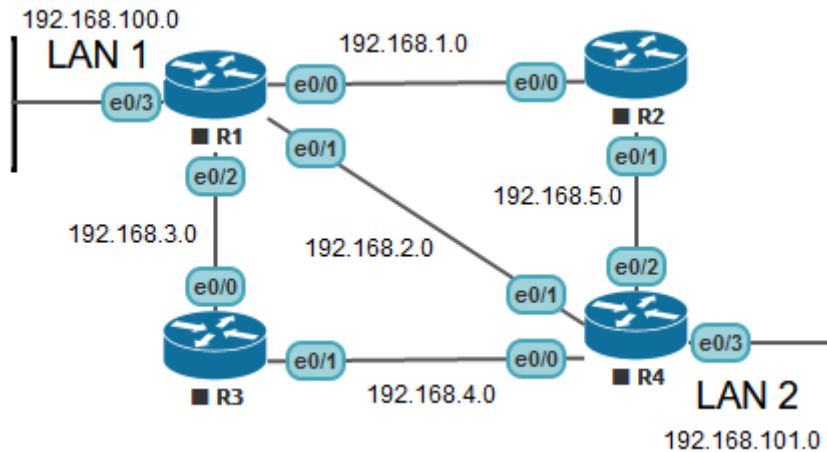Table 4.4: Cisco IS-IS - Topology 1: Scenario 2

Figure 4.2: Topology 2

4.1.3 Topology 2

This topology again consists of 4 routers but there are 3 possible paths between LAN1 and
LAN4. Mask of every involved IP address is `255.255.255.0`. This topology is shown on
figure 4.2. There are 2 scenarios for this topology.

- **Scenario 1** - All routers are configured for particular routing protocol. Start of
  interfaces *e0/0, e0/1, e0/2* is delayed until the rest of topology is convergent and
  DR/BDR/DS are elected. Start-up of these interfaces is the starting reference point.

- **Scenario 2** - All routers are configured for particular routing protocol and whole
  topology is convergent. Command `shutdown` is executed on interface *e0/1* on both
  R1 and *R4* at the same time.

OSPF

Scenario 1

Attached files: `cot2s1R1e0/1/2m1/2/3.pcap`, `cot2s1R4e0/1/2m1/2/3.pcap`

There is 3 seconds delay between start of the interface and its first Hello across all mea-
surements. *R1* is elected as the BDR on all interfaces a this election is done before syn-
chronization even starts. Communication continues with exchange of DBDs, LSRs, LSUs
and LSAcks. *R1* requests LSAs only on interface *e0/0* in all measurements. In case of
measurements 3, there are no recorded LSRs on other links, Two-Way State is achieved
earlier, LSU exchange is done faster and SPF Throttling did not occur as routing table was

updated immediately. This resulted in much faster convergence. The whole communication on interfaces *e0/0*, *e0/1* and *e0/2* of *R1* consists of **6482 Bytes** excluding Hello packets. This number is similar across all measurements. Recorded values are listed in table 4.5. **X** listed in the table could not be measured as this link does not exist.

	Measurement no.	1 [s]			2 [s]		
	Link towards	R2	R3	R4	R2	R3	R4
	Interface startup	0.000	0.000	0.000	0.000	0.000	0.000
R1	First Hello	2.994	2.995	3.000	2.992	2.997	2.998
	Received Hello	8.686	9.278	10.454	9.530	11.809	12.401
	Last LSU	14.190	14.190	14.191	15.034	15.034	15.034
	Routing table update	18.503			18.507		
R2	Routing table update	23.700			24.547		
R3	Routing table update	23.697			24.546		
R4	Last LSU	14.191	14.191	**X**	15.034	15.034	**X**
	Routing table update	23.694			24.546		

	Measurement no.	3 [s]		
	Link towards	R2	R3	R4
	Interface startup	0.000	0.000	0.000
R1	First Hello	2.990	2.990	2.999
	Received Hello	5.129	5.376	5.550
	Last LSU	8.497	8.497	8.498
	Routing table update	8.498		
R2	Routing table update	10.138		
R3	Routing table update	10.138		
R4	Last LSU	8.498	8.498	**X**
	Routing table update	10.135		

Table 4.5: Cisco OSPF - Topology 2: Scenario 1

Scenario 2

Attached files: `cot2s2R1e0/2m1/2.pcap`, `cot2s2R4e0/2m1/2.pcap`

LSU generation by *R1* and *R4* is delayed for 0.5 seconds. Every router delays the SPF calculation by 5 seconds due to SPF Throttling. Communication consists of LSUs and LSAck and on *e0/0* and *e0/2* on *R1* measures **1028 Bytes**.

IS-IS

Scenario 1

Attached files: `cit2s1R1e0/1/2m1/2/3.pcap`, `cit2s1R4e0/1/2m1/2/3.pcap`

When interfaces are started up, there is a 3 second delay before first Hello PDU is send. Neighbors Hello is received immediately in most cases with maximum delay of 1 second. *R1* never acts as the DS. *R1* starts receiving LSPs across all interfaces. If unknown LSPIDs are seen inside a CSNP from the the DS on given link, they are requested with a PSNP. Communication does now always contain these, because CSNP are send in regular 10 seconds intervals so it is just a matter of timing. Because not all received LSPs directly affect

Measurement no.		1 [s]		2 [s]	
	Link towards	R2	R3	R2	R3
	Interface shutdown	0.000	0.000	0.000	0.000
R1	Last LSU	0.567	0.567	0.573	0.536
	Routing table update	5.499		5.506	
R2	Routing table update	5.500		5.507	
R3	Routing table update	5.495		5.502	
R4	Last LSU	0.503	0.532	0.502	0.502
	Routing table update	5.502		5.509	

Table 4.6: Cisco OSPF - Topology 2: Scenario 2

routing table, there are some LSPs received after the routing table is updated. Communication on interfaces *e0/0*, *e0/1* and *e0/2* on *R1* consists of **3408 Bytes** excluding Hello PDUs and also excluding CSNPs unless its LSPIDs are requested with PSNP. Recorded values are listed in table 4.7. **X** listed in table could not be measured as this link does not exist.

Measurement no.		1 [s]			2 [s]		
	Link towards	R2	R3	R4	R2	R3	R4
	Interface startup	0.000	0.000	0.000	0.000	0.000	0.000
R1	First Hello	2.998	3.003	2.998	3.452	3.281	2.990
	Received Hello	3.506	3.011	3.866	3.460	3.289	2.999
	Last LSP	9.113	9.101	9.101	9.744	9.707	9.719
	Routing table update	9.573			9.569		
R2	Routing table update	10.502			9.578		
R3	Routing table update	9.569			9.586		
R4	Last LSP	9.134	9.126	**X**	9.128	9.128	**X**
	Routing table update	9.562			9.552		

Measurement no.		3 [s]		
	Link towards	R2	R3	R4
	Interface startup	0.000	0.000	0.000
R1	First Hello	2.998	3.297	2.998
	Received Hello	3.006	3.305	3.006
	Last LSP	9.101	9.101	9.101
	Routing table update	9.564		
R2	Routing table update	9.565		
R3	Routing table update	9.595		
R4	Last LSP	9.134	9.134	**X**
	Routing table update	9.561		

Table 4.7: Cisco IS-IS - Topology 2: Scenario 1

Scenario 2

Attached files: `cit2s2R1e0/2m1/2.pcap`, `cit2s2R4e0/2m1/2.pcap`

When the `shutdown` command is executed, the first set of LSPs is generated between 0.1

and 0.2 seconds. Another sets are generated later but these do not have an impact on SPF calculation as update of the routing table is not delayed by these. LSP exchange on interface *e0/0* and *e0/2* on *R1* consists of **850 Bytes**. Recorded values are shown in table 4.8.

	Measurement no.	1 [s]		2 [s]	
	Link towards	R2	R3	R2	R3
	Interface shutdown	0.000	0.000	0.000	0.000
R1	Last LSP	4.523	4.575	0.140	0.131
	Routing table update	5.556		5.555	
R2	Routing table update	5.527		5.539	
R3	Routing table update	5.527		5.528	
R4	Last LSP	0.121	0.121	0.132	0.131
	Routing table update	5.558		5.562	

Table 4.8: Cisco IS-IS - Topology 2: Scenario 2

4.1.4 Comparison

OSPF and IS-IS are very similar in behaviour as expected. Faster convergence time would be achieved with point-to-point settings. All of these findings only apply on broadcast links. Both protocols are running on the same default timers and optimization of these timers would greatly improve the convergence time. SPF Throttling plays a big role when measuring convergence time for both protocols as it delays calculation of shortest path every time topology is changed until the topology becomes stable.

In the case of routing table update speed, IS-IS is faster across most of the measurements. DIS sends Hello with higher frequency when at least one IS is found on a link. Also when first IS is found on a link, the responding Hello PDU is send almost immediately and thus Up State is always achieved faster. Because responding Hello packets are not created immediately, the Two-Way State is usually achieved with a delay. OSPF always uses the LSDBs and LSRs for the first synchronization before LSUs are exchanged. IS-IS mainly just uses LSRs for the initial synchronization. Request in a form of a PSNP is only used when CSNP is received with unknown links. IS-IS uses implicit acknowledgement and OSPF uses LSAck. IS-IS measurements are overall much more consistent as routing table updates differ by 1 seconds at worst and very similar time is achieved on both topologies. OSPF achieves same convergence time in the second scenario on both topologies as IS-IS. IS-IS generated LSPs quicker by 300ms - 400ms but throttling delayed routing table update by 5 seconds. In the first scenario OSPF results are not very consistent. On first topology the routing table updates differ by almost 2 seconds and second topology shows difference of almost 15 seconds. This delay is mostly caused by SPF and LSA Throttling. Default settings for OSPF and IS-IS are different. Even though throttling increases efficiency it also increases the convergence time.

In the case of bandwidth usage, OSPF and IS-IS differ in design. OSPF traffic during network stability is kept on minimum. However, also generates a significant amount of traffic during synchronization. OSPF *R1* in the first topology exchanged **1746 Bytes** with *R2* while IS-IS *R1* only exchanged **939** Bytes during initial synchronization. In the second topology, OSPF *R1* exchanged with other routers total of **6482 Bytes** while IS-IS exchanged only **3408 Bytes**. These numbers do not include Hello messages. However,

during network stability IS-IS uses much more bandwidth. On multi-access segments, IS-IS DIS has 3 times shorter Hello interval and also periodically sends CSNPs while other ISes send one Hello on every Hello interval. All these Hello PDUs are also padded with zeros to the Maximum Transmission Unit (MTU) size (in this case 1514B) in order to detect MTU mismatches. This can be safely disabled if all MTUs on a link are the same. OSPF only uses Hello Packets during network stability just to keep adjacency.

4.2 OMNeT++

Omnet++ is a great way to simulate and observe behavior of network routing protocols such as OSPF and IS-IS. My goal in this chapter is to set up the same topologies and scenarios as in the Cisco section 4.1 to produce somewhat comparable values. Because of undeniable determinism of simulation, I expect result to be more predicable.

Both INET and its extension ANSAINET frameworks are developed by the community around OMNeT++ and thus some bugs are expected to exist. Both protocols are not very well tested and they are also updated a lot. Because both protocols in their most current forms are not included in the same framework, there are 2 settings for testing:

- `OMNeT++ 5.4.1` & `INET 4.1.0` [8] & Windows 10 for OSPF

- `OMNeT++ 5.4.1` & `ANSAINET 4.0` [7] & Ubuntu 18.04 for IS-IS

All tested scenarios are fingerprinted. Fingerprints are explained later in this section. All source files are included in the file attachment.

4.2.1 Metodology

Each protocol is measured on two topologies with two different scenarios. First scenario focuses on the initial convergence time and second focuses on the convergence time after change in the topology occurred.

First scenario is accomplished by simply delaying start of a single router. This simulates connection of new router to already converged topology. This is accomplished by delayed start in OSPF measurements and by connection of a new link in IS-IS measurements. Second scenario is accomplished by disconnecting a link between specific routers. Both topologies that are tested in these scenarios are the exact same as in the Cisco section 4.1. All interfaces are set as on multi-access network to generate more complex traffic.

All traffic is monitored and analyzed. This traffic is also saved in `.pcap` files in case of OSPF. IS-IS does not support such export as packet serializer is not implemented at the moment. Functionality of IS-IS is analyzed only from simulation event logs. All inputs and outputs of these measurements are included in the file attachment. All routers have equivalent setting in order to measure meaningful values. Starting point for each measurement is a event — either router startup or a link disconnection. Because of few bugs in both protocols, some measurements lack some values as they could not be measured. All bugs and strange behaviour is listed at the end of this section. Setting of both protocols are as follows:

OSPF

All routers are set in the same area - `0.0.0.0`. OSPF is enabled on both connected networks on every router. All interfaces are set as broadcast and their priority is `1`. Output

cost of all interfaces is 0. IPv4 routing tables are monitored for updates.

IS-IS

All routers are set in area 49.001. All ISes are operation only on level-1. Priority on all interfaces is 1. Because actual implementation of IS-IS does not support IPv4, CLNS routing tables are monitored instead.
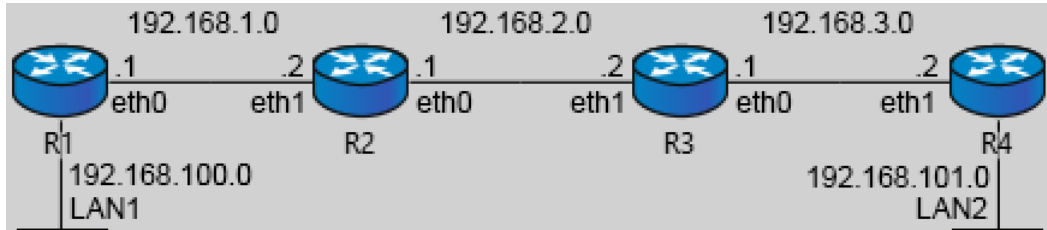
4.2.2 Topology 1

Figure 4.3: Topology 1

First topology is a simple router line. Disruption of any link between *R1* and *R4* would result in total communication disruption between *LAN1* and *LAN2*. Mask of every involved IP Address is 255.255.255.0. Topology is shown on figure 4.3.

- **Scenario 1** - All routers are configured for specific routing protocol. Start of router *R1* or its connection to the rest of topology is delayed by 65 seconds. This is enough time for routers to synchronize their databases and to elect designated routers. Start of *R1* is the starting reference point.

- **Scenario 2** - All routers are configured for specific routing protocol. All routers are running and their databases are synchronized and designated routers are elected. Link between *R2* and R3 is disconnected after 65 seconds. This represents the starting reference point.

OSPF

Scenario 1

Attached files: `oot1s1R1.pcap, oot1s1R4.pcap`

Communication between *R1* and *R2* begins with exchange of Hello packets. *R1* sends its Hello almost instantly as there is basically no delay. *R2* has correctly elected itself as the DR on this segment and election of *R1* as the BDR is done after initial synchronization. All DBD and LSR packets are send as unicast, content of these packets is as expected. All initial LSUs are send to 224.0.0.5 and retransmissions are correctly send directly to the neighbor. Communication between *R3* and *R4* contains few LSUs and there is also a need for a retransmission. All retransmissions that occur in this scenario are, however, badly structured as 4 Bytes indicating the *Number of LSAs* is always set to 0. It does not seem to matter as they are correctly acknowledged by the recipient. Communication between *R1* and *R2* consists of **2082 Bytes**, communication between *R3* and *R4* contains **898 Bytes** both excluding Hello packets. There also is a seemingly useless LSU and its LSAck on both ends after approximately 25 seconds. These information were already transmitted

		[s]
	R1 Startup	0.000
	First Hello	0.096
R1	Received Hello	5.107
	Last LSU	15.107
R4	Last LSU	15.107

Table 4.9: OMNeT++ OSPF - Topology 1: Scenario 1

and acknowledged. Recorded values are listed in table 4.9.

Scenario 2

Attached files: `oot1s2R1.pcap`, `oot1s2R4.pcap`

When virtual cable between *R2* and *R3* is disconnected, *R2* generates new LSUs for *R1*. *R1* receives first LSU instantly as there is zero measurable delay. *R1* acknowledges only first received LSU as `Sequence Number` of LSAck corresponds to the first received LSU. Retransmission of not acknowledged LSU by *R2* then occurs. It is correctly send as unicast but its `Number of LSAs` is again set to 0. Same misbehavior as in the first scenario. This badly structured packet is again acknowledged by *R1*. *R3* also sends new LSUs to *R4* instantly. All these LSUs are send in quick succession after each other and even contain one duplicate. *R4* acknowledges only two of the three. Retransmission then occurs and is again acknowledged correctly even though its `Number of LSAs` is set to 0. Convergence time on both ends is prolonged by 5 seconds because of these retransmissions. Communication between *R1* and *R2* consists of **482 Bytes** and communication between *R3* and *R4* accounts for **780 Bytes** both excluding Hello packets. Measured values are listed in table 4.10.

		[s]
	Link disconnected	0.000
R1	Last LSU	5.000
	Routing table update	5.089
R4	Last LSU	5.000
	Routing table update	5.089

Table 4.10: OMNeT++ OSPF - Topology 1: Scenario 2

IS-IS

Scenario 1

Attached files: `oit1s1.txt`

When a link between *R1* and *R2* is connected, the first set of Hellos is exchanged quickly. Communication continues with *R2* sending CSNP, *R1* responds with PSNP and LSP exchange begins. Routing table of *R1* is then updated. Even though there are some LSsP received by *R4* as early as 6 seconds after *R1* is connected to the topology, routing table is updated after almost a whole minute. Overall communication contains many more LSP exchanges than expected and some are even 40 second apart. This however seems to stop after 85 seconds when topology is stable until the end of the simulation. All exchanged PDUs

are exactly 72 Bytes long, this is probably caused by padding. Communication between *R1* and *R2* before *R1* updated its routing table contains **1368 Bytes**. Traffic Between *R3* and *R4* contains **360 Bytes**. Measured times are listed in table 4.11.

		[s]
	R1 Startup	0.000
R1	First Hello	0.949
	Received Hello	2.974
	Last LSP	83.630
	Routing table update	16.320
R4	Last LSP	57.640
	Routing table update	51.976

Table 4.11: OMNeT++ IS-IS - Topology 1: Scenario 1

Scenario 2

Attached files: `oit1s2.txt`

Because routing table of *R4* after 65 second still did not contain `R1`, this scenario had to be postponed by 10 seconds. So link between *R2* and `R3` is disconnected after 75 seconds total. After this link was disconnected, no immediate reaction occurred. Routers exchange first LSP after 30 seconds and very small flow of LSP continues for another 40 seconds. However, even after 125 seconds routing table of `R1` still contains disconnected path to `R4` and vice versa. This was also tested with doubled simulation time but with the same result. Communication between *R1* and *R2* contains **1008 Bytes**, excluding Hello Packets and almost all of CSNPs. Traffic between *R3* and *R4* contains **936 Bytes**. Recorded times for the last LSPs are listed in table 4.12.

		[s]
	Link disconnected	0.000
R1	Last LSP	72.310
R4	Last LSP	82.970

Table 4.12: OMNeT++ IS-IS - Topology 1: Scenario 2
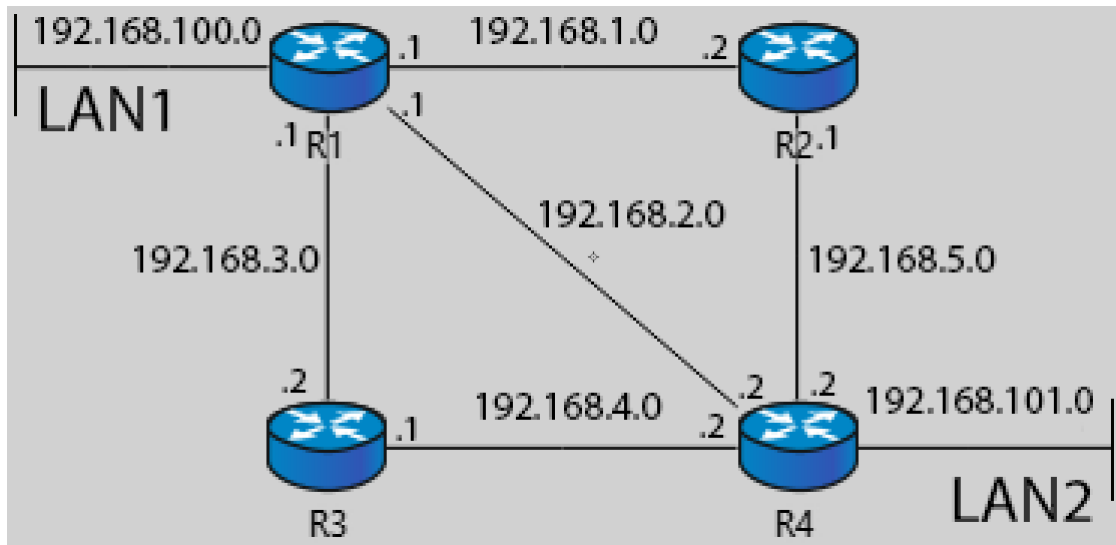
4.2.3 Topology 2

Figure 4.4: Topology 2

Second topology is a little more complex as there are multiple paths between LAN1 and LAN2. Mask of every involved IP Address is `255.255.255.0`. Topology is shown on figure 4.4.

- Scenario 1 - Routers are configured for specific protocol. Start of router *R1* is delayed by 105 seconds. All the other routers are synchronized by this time. Start of *R1* is the starting reference point.

- Scenario 2 - Routers are configured for specific protocol. All routers are synchronized and designated routers are elected. Link between *R1* and *R4* is disconnected after 105 seconds. This represents the starting reference point.

OSPF

Scenario 1

Attached files: `oot2s1R1.pcap`, `oot2s1R4.pcap`

Communication between *R1* and *R2,R3,R4* begins correctly with Hello packets. There is again basically zero delay. Communication continues with DBD, LSR and LSU exchange.*R1* on all link is elected as the BDR after 20 seconds. There is total of 8 LSU retransmissions and all of those have the `Number of LSAs` set to 0. Convergention is prolonged by at least 5 seconds because of these retransmissions. Total amount of traffic recorded on interfaces *e0/0*, *e0/1* and *e0/2* accounts for **10010 Bytes** excluding Hello packets. Measured values are listed in table 4.13. **X** listed in table could not be measured as this link does not exist.

Scenario 2

Attached files: `oot2s2R1.pcap`, `oot2s2R4.pcap`

After the link between *R1* and *R4* is disconnected both routers immediately generate LSUs

	[s]		
Link towards	R2	R3	R4
R1 Startup	0.000	0.000	0.000
R1 First Hello	0.089	0.098	0.111
R1 Received Hello	5.092	5.096	5.090
R1 Last LSU	20.092	20.092	20.092
R4 Last LSU	20.092	20.092	**X**

Table 4.13: OMNeT++ OSPF - Topology 2: Scenario 1

for both *R2* and *R3*. These are acknowledged by both. Retransmission occurs on all links once, potentially delaying convergence time. Routers that are the closest to the disruption update their routing tables instantly. Total amount of traffic recorded on interfaces *e0/0* and e0/2 on *R1* accounts for **2870 Bytes**. Recorded values are listed in table 4.14.

Way more interesting traffic can be observed on the link between *R1* and *R4* before it is actually disconnected. Communication seems to be stuck in endless loop. Routers synchronize without electing DR and BRD as they wait for the `Wait timer` (Dead interval) to expire. Routers correctly elect *R2* as DR and *R1* as BDR after these 40 seconds and they begin to synchronize again. This synchronization lasts for whole 65 seconds until the link is disconnected by the scenario manager. Routers are stuck in endless loop with *R1* retransmitting LSU every 5 seconds without *R4* ever acknowledging. This LSU does not contain any information about any link and has correctly `Number of LSAs` set to 0. Adjacency between the two is however never destroyed as there still is a flow of Hello packets.

File `oot2s2R1_R4.pcap` only contains communication between *R1* and *R4* for easier analysis.

		[s]	
Link towards		R2	R3
Link disconnected		0.000	0.000
R1	Last LSU	5.000	5.000
R1	Routing table update	0.000	
R2	Routing table update	5.000	
R3	Routing table update	5.000	
R4	Last LSU	5.000	5.000
R4	Routing table update	0.000	

Table 4.14: OMNeT++ OSPF - Topology 2: Scenario 2

IS-IS

Scenario 1

Attached files: `oit2s1.txt`

When links are connected to all the other routers, first set of Hellos is generated in reasonable time. When the Two-way state is established, communication continues with mostly LSP exchange. There are also some PSNP requests. LSP exchange continues even after the routing table is updated. Routing tables are updated quickly compared to the first

scenario. Total of 160 LSPs, 5 PSNP responses to 5 CSNPs are exchanged on *eth0, eth1 and eth2* on *R1*. Because all exchanged messages are padded to 72 bytes, this gives total of **12240 Bytes**. Recorded values are listed in table 4.15. **X** listed in the table could not be measured as this link does not exist.

		[s]		
	Link towards	R2	R3	R4
	Link connected	0.000	0.000	0.000
R1	First Hello	0.664	2.147	2.223
	Received Hello	2.447	2.417	2.932
	Last LSP	33.630	33.960	33.630
	Routing table update	9.036		
R2	Routing table update	11.980		
R3	Routing table update	11.980		
R4	Last LSP	33.630	33.630	**X**
	Routing table update	5.660		

Table 4.15: OMNeT++ IS-IS - Topology 2: Scenario 1

Scenario 2

Attached files: `oit2s2.txt`

When link between *R1* and *R4* is disconnected, there is no immediate reaction. First set of LSPs is generated after 15 seconds and this exchange lasts for 25 seconds. Routing table of *R2* and *R3* did not change at all, but route between *R1* and *R4* changed its `nexthop` and doubled metric. Interfaces *eth0* and *eth2* on *R1* processed **4032 Bytes**. This does not include Hello PDUs. Recorded times are listed in table 4.16.

		[s]	
	Link towards	R2	R3
	Link disconnected	0.000	0.000
R1	Last LSP	40.280	40.280
	Routing table update	32.640	
R4	Last LSP	40.610	40.280
	Routing table update	33.300	

Table 4.16: OMNeT++ IS-IS - Topology 2: Scenario 2

4.2.4 Comparison

These two models are hard to compare. OSPF uses IPv4 addresses and IPv4 routing table while IS-IS at this current stage supports only CLNS. While OSPF model is able to generate traffic into `.pcap` files, IS-IS does not contain parser for viewing TLV fields. Both models are functional as they exchange link information, respect their assigned areas and levels, have configurable timers and support some simple manipulation. My scenarios also tested only small portion of functionalities.

First measurement on the first topology shows that IS-IS is much slower with updating *R4*. While *R1* seems to be updated fairly quickly on both models, *R4* updates its routing

table much later. First LSP is received after 35 seconds and another set follows every 15 seconds or so. During second scenario on first topology, IS-IS was so slow that even after 65 seconds *R4s* routing table did not contain path to the *R1*. Also when scenario was finally executed, not a single router removed entry from its routing table. OSPF model performed comparably to the Cisco in this case. IS-IS generated less traffic, but the difference should be more significant, because OSPF has much lower bandwith usage after the synchronization.

First scenario on second topology shows that IS-IS can deliver pretty solid convergence time. However, LSP traffic is present ever after routers are synchronized. *Routing table update* could not be measured on OSPF model. Second scenario is in heavy favor of OSPF. Routers closest to the disruption updated routing tables instantly, more than 30 second earlier than IS-IS. IS-IS also used more bandwidth even for synchronization.

IS-IS model seems to have slower convergence than OSPF, especially on the first topology. OSPF also used much less bandwidth overall. IS-IS generates too much unnecessary traffic and sometimes reacts very slowly to a change in the topology.

Problems

Acknowledgement has to be made right at the start. All people included in the creation of these models did incredible job and these models are absolutely usable with just a little power of will. Goal of this comparison is to expose some strange and possibly unintended behaviour to help those people make these models even better. Some of these problems are not a straight violations of the RFC but they are still worth pointing out.

INET 4.1.0 - OSPFv2

Version: `inet-4.1.0-810053f713`.

Implementation of OSPFv2 on Cisco devices proved to be much more polished and efficient than the one in INET. Brief look at the recorded `.pcap` files exposed following problems in the communication:

- Missing acknowledgements for multiple LSUs describing the same Link.

- Retransmitting badly structured unicast LSUs (at least generated to `.pcap` files).

- Generation of pointless LSUs.

- Getting stuck in endless loop of retransmissions after the `wait-timer`[1].

Because of these problems `.pcap` files captured almost double the amount of OSPF traffic than referential (Cisco). Sending multiple LSAs inside LSUs would also reduce this traffic a little bit and reducing useless LSU generation would decrease CPU time. Other problems are mostly about node manipulation as OSPF sometimes struggles to adapt to new node properties.

Routing table problem

If a partoicular router starts with a time delay (by setting startupTime cvar inside OSPF or with Scenario Manager), any other router that receives information about its links will not use it to update their routing table. Standard OSPF communication still occurs but

---

[1]Topology 2 - Scenario 2

not a single routing table actually changes. Delayed router also wont update its routing table even after the OSPF synchronization has ended. Because of this problem, I could not monitor routing table changes during some measurements.

There is a fingerprinted scenario showcasing this problem inside the *OSPF1* folder in the file attachment.

Link manipulation problem

If a virtual link between routers is disconnected and then reconnected, OSPF starts to send Hello packets but will never discover neighbors. After the whole router is restarted, this problem perishes, but routing table problem occurs.

There is a fingerprinted scenario showcasing this problem inside the *OSPF1* folder in the file attachment.

ANSAINET 4.0.0 - IS-IS

Because TLV fields are not readable, functionality of this model is much harder to analyze. Following problems have been observed from the simulation logs:

- Slow (or non-existent[2]) routing table updates.

- Probable generation of useless LSPs.

I did not observe any other problem with communication itself. Some interesting node and interface manipulation errors appeared.

Unspecified NET problem

This problem seems to be tied to the first topology. If interface *eth0* on *R1* is not specified in the `.xml` file and *NET* of *R3* is set to specific address, simulation crashes during first CSNP exchange.

Example of this crash is showcased inside the `ISIS1` in the file attachment.

Router shutdown problem

If a particular router is shut down with the `ScenarioManager` while its gates are still connected, then message on this gate will crash the whole simulation. `ScenarioManager` can also be used as a workaround as it is possible to disconnected virtual links instead. This has the same effect while not crashing the simulation.

Example of this scenario is listed inside the `ISIS1` folder in the file attachment.

4.2.5 OMNeT++ Fingerprints

Fingerprint in OMNeT++ is a generated hash that is calculated from various simulation events. Fingerprint captures trajectory of a simulation and is calculated continuously during the simulation. By comparison of two fingerprints it is possible to determine if a change in the code made simulation deviate from the previous version. This is useful for verification of code refactoring for example. OMNeT++ provides option to include referential fingerprint for a simulation. When this simulation is finished, the new and the referential fingerprints are compared. These section is based on OMNeT++ Manual [16].

---

[2]Topology 1: Scenario 2

Ingredients

OMNeT++ provides an option to select simulation aspects which are used for the hash calculation. These are specified after the '/' char at the end of a fingerprint. All fingerprints included in this theses use following ingredients: `telx`. Whole fingerprint may look like this: `4a00-8cfc/telx`.

- e: event number
- t: simulation time
- n: message/event full name
- c: message/event class name
- k: message kind
- l: message (packet) bit length
- o: message control info class name
- d: message data
- i: module id

- m: module full name
- p: module full path
- a: module class name
- r: random numbers drawn
- s: scalar results
- z: statistic results
- v: vector results
- x: extra data added programmatically

Multiple Fingerprints

It is also possible to set multiple fingerprints. In that case they are separated by a comma or a space. Space acts as logical OR while comma acts as logical AND. Because Intel and AMD CPUs handle floating point number differently, calculated fingerprints can differ. It is good practice to include both if possible with a space separator.

Chapter 5

Conclusion

The goal of the theory chapter is to introduce fundamentals of routing protocols OSPF and IS-IS. These sections provide basic understanding about overall architecture, used message types and also describe its process as a whole. The best sources of information for this chapter are, of course, the standards. Even though they can be hard to read, they provide the best reliable source of information. Standards `RFC 2328` and `RFC 5340` have proven to be very detailed. Because `ISO/IEC 10589` standard for IS-IS is much harder to read and study from compared to RFCs, I mostly used the book *The Complete IS-IS Routing Protocol* by *Hannes Gredler* and *Walter Goralski* which is much easier to grasp.

The section with configuration shows a simple way to configure these protocols on Cisco devices. Because it includes only basic configuration relevant for this thesis, sources with full sets of commands for both OSPF and IS-IS are referenced.

Practice chapter contains measurements of both protocols on Cisco devices and also in OMNeT++ simulator. Cisco measurements mostly serve as a reference point.

Even though both protocol models are operational, they are far from complete. Measurements of the OSPF model have produced some pretty comparable values. This model was slightly slower with the distribution of updates and also generated much more traffic. This is mostly caused by bad acknowledgements. My measurements also exposed some interesting behaviour, which is described in the OMNeT++ chapter. Protocol model of IS-IS does not yet support IPv4 or IPv6, so all measurements are done using CLNS. This model reacts slowly to the changes in the topology but otherwise is fully operational. Parser for reading TLV fields during simulation is missing, which also reduces the amount of the available information about the functionality of this model. OSPF on the other hand even supports generation of `.pcap` files.

In regard to this project, the next step for me is to contribute to the ANSAINET project. I think there is a great potential in this project and maybe I can prove myself useful.

I hope that my work can also prove useful for the creators of these models. Hopefully, they can use some of my findings to further improve their functionality.

Bibliography

[1] *Cisco IOS IP Routing: ISIS Command Reference.* [Online].
Retrieved from: https://www.cisco.com/c/en/us/td/docs/ios/iproute_isis/
command/reference/irs_book.pdf

[2] *IS-IS Overview and Basic Configuration.* [Online].
Retrieved from: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/
iproute_isis/configuration/15-mt/irs-15-mt-book/irs-ovrw-cf.pdf

[3] *OSPF Design Guide.* 2005. [Online].
Retrieved from: https://www.cisco.com/c/en/us/support/docs/ip/open-
shortest-path-first-ospf/7039-1.html

[4] *OSPF Configuration Guide.* 2018. [Online].
Retrieved from: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/
iproute_ospf/configuration/xe-16/iro-xe-16-book/iro-cfg.html

[5] *OSPF Link-State Advertisement Throttling.* 2018. [Online].
Retrieved from: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/
iproute_ospf/configuration/xe-16-6/iro-xe-16-6-book/iro-lsa-throt.pdf

[6] *OSPF Shortest Path First Throttling.* 2018. [Online].
Retrieved from: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/
iproute_ospf/configuration/xe-16-6/iro-xe-16-6-book/iro-short-path.pdf

[7] *Automated Network Simulation and Analysis.* 2019. [Online].
Retrieved from: https://ansa.omnetpp.org/

[8] *INET Framework.* 2019. [Online].
Retrieved from: https://inet.omnetpp.org/

[9] Coltun, R.; et al.: OSPF for IPv6. RFC 5340. RFC Editor. July 2008.

[10] Gredler, H.; Goralski, W.: *The complete IS-IS routing protocol.* 01 2005. ISBN
978-1-85233-822-0.

[11] ISO/IEC 10589:2002. Standard. International Organization for Standardization.
Switzerland. November 2002.

[12] Lindem, A.; et al.: Support of Address Families in OSPFv3. RFC 5838. RFC Editor.
April 2010.

[13] Marek, M.: *Modelování protokolů IS-IS a TRILL.* Master's Thesis. Brno University
of Technology, Faculty of Information Technology. Brno, Czech Republic. 2013.

[14] Moy, J.:   OSPF Version 2. RFC 2328. RFC Editor. April 1998.

[15] Ruprich, M.: *MODELING OF OSPFV3 LINK-STATE ROUTING PROTOCOL*. Master's Thesis. Brno University of Technology, Faculty of Information Technology. Brno, Czech Republic. 2017.

[16] Varga, A.: *OMNeT++ Simulation Manual*. 2019. [Online]. Retrieved from: https://doc.omnetpp.org/omnetpp/manual/

[17] Veselý, V.; Grégr, M.: Open Shortest Path First. 2016.

A Attached Files

Attached files are included on the enclosed CD with following content.

- **xzavre10.pdf** - PDF version of this thesis

- **source/** - OMNeT++ simulation files

- **output/** - Recorded `.pcap` and `.txt` files with traffic

- **README** - File structures and description

B Cisco Configuration

OSPFv2

```
R4> enable
R4# configure terminal
R4(config)# interface ethernet 0/0
R4(config-if)# ip address 192.168.104.2 255.255.255.0
R4(config-if)# no shutdown
R4(config-if)# exit
R4(config)# interface ethernet 0/1
R4(config-if)# ip address 192.168.101.2 255.255.255.0
R4(config-if)# no shutdown
R4(config-if)# exit
R4(config)# interface ethernet 0/2
R4(config-if)# ip address 192.168.103.2 255.255.255.0
R4(config-if)# no shutdown
R4(config-if)# exit
R4(config)# router ospf 1
R4(config-router)# network 192.168.104.0 0.0.0.255 area 0
R4(config-router)# network 192.168.101.0 0.0.0.255 area 0
R4(config-router)# network 192.168.103.0 0.0.0.255 area 0
R4(config-router)# end
```

IS-IS

```
R3> enable
R3# configure terminal
R1(config)# router isis 0
R1(config-router)# net 49.0001.0000.0000.3333.00
R3(config-router)# is-type level-1-2
R3(config-router)# exit
R3(config)# interface ethernet 0/0
R3(config-if)# ip address 192.168.102.1 255.255.255.0
R3(config-if)# ip router isis 0
R3(config-if)# no shutdown
R3(config-if)# exit
R3(config)# interface ethernet 0/1
R3(config-if)# ip address 192.168.103.1 255.255.255.0
R3(config-if)# ip router isis 0
R3(config-if)# no shutdown
R3(config-if)# exit
R3(config)# interface ethernet 0/2
R3(config-if)# ip address 192.168.2.1 255.255.255.0
R3(config-if)# ip router isis 0
R3(config-if)# no shutdown
R3(config-if)# exit
```