UCLouvain

École polytechnique de Louvain

Flooding in IS-IS protocol

Author: **Arnaud Stoz**
Supervisor: **Olivier Bonaventure**
Readers: **Thomas Wirtgen, Tom Rousseaux**
Academic year 2020–2021
Master [120] in computer science and Engineering

Contents

Acknowledgments

I would like to thank my supervisor, Olivier Bonaventure, for his good advises and his availability. I also would like to thank Thomas Wirtgen for his availability and patience to answer all my questions and help me. Finally, I would like to thank Adrien Allard who was doing his master thesis at the same time, for his support and mutual encouragements

Chapter 1

Introduction

Routing protocols are the foundation of the highly connectivity in today's world. They let the routers exchange information allowing them to select the best routes, based on some criteria, to connect the humankind. There are two different routing protocols: The Interior Gateway Protocols (**IGP**) that are used inside an Autonomous System (**AS**) and the Exterior Gateway protocols (**EGP**) that are used between different ASes. IS-IS [19] is, with OSPF [16], a widely used IGP routing protocol, particularly in datacenters.

More and more people are connected to the internet every day [1], and the 2020 coronavirus crisis brought a lot of people working from home [23] and relying on internet service to connect to their working environment. It is hence essential, in the today's internet, to consider service interruption time with respect to the traffic type. For instance, an interruption of a half-second would be unnoticed in a web page download, annoying when watching your favorite show on any streaming provider, but unacceptable in an important working meeting.

Having a fast failure detection methods and fast network convergence is crucial for an Internet Service Provider (**ISP**). ISPs usally guarantee their service to their enterprise customers in the form of Service Level Agreements (**SLAs**) that specify levels of reliability. Any failure in those SLAs cost money to the ISP [21] and provide a poor user experience for the end user. Another point to reach fast network convergence is to have a fast and efficient flooding of a large amount of routing information.

Because of all these considerations, we will have a look at the flooding capability of IS-IS. We would like to improve these capability by changing the paradigm of IS-IS and using the famous Internet Protocol (**IP**) [18] and Transmission Control Protocol (**TCP**) which have already experienced reliability and retransmission

mechanism. To achieve that, a new TLV is proposed to extend the protocol while keeping backward compatibility with previous version.

Structure of the thesis

This master thesis is organized as follows:

- In Chapter 2 the standard IS-IS protocol operation will be described, see what are the core concepts and how it works.
- In Chapter 3 the methodology to perform the experimentation on the LSPs exchange speed is explained. The different tools used to perform the analyse are introduced. Finally, performances of standard IS-IS protocol in various link capability are analyzed and discussed.
- In Chapter 4 the TCP extension brought to the protocol is discussed. For that the new TLV is described and the behavior of the protocol in different cases is detailed. Performances of this extension are analyzed and compared to the ones from the classical IS-IS protocol.
- Chapter 5 gives some insight of what could be done next.

Chapter 2

IS-IS

The intermediate System to Intermediate System (**IS-IS**) routing protocol is an Interior Gateway Protocol (i.e. it operates inside a single AS), that can be used to carry both IPV4 and IPV6 information but also, due to its large scalability, almost any transport protocol. It becomes the de facto standard for large service provider network backbones (even if OSPF is probably better known by a wider public). IS-IS belongs to the link-state routing protocol class, which means that any router running IS-IS gets an entire view of the network topology. This view is a direct graph with routers being nodes and vertexes being the unidirectional links between those. To compute this entire view of the network topology, IS-IS relies on the famous Dijkstra algorithm [24] to compute the shortest path tree with the router as root.

This chapter will describe the main ideas behind IS-IS and provide some internal knowledge of how the protocol is working.

2.1 IS-IS protocol

IS-IS is very different from other network routing protocols because it runs on Layer2 of the OSI [4] Reference model. This means that, unlike the IP routing protocols like BGP or OSPF, IS-IS does not need any valid interface addressing information to transmit a message. This assumption is discussed and modified when we talk about the TCP solution in Chapter 4.
Running on Layer 2 has others impacts amongst which we can list:

1. IS-IS is totally agnostic about which kind of prefixes it transports in its messages. It can thus transport a lot of various network layer protocols, such as IPV4, IPV4 or even IPX [6].

2. As it runs directly on Layer 2, it can not rely on reliable transmission and on efficient retransmission mechanisms. This has a big impact when it comes to LSP exchange as it is discussed later.

Before digging more deeply into IS-IS, it is important to do a bit of terminology in the OSI model with regard to the TCP/IP model. This terminology is shown in Table 2.1.

OSI	TCP/IP
System	Node
End System	Host
Intermediate System	Router
Circuit	Interface/link
Domain	Autonomous System

Table 2.1: Some terminology between the OSI model and the TCP/IP.

2.1.1 Area

The first element to look at in order to understand how IS-IS work is: how does it structure its network topology?
The OSI model structures its network topology in a distinctive way of how OSPF will structure its network topology. As this is not the main purpose of this work, the part explaining how the area can be migrated or any other subject linked to the notion of area will be let to the curious reader. The Main idea to have in mind is that unlike OSPF, routers in the IS-IS belong only to one area and are not at the junction of several areas, like shown on Figure 2.1. This notion of area directly

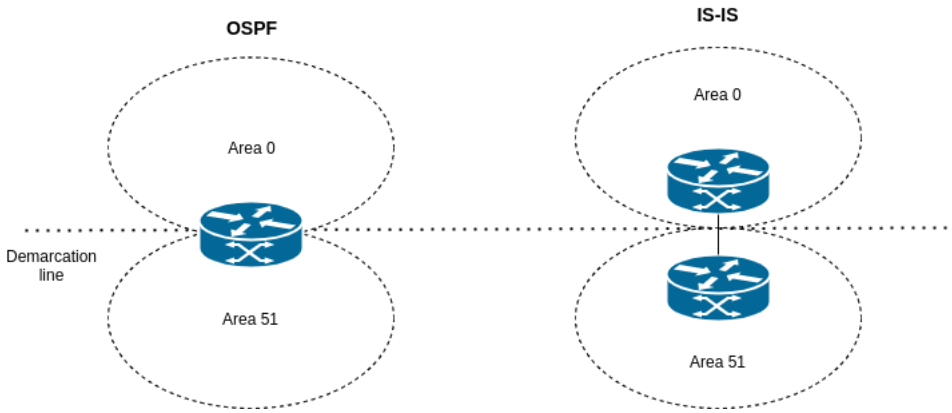

Figure 2.1: The difference of representation between OSPF and IS-IS.

leads to the notion of level which will briefly describe in the next subsection.

2.1.2 Level

As shown on Figure 2.1, it can be seen that, in IS-IS, a router does belong to only one area. This leads to the following big difference with OSPF: in IS-IS, the area-ID does not necessarily have to match in order for an adjacency to come up. The notion of level came so in order to establish adjacency. In fact, a router can be part of three different levels:

1. **Level1**: this type of level is only for in area adjacency.
2. **Level2**: this type of level is only for extra area adjacency.
3. **Level1-2**: this level is of course both level1 and level2

A router can only establish an adjacency with a router having the same level as him. Level1-2 can establish adjacency with both level1 and level2 router. The interesting point about this notion of level, is that each level has its own link state database (**LSDB**). The level1 LSDB will leak its information to the level2 LSDB. However, the level2 LSDB will not leak any information to the level1.

2.1.3 Neighbor discovery and handshaking

All routing protocols include a method of automatic neighbor discovery that enables a router to determine if there are any other adjacent routers running the same routing protocol.
IS-IS performs like any other routing protocol, it uses Hello messages to discover its neighbors and perform handshaking. This is done by what IS-IS calls Intermediate System to Intermediate System Hello (**IIH**) messages. As seen before, a router can be a two type of topology. IS-IS needs to use two types of IIH message: one Hello for the level1 adjacency and one Hello for the level2 adjacency.
IS-IS also supports two different circuit types: point-to-point(**p2p**) and broadcast (**LAN**) circuits. Once again, there is a dedicated Hello message for point-to-point circuits and another one for broadcast circuits. In theory, there should be 4 hello messages types. However, in ISO 10589 [11], there was some concern about having two hello types for the p2p circuit because it could take too much bandwidth. IS-IS is hence *optimized* for p2p circuits and only uses one hello types for both levels.

Now that the hello message has been described, we can focus on how handshaking is performed. In the IS-IS specification there are two general ways of handshaking:

- 2-way handshake
- 3-way handshake

2-way handshake

In a 2-way handshake, as the name let it think, there is only need for 2 **IIH** to declare an adjacency up. Figure 2.2 shows what happen during a 2-way handshake. A **IIH** message is sent to router B. When router B receives this IIH message, it declares the adjacency with router A up. Once router A receives a **IIH** from B, it will declare the adjacency with router B up. The important point here is that Router A does not know if the Hello message was a response from router B or if its Hello message was lost on the wire.

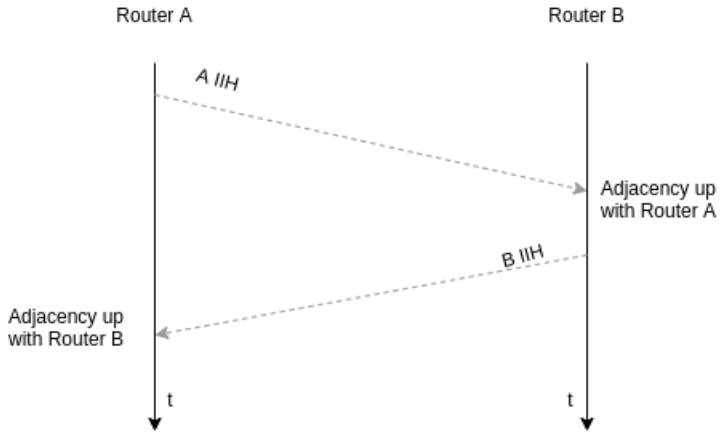

Figure 2.2: 2-way handshake from the view point of router A.

3-way handshake

In the 3-way handshake, there is a need of 3 Hello messages in order to declare the adjacency between 2 routers up. Figure 2.3 shows how a 3-way handshake work. First Router A sends a **IIH** message to router B. Router B then responds with an **IIH** which carries an indication that this hello message is a response to a previously sent hello message. This is achieved by mentioning explicitly Router A in the TLV of the hello message. Finally, Router A sends a third hello message to confirm to Router B that it has also seen it. On LANs the 3-way handshake is generally used and on the p2p link, the original ISO 10589 specification proposed just a 2-way handshake. Nevertheless, through implementation and deployment experience, several scenarios are known where the use of a 2-way handshake cause IS-IS to get blind spotted. But those special scenarios are out of the scope of this work. The important point is to know that there exist these 2 types of hello, because they will be discussed in later.

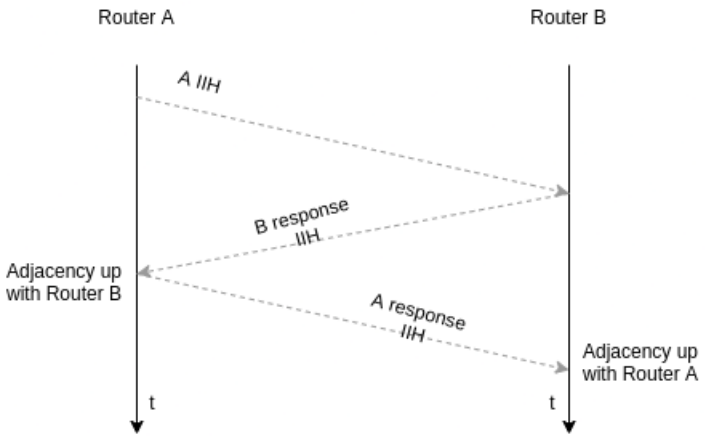

Figure 2.3: 3-way handshake from the view point of router A

2.1.4 LAN Circuit

As said before, IS-IS can run on two different types of circuits: a peer to peer circuit and a LAN circuit. In the case of a p2p circuit, there is nothing special as each circuit is connected to only one neighbor.
In the case of a LAN circuit, this is a bit more complicated as on this circuit the router can have multiple neighbors. Due to this large number of routers on the LAN, there are several aspects of the protocol to care about. First, if there is a large number of speakers on the LAN there is a lot of Hellos to process. If those Hellos arrive in a very short period, this could overwhelm routers and impeach them to process routing information. In order to avoid this hello synchronization, ISO 10589 mandates to jitter timers for scheduling Hellos.
The other problem that arises on LANs with N routers is that each time there is a route modification like a new router $N + 1$ getting on the LAN, all the N other routers that have been on the LAN previously have to update their LSPs to list the adjacency to the new router. This results in a massive LSP update storm because *all* the routers on the LAN need to tell the network the adjacency changes. The solution to this problem is changing the representation of the LAN in the link-state database. The LAN is represented by so-called pseudonodes.

Pseudonode representation

The idea of pseudonodes is to give the LAN a nodal representation in the link-state database. On each LAN circuit, a Designated Intermediate System (**DIS**) is elected. The DIS is a router among the IS-IS routers on the LAN, which has, additionally to its normal duties, the purpose of representing the LAN in the LSDB. This changes the representation in the LSDB from a full mesh topology to a star topology with

the pseudo node in the middle as shown on Figure 2.4. The method to elect the

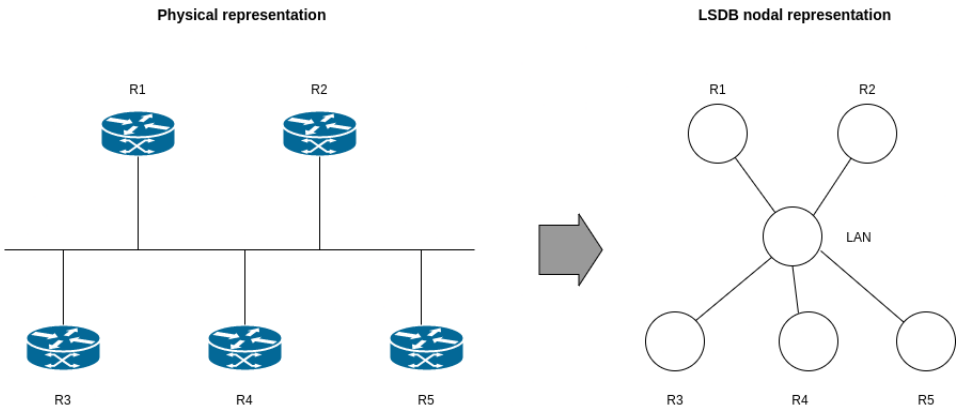

Figure 2.4: The LSDB representation of the DIS

DIS and all the issues that can arrive are beyond the scope of this work, but the curious reader can have a look at *The complete IS-IS routing protocol* book [9].

2.1.5 Link-state protocol data unit

As said before, IS-IS, like OSPF, belongs to the link-state routing protocol. Which, in opposite to distance vector routing, distribute both their IP reachability and topological view far beyond their adjacent neighbors, ultimately flooding this information to all routers in an area.
To keep the reachability information in the network up-to-date, link state protocol requires a basic set of function used to *originate*, *distribute*, and finally *time-out* topology information. In IS-IS vocabulary, this piece of topology information is encoded in a link-state protocol data unit(**LSP**).
The Figure 2.5 shows the structure of a link-state PDU. The ISIS common header is the header fixed for all types of IS-IS packet and can be found in Appendix B. On the LSP header we can see the following important field:

- **PDU length**: this gives the total length of the PDU, including the common header, the LSP header and the TLV section.
- **Lifetime**: this gives the total time the LSP must be considered valid.
- **LSP-ID**: the id of the LSP as the name suggests it.
- **Sequence Number**: the sequence number is used to find the most recent LSP between two same LSP-ID.

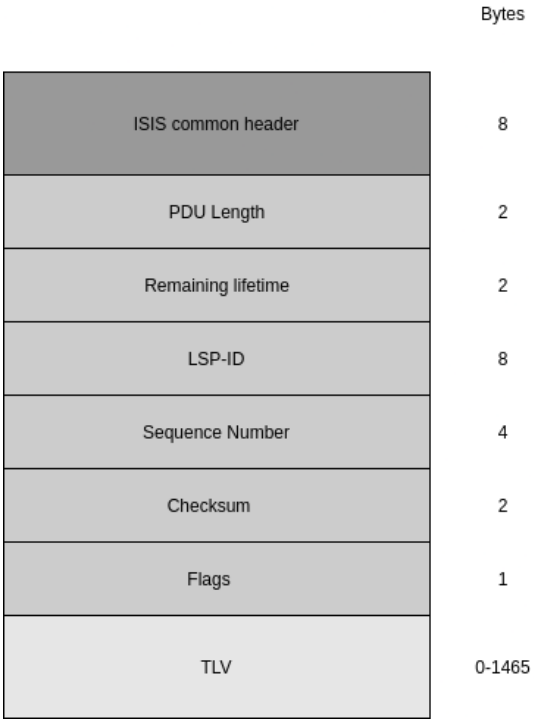

Figure 2.5: the format of a link state PDU.

- **Checksum**: the checksum used to check that the LSP was not corrupted as IS-IS run directly above layer 2.

The last byte of the LSP header is a bunch of flags which will not be discussed here as it does not represent too much interest for the rest of this work.

2.1.6 Sequence Number Packet

There is still one type of packet to discuss. This packet type is called Sequence Number Packet (**SNP**).
There are 2 different types of SNP packet:

- The Partial Sequence Number Packet (**PSNP**).
- The Complete Sequence Number Packet (**CSNP**).

PSNP

The PSNP has 2 different uses depending if IS-IS is running on a p2p circuit or on a broadcast circuit.

On the p2p circuit, PSNP is used to acknowledge the LSP received, as IS-IS runs above layer 2 and thus can not be sure about the reliability of the transmission. The retransmission mechanism of IS-IS is explained in Section 2.2.
On the broadcast circuit, the PSNP are used to request LSP from the DIS, after having received a CSNP. More details about the use of PSNP on a broadcast circuit is provided in the following.

CSNP

The CSNP, like the PSNP, is used differently depending on which circuit IS-IS is launched. On a p2p circuit, CSNP are only sent once, even if as this will be shown later this is not always the case, when the circuit is declared up. The CSNP is a packet which contains all headers of the LSP contained in the LSDB of the router. This is how routers exchange which route/LSP they have knowledge of.
In the case of a LAN circuit, unlike the p2p circuit, the CSNP are exchanged periodically. This is only sent by the DIS. Each router can compare the LSP found in the CSNP with the LSP in its LSDB. Three different situations can arise:

1. LSPs in the CSNP are older than LSPs in the LSDB. In this case the router will simply flood LSPs it has in the LSDB

2. LSPs in the CSNP are the same as LSPs in the LSDB. In this case nothing append, all is good.

3. LSPs in the CSNP are newer than LSPs in the LSDB. Then the router will send a PSNP to request newer LSPs.

2.1.7 TLV

As already mentioned before, when showing the different packet structures, there is always a TLV section. TLV stands for Type Length Value. The TLV section makes IS-IS an extensible protocol. The structure of the TLV is shown in Figure 2.6. The

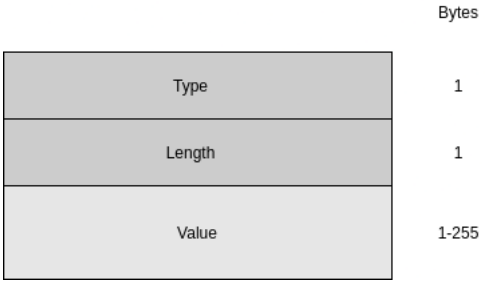

Figure 2.6: Structure of the TLV

Type field is used to tell which type of information the TLV carries. This is a 1 byte field which means that there can be at most 255 different TLV types. The Length field indicates the size of the Value field. Again this is a 1 byte field which means that the maximum size of the value field is 255 bytes. Finally, the Value field is the place where the information is stored. For example, it can be the IP reachability in the case of LSP. An important point to notice is that the TLV type is not linked to any IS-IS packet, which means that each TLV can theoretically be present in all the three IS-IS packets, which are:

- IIHs (hello message)
- SNPs (sequence number PDUs)
- LSPs (link-state PDUs)

It is finally possible to use sub-TLVs. This is a TLV encapsulated in a TLV. Theoretically a dedicated TLV could be used as well for new message elements. However, this would quickly exhaust the TLV space. The structure of a sub-TLV is exactly the same as a TLV, with a Type, Length and value field.

2.2 Retransmission mechanism in IS-IS

Ensuring a good goodput between two routers is the responsability of layer 4 in the OSI model with the famous Transmission Control Protocol (**TCP**). However, as previously said, IS-IS runs directly above the layer 2 and therefore can not benefit from TCP mechanisms. So, IS-IS must implement and run its own retransmission mechanism to be sure that LSPs are not lost in the wire and arrive at destination. Of course, for an IGP protocol, the LSP exchange must be one hundred percent reliable, no loss of any LSP information is permitted. Otherwise, it is possible that some entire part of internet might be unreachable for you.
But on the other hand, this exchange must also be as fast as possible, because as an Internet Service Provider (**ISP**), the longer the convergence of your network takes, the less money and the more angry customer you make.
In IS-IS, the reliability of LSP exchange is not achieved in the same way depending if IS-IS is run on a p2p circuit or on a LAN circuit.
In order for the router to keep a better track of which LSP needs to be retransmitted or which LSP needs to be acknowledged or requested, it uses 2 flags, which are used purely internally and are not part of the LSP header or whatever.

- The Send Routing Message flag (**SRM**). This flag is set when there is a need to flood the LSP.

- The Send Sequence Number flag (**SSN**). This flag is used in one of the two following case depending again on which circuit is IS-IS
  - To acknowledge LSPs received on p2p link.
  - To request LSP on LAN link.

2.2.1 Reliability in p2p

In p2p, the reliability is done by acknowledging the LSP received. As explained before, this is achieved by sending a PSNP containing the header of the LSP received. When a router needs to send a LSP on a p2p link, it set the SRM flag. While no PSNP is received for this LSP, the SRM flag remains and the LSP will be retransmitted each five second. Once the PSNP for this LSP is received, the SRM flag is cleared and the LSP is considered as well received.

2.2.2 Reliability in LAN

On a LAN circuit, unlike on a p2p circuit, there is no need of acknowledging a LSP received as CSNP are sent periodically.
If a router does not have a LSP which is advertised in the CSNP, it sends a PSNP requesting this LSP as explained before. But if the CSNP, the PSNP or the LSP is/are lost, there is no way for the sender to notice this loss. The only way to recover this loss is to wait for the next CSNP and to resend the PSNP requesting the LSP and then hopefully receive the LSP.

Chapter 3

IS-IS Experimentation

As mentioned before, the goal of this work is to improve the speed at which the LSPs are exchanged and hence improve the speed at which the network could converge. The experimentation and the results shown in this chapter aims at highlighting some weaknesses of the IS-IS protocol and brings a performance comparison with the TCP extension discussed in Chapter 4
This chapter details the methodology for measuring the LSPs exchanges between routers. In Section 3.1, the experimental setup is explained. As all the IS-IS software used was the one from FRRouting [12], we discuss a bit in Section 3.2 the implementation and how a custom LSDB can be totally controlled and loaded in the daemon. Section 3.3 describes how IS-IS normally synchronizes LSDB after the adjacency is up.

3.1 Experimentation setup

The experimental setup used to make measurements is relatively simple. It is composed of 2 machines interconnected by a 10 Gbits/sec Ethernet cable. They both have an Intel Xeon X3440 CPU with 8 cores running at 2.53GHz and 8GB of RAM. One runs with *Debian GNU/Linux 9* while the other runs with *Debian GNU/Linux 10*. The topology is then quite simple, as shown on Figure 3.1.
The routers are in the same area and thus running in level1. One router acts like a master, which means this is the router which loads the LSDB. The second router acts asa slave which means this is the one which needs to synchronize his LSDB. Both routers are configured to be L1 routers.

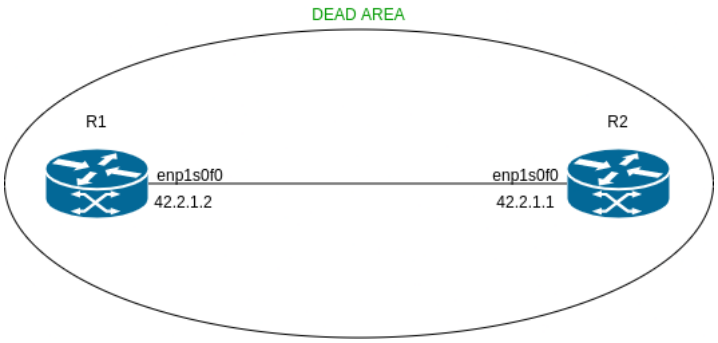

Figure 3.1: The topology used for the experimentation. R1 is running Debian GNU/Linux 9 and R2 Debian GNU/Linux 10

3.2 FRRouting implementation

All the work and the result presented in this thesis are done using an IS-IS implementation. As the goal was not to implement the IS-IS from scratch, the IS-IS implementation used in this work was the one from FRRouting.
FRRouting (**FRR**) is a free and open source internet routing protocol suite for Linux and Unix platforms. It contains daemons for BGP, OSPF, RIP, IS-IS, and many others. It was forked from Quagga [22], another routing protocol suite for Linux. The use of FRR is motivated by the fact that the documentation is pretty good, it is actively maintained and the community is very reactive.

3.2.1 Breaking up the IS-IS implementation

Before going further, we break up the IS-IS implementation to bring out all keys features of the protocol. The following list enumerates the main operations of the protocol. As this is a brief overview, a lot of secondary features are not explained as it's not the purpose of this work. In this list, the main features are linked to the function name in the FRRouting implementation.

- Circuit up (function *isis_circuit_up*): this mechanism is used to setup the circuit which implies setting the timer for the CSNP, preparing the sending/receiving buffer....
- Process PDU (function *isis_handle_pdu*): Once something is received by the IS-IS process, it needs to be analyzed to determine if it is a IIH, a SNP or an LSP and thus be processed accordingly.

These are the main features on which this work relies on and modifies to load a custom LSDB and process the PDU.

3.2.2 Load custom LSDB

The first step to make measurements was to be able to have a full control on the LSDB. We intend to do several measurements with different numbers of LSPs present in the LSDB, and also potentially generate different sizes of LSPs.
To do so, the implementation has been modified to load a custom LSDB in a json format. The LSDB is loaded when the circuit is set up (in function *isis_circuit_up*).
A new option was added to the FRR implementation in order to load a custom LSDB in a json form as shown on listing 3.1.

Listing 3.1: Load a custom LSDB

```
1 ./isisd -L lsdb.json
2 or
3 ./isisd --lsdb_file lsdb.json
```

The json file is a list of json objects representing an LSP. The LSP json object contains itself 2 json objects, *hdr* and *tlv*. The *hdr* object contains all the fields of a LSP header described in Section 2.1.5. The *tlv* object contains only a subset of the possible tlvs, the most important being *protocol supported* tlv and *extended ip reach* tlv. The complete format of the json object can be found in Appendix A.
A tool was written in order to generate the LSDB in the correct json format and simply callingd *generate_lsdb.py*. It generates a LSDB with a given number of LSP.
For now this tool only generates LSDB for a ring topology. Which means that each router has two neighbors and thus each LSP contains 2 "extended reach" TLVs. This implies that, at present, a LSP generated by this tool has a size of 99 bytes. The decision to only generate LSPs for a ring topology is motivated by the fragmentation of the LSP which leads to a much more complicated scheme to generate a LSDB when there is fragmentation.
There are also some other limitations to this tool. All the LSPs generated belong to the same Area.

3.3 IS-IS LSDB synchronization

This section describes how the LSDB synchronization works in IS-IS. The mechanism is changing depending upon IS-IS is running on a p2p or a broadcast circuit. The following explanation will take place just after the adjacency is declared up, that means just after both routers have seen each other.

3.3.1 Peer to peer synchronizing LSDB

The synchronization on the p2p circuit has three distinct phases.

1. Each router sends a CSNP containing all the LSP headers it has in its LSDB.. This CSNP is not sent directly after the adjacency is declared up, but each router jitters a 5-seconds timer by 25 percent before sending the CSNP. Jittering by 25 percent means that it calculates a random number between 75-100 percent of the underlying timer. This is done to avoid a synchronization in the transmission of CSNPs resulting in traffic spikes between the two routers.

2. Once the router has received the other CSNP, it can simply compute the differences between CSNPs received and its own database and then decide if it needs to send some LSP. This is achieved by setting up the SRM flag on missing LSPs in the CSNP.

3. On the p2p circuit, the flooding needs to be reliable. This means that until the router receives a PSNP to acknowledge the LSP it has sent, the SRM flag remains enabled and every 5 seconds the router resends all the unacknowledged LSPs with the SRM flag.

Once all the SRM flags have been cleared, which means that every LSP had been acknowledged, there is no further need to send other CSNPs. Once both LSDBs have been synchronized, the only things that could happen is a new LSP coming from the outside. This LSP is thus flooded until the PSNP to acknowledge this LSP is received.

3.3.2 LAN synchronizing LSDB

Synchronizing databases on broadcast LAN circuits is quite different from p2p circuits. This is due to the fact that in a broadcast circuits there is a need to elect a Designated Intermediate System (**DIS**) as explained before. The DIS has a key role in the synchronization of link-state databases on LANs. Periodically (typically every 10 seconds), the DIS broadcasts a CSNP of its own link-state database, which is received by all routers on the LAN.
Once the CSNP is received by one of the routers, there are 3 different possible cases:

1. LSP advertised by the CSNP is older than the one stored in the LSDB of the router. In this case, the action taken is simple. As it appears that the DIS is not up to date, the router informs the DIS about the latest version of the LSP by re-flooding it onto the LAN. The main difference here with the

p2p is that there is no need to acknowledge the LSP. This is what is called *implicit acknowledgment*. If the LSP was lost, we only have to wait for the following CSNP to notice that the LSP is not up to date.

2. LSP advertised by the CSNP is newer than the one in the LSDB. In this case, the router has to tell the DIS that its LSDB is out of sync by setting up the SSN flag which triggers a PSNP to be sent to the DIS to request the new version of the outdated LSP. Once the DIS receives the PSNP, it only re-floods the LSP on the LAN.

3. LSP advertised by the CSNP is new or unknown. This case is very similar to the case of the newer LSP, the router sends a PSNP requesting the LSP to the DIS.

We can directly notice it is the rate of sent CSNPs which gives the time to recover from a loss. Increasing too much the sending rate of the CSNPs is not ideal because this increases the use of the bandwidth as well as consumes more CPU power on the router for processing those CSNPs. There is thus a trade-off to perform between having a quick time to detect and recover from LSP losses without overwhelming the network.

3.4 Monitoring IS-IS LSP exchange

Monitoring routing protocols is and has always been essential. Recording data about the flow of the protocol can for example help network operators to understand the weaknesses/bottlenecks of their network and allow them to configure it in a better way. In the case of IS-IS, there is few technique to monitor the flooding of the LSP [13]. Or at least, no mechanism directly implemented in the FRRouting binary.
The interesting point here is to measure the time for an entire LSDB of hundreds of LSPs to be loaded and synchronized between two routers, with one LSDB being empty at the beginning. We then capture the LSP exchange process with tcpdump running on the interface of both routers. The result of this capture is analysed is two distinct phases:

1. Using tshark [3] to parse the pcap and get the output in CSV format which is much easier to parse with python. This CSV file contains each LSP-id and the time at which they were captured on the wire.

2. Using a tool written in python which takes the CSV output from thsark and perform statistics on LSPs. The statistics here imply the 3 following points of interest:

(a) The number of LSPs which were not retransmitted. This is called **one sent LSPs** in the rest of this work.

(b) The time needed to exchange all LSPs.

(c) The sending pattern of the LSPs exchange.

3.5 Result

We can now have a look at how IS-IS behaves when he needs to flood LSP to realize a link-state database synchronization. Experimentations were made for a few different link capabilities :

1. No modification on the link, it performs under normal condition, this is called the *no delay situation* thereafter.
2. The bandwidth of the link is modified.
3. The MTU of the link is modified.
4. Some delay is added on the link.
5. Some loss is added on the link.

3.5.1 No delay

In this case, no modification was made on the link. The link has a MTU of 1500 bytes. We can first have a closer look at the time needed to flood all the LSPs between the two routers. The result is shown on Figure 3.2. One important things about this result is that the time shown is the time starting when the first LSP is seen on the wire. This figure shows the mean time to flood from 100 to 800 LSPs between the two routers. This mean is calculated on a 10 run repetition.
There are some observations we can make on this figure. The first thing we can directly notice is that despite some outliers, which are due to variation either on the link or in the cpu usage of routers, the time needed to flood the LSPs is extremely constant between the different runs.
The time needed to flood LSPs increases with a larger LSDB. This behaviour was expected.
However, there is something surprising here. The specification tells that if a LSP is not acknowledged, the LSP is sent back five seconds later. Thus, we expected to see stages by multiples of five. That is obviously not the case here as we clearly see the five seconds delay between 100 LSPs and 200 LSPs but not after anymore. Lets go deeper into why we do not see the five seconds delay between every

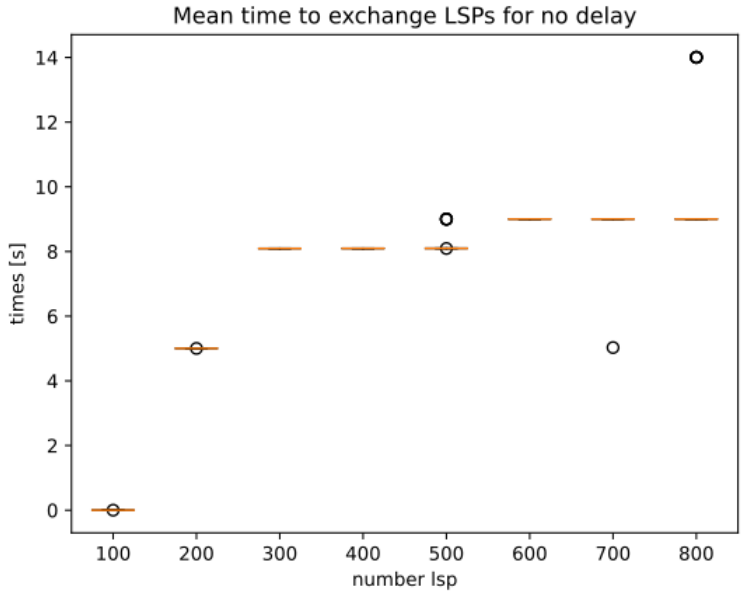

Figure 3.2: boxplot of the mean time for no delay

increasing stages.
For this, we need to recall what can trigger the transmission of LSPs. As explained before, two events can trigger the transmission of LSPs:

1. A CSNP with outdated LSP.

2. A PSNP with an outdated LSP in it which implies de facto that there was no acknowledgement for this LSP.

Let's recall that for this measurement we are on a p2p circuit, so normally, as the IS-IS specification says, CSNP must be sent only one time when the adjacency is declared up, and then PSNPs must be used to acknowledge LSPs.
In order to find what happens, we will take a look at the pcap file of the test. By analysing it with wireshark and taking a closer look at the CSNP packet, we can notice that, unlike what the specification is saying, more than one CSNP is sent. This points out that the FRRouting implementation does not strictly follow the specification concerning the p2p circuit.
We can trace exactly what happen when synchronizing the LSDB and why we do not clearly see the five seconds stages.

1. Once the adjacency is declared up between the two routers, they send CSNP to each other.

2. Instead of letting the master router calculate which LSPs are missing in the slave's LSDB, the slave inserts all the LSP contained in the master's CSNP inside its LSDB with the seqnum set to 0.

3. A few seconds later, the slave sends a CSNP of its LSDB with the LSPs of the master inserted but with all their seqnums set to 0. Once the master has received the CSNP, it notices that all the LSPs are out of date and starts sending all the outdated LSPs.

4. Some LSPs are lost and not acknowledged -we will see that a bit later- and the five seconds timer expires. The master resends all the LSPs which were acknowledged.

5. A new CSNP is sent by the slave, this one acknowledges some LSPs but also lets the master know that some are still out dated and will then reflood them.

To explain that we have not this five second stages, we need to have a look at the implementation proposed for IS-IS.
In fact, FRRouting does not acknowledge directly a LSP received. Instead, it setsthe SSN flag on it. There is an interval which can be set for sending a PSNP. In FRRouting this interval is between 1 second and 120 seconds. This implies that if the timer exceeds 5 seconds, IS-IS will perform a retransmission even if it is not necessary. This is a important point to pay attention to when we do the IS-IS configuration file, even if the default value is 1 second. The point when IS-IS acknowledges directly a LSP send will be discussed later in Section 3.5.2.
This explains why we do not see some clear steps five by five seconds.It does however not yet explain why sending more LSPs takes more time. For this we need to take a look at the sending pattern of LSPs. This pattern is shown for four different numbers of LSPs on Figure 3.3.
There are several observations which can be made from this figure.

1. The first element to be noticed, is that for 100 LSPs, there is absolutely no loss and all the LSPs are very quickly flooded.

2. The second thing that can be noticed is that all the LSPs are sent almost at the same time, they are sent by burst and this, totally independently of the number of LSPs.

3. Finally, we clearly see the pattern of the five seconds stages for 200 LSPs which matches with the behavior we expected but see that this pattern is not respected anymore, this is when the CSNP are sent as explained before.

The sending pattern confirms what we said before with CSNP, but we still don't know why there is so much loss on the LSPs sent. For that, the second observation

is very useful. LSPs are sent in burst, which means that the amount of LSPs can totally overwhelm the router, the sender does not care about it, it just sends every LSPs of its LSDB in almost no time.
There is also another factot that impacts this bad result. To find that out, we need to go a bit deeper on how IS-IS is implemented and more precisely which socket it uses.

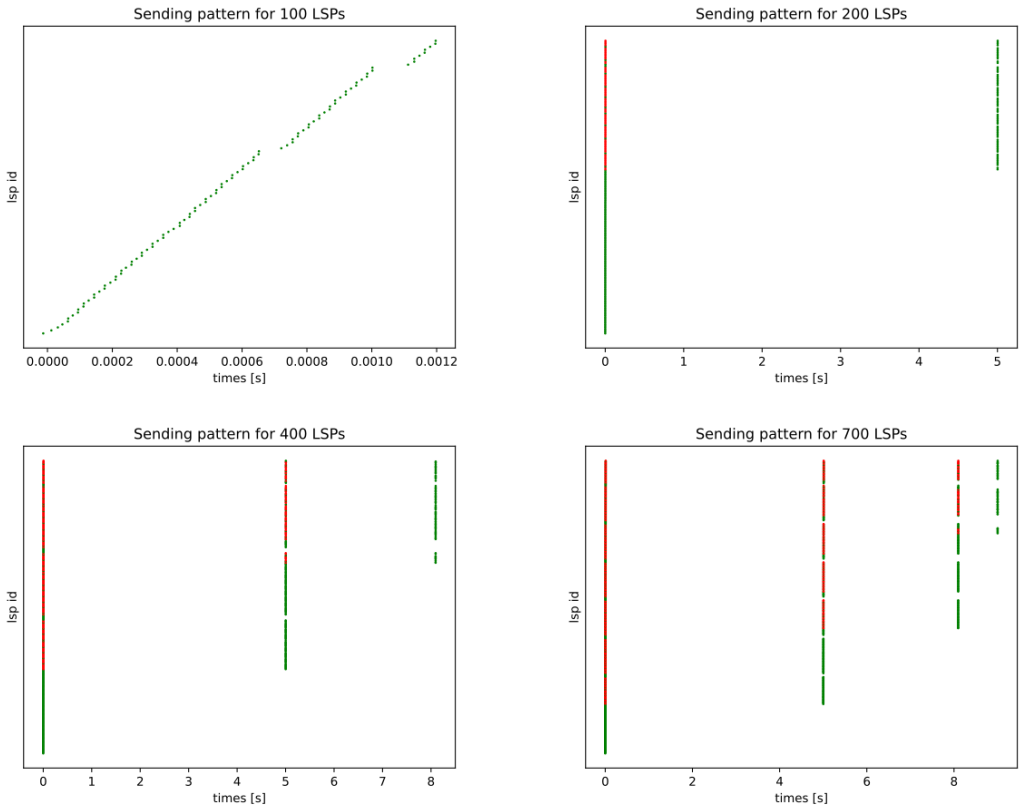

Figure 3.3: Sending pattern for four different number of LSP.

Discussion

As IS-IS runs directly above layer 2 of the OSI model, it can not use the well known *DATAGRAM* or *STREAM* socket. Instead, it uses a *RAW* socket. This socket, unlike the *STREAM/DATAGRAM* socket, lets the user directly manipulate the header and the trailer of the network and the transport layer (layer 3 and 4 of the OSI model).
One first idea that might come to mind is that the buffer size of the raw socket is too small to handle the large amount of packet incoming. In the case of this work,

the buffer size of the raw socket was the standard one which can be found in the file */proc/sys/net/core/rmem_default*. On the machine running the experimentation, this value was of $212Kbytes$.
A bit of mathematics show us that this value should be enough to handle all LSPs for the number used in this experimentation. As the size of LSP in this case are $102bytes$ (we need to add the LLC header to the length of the LSP), the max size received is $81,6Kbytes$, which is far below the maximum size of the buffer. To validate this calculation, we used the function *getsockopt* from the socket API [20] in order to retrieve the statistic of the socket, and see if some packets are dropped. As expected no packet drop was shown on the socket. We hence need to search in another direction.
Another possibility might be that the ring buffer of the NIC was too small. But this hypothesis was quickly abandoned as if it was the case the tcpdump trace should not show LSPs.
So for now there is no precise explanation on this weird behavior.

3.5.2 Acknowledge directly

Now that we have seen that the FRRouting implementation is a bit different from what we expected from the IS-IS specification, let's have a look at what happens if we stick a bit more to the actual IS-IS specification. We will again here discuss the case where there is no delay or anything else, like the bandwith or the MTU, changed on the interface.
In this case, LSPs received are directly acknowledged by the receiver. The mean time for the LSPs exchange are shown in Figure 3.4. The first element to notice is that we see better the five seconds stages even if it's not exactly a five seconds stages everywhere. We can directly see that it is less efficient than the FRRouting standard implementation as for 800 LSPs the FRRouting standard implementation took more or less 9 seconds to be synchronized while the acknowledge directly implementation took more than 17 seconds.
This can be explained by the fact that the router having to acknowledge directly the LSP, he has more work to do and is potentially more easily overwhelmed. Indeed, instead of processing a lot of LSPs and only setting up a flag for each of them, he has to craft for every LSP processed the corresponding PSNP. This PSNP will in addition contain only one LSP header which is obviously not optimal.
The sending pattern is of course modified accordingly with this new mean time.

3.5.3 Number of LSPs send one time

One other parameter interesting to compare between the two different implementations is the number of LSPs which are sent only once. If the receiving router is

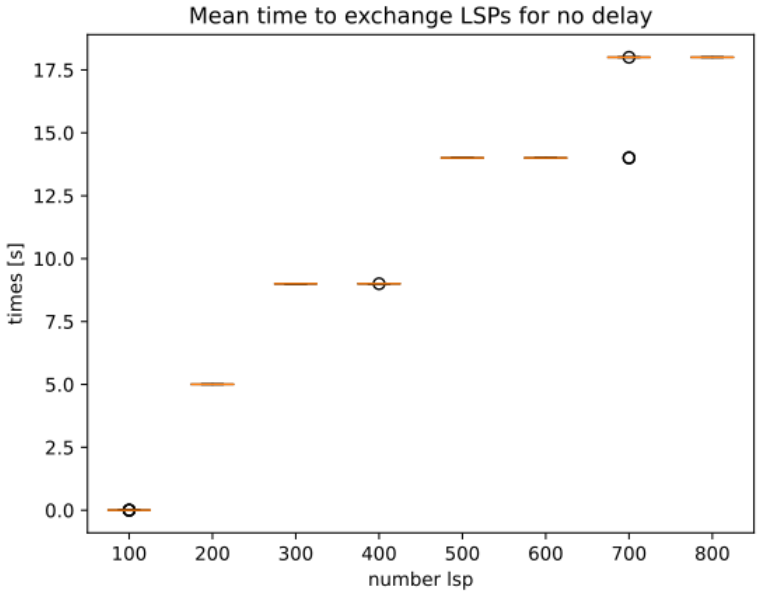

Figure 3.4: Mean time depending on the number of LSPs when acknowledge directly

overwhelmed, there must be a plateau at a given number of LSPs. The graph for the two implementations are shown in Figure 3.5. Some elements can be analysed from those two graphs.

- As expected for the case where IS-IS directly acknowledges a LSPs, we can see that we reach a plateau for 500 LSPs. This confirms the hypothesis formulated before.As the router has to do more work to build the PSNP directly in response and send it, he is more easily overwhelmed.
- In opposite, even if there are some LSPs which are not processed as we have seen before, in the standard implementation of FRR, the number of LSPs sent once increases with the number of LSPs sent.

In fact, the observation when every LSP is acknowledged directly is exactly what is expected but the one from the standard implementation of FRR is surprising. The packet losses are not related to the router itself because it is still able to process many LSPs. The Figure 3.5 confirms this assumption as there is no bottleneck, even for 275 LSPs.
This behavior is suprising and as said before, the cause of it is not totally determined.

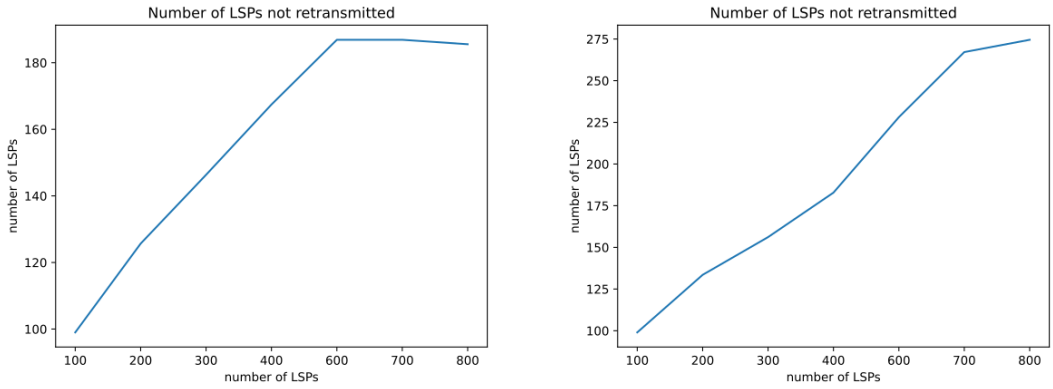

Figure 3.5: Number of LSP sent only one time for the IS-IS specification (left) and the FRRouting implementation (right)

3.5.4 Link Modification

The behavior of IS-IS was also tested in various numbers of scenarios with different capabilities on the link.
The modification on the link implies:

- A bandwidth modification (from $100kB/sec$ to the normal capabilities of the link $10GB/sec$).
- A MTU modification. The modification was made to test from a MTU of 200 to the max value ethernet can handle (without the jumbo frame) of 1500.
- Adding some delay on the link starting from $10ms$ delay to $100ms$ delay.
- Adding 1% and 5% loss on the link.

Bandwidth and MTU modification

Suprisingly, those modifications of the bandwidth and the MTU had almost no impact on the IS-IS performance. The impact of a $100kB/sec$ bandwidth and a 500 MTU is shown on Figure 3.6. As we can see, the mean time to exchange LSP is not really impacted by those modifications.
The only noticeable impact these tests revealed were the discovery of a bug in the IS-IS implementation for a MTU of 200. The bug consists of an infinite loop in the LSP fragmentation which leads to an out of memory and thus a SIGSEV. Nevertheless, this is not a critical bug as this is very rare to have such low MTUs.

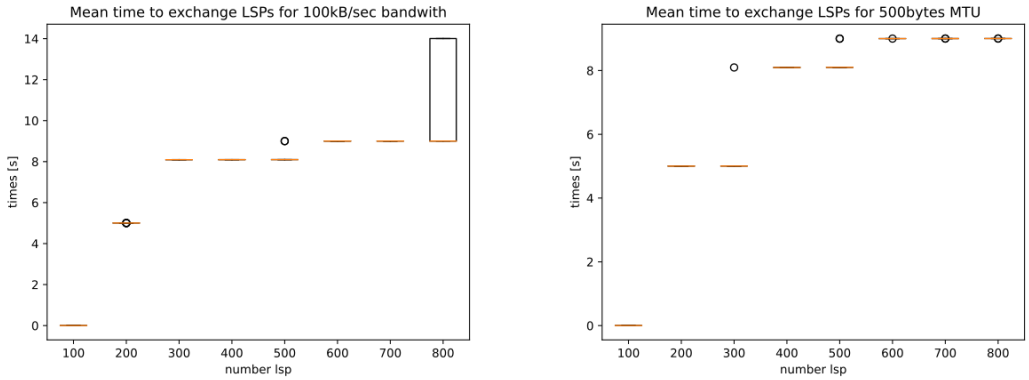

Figure 3.6: Mean time of LSP exchange for a $100kB/sec$ bandwidth and a MTU of 500

Delay

In the case of adding delay there is of course a small impact. There is no difference in the mean time for the different delay, this is why the $30ms$ is only shown here. In the case of the Delay, the mean time is naturally increased with respect to the delay set but the pattern of the LSP send is not very impacted by a delay of $30ms$ as shown on Figure 3.7. We can see that the mean time is slightly increased but

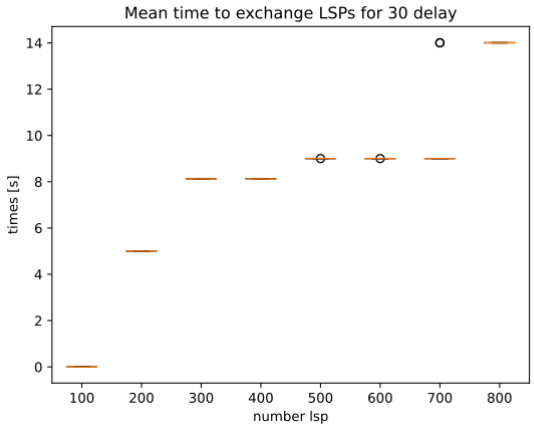

Figure 3.7: Mean time for $30ms$ delay

for 800 LSPs where the mean time to synchronize the LSDB is increased a lot compared to the normal case.

Loss

In the case of adding loss on the link, there is also a small impact. This is naturally

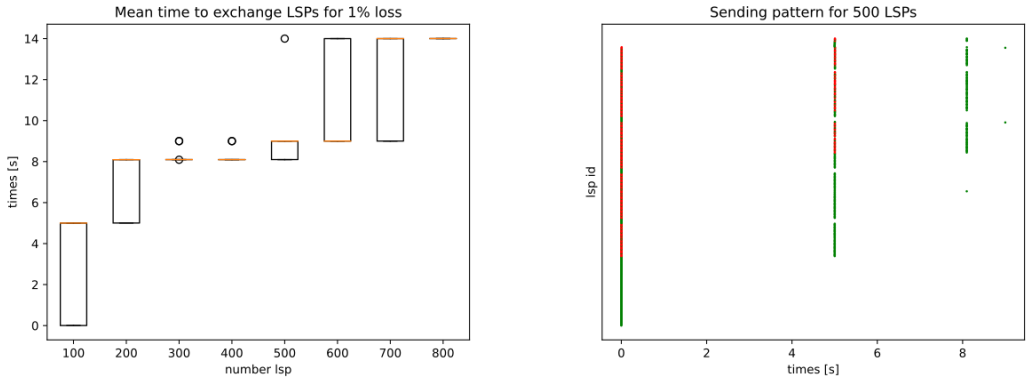

Figure 3.8: Mean of the synchronization and sending pattern for 1% loss.

due to the retransmission mechanism implemented by IS-IS. The impact on the mean time to synchronize the LSDB for 1% loss and the sending pattern is shown on Figure 3.8
We can directly notice that the mean time to synchronize LSDB increases and is also less concentrated than before due to the loss of LSPs. The pattern of sending LSPs is also a bit impacted by the lost as we could expect. The conclusion is that in case of lost, the retransmission mechanism of IS-IS is not very efficient as it increases the time needed to synchronize the LSBD.

3.5.5 Broadcast

We can have a quick look on the result when IS-IS is on a broadcast circuit, in order to better see the difference between the two methods to synchronize databases. The mean time needed to synchronize the LSDB when routers are on broadcast circuit is shown on Figure 3.9. We can directly notice that the time needed to synchronize the LSDB is far more longer. This is due to the fact that on broadcast circuit, the synchronizing is done only when CSNP is received as explained earlier. When running the test, the standard value for the sending of CSNP was kept with default value of 10 seconds. In FRR, this 10 seconds is also jittered by 25%. This explains why we have these different stages which are a lot more important than on p2p circuit.

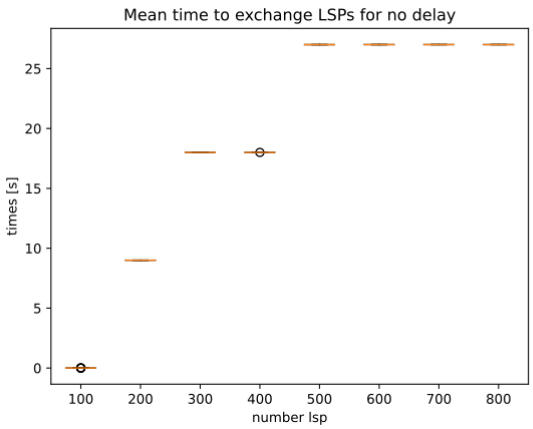

Figure 3.9: Mean time to synchronize LSDB on broadcast circuit.

3.6 Conclusion

In this chapter we analysed the performance of LSDB synchronization between two routers. We saw that the synchronization is not very fast and the time needed depends on which circuit IS-IS is running (p2p or broadcast). This is mainly due to the poor retransmission mechanism of IS-IS.
We also discovered that FRR does not strictly implement the IS-IS specification on the p2p circuit but it increases the synchronization time.
One final observation is that modification on the link does not, or only slightly modifies the time needed to synchronize LSDB and the pattern of sending LSPs. The main point to improve is hence the retransmission mechanism and not overwhelm the receiver.

Chapter 4

The TCP extension

We have seen some measurements of how IS-IS works and its efficiency. The goal is now to try to improve the speed at which the LSPs are exchanged in order for an ISP to have faster network convergence. There are already some propositions to increase the speed and the reliability of the exchange such as the one proposed by B. Decraene et al. [5] or the dynamic flooding on Dense graph [10]. The solution imagined by Decraene et al. proposes a few new TLVs in order to implement a better loss detection and congestion control. It aims at bringing some TCP mechanisms directly into IS-IS.
The dynamic flooding Dense graph aims to reduce the number of LSPs exchanged. The solution proposed here is a bit different from the two cited before. We propose to modify the IS-IS protocol such that it exchanges LSPs over a TCP session instead of operating directly on OSI layer 2. Modifications on the protocol are discussed in Section 4.1. Some specific cases are discussed in Sections 4.2 and 4.3. In Sections 4.4 and 4.5, the impact of this TCP extension on the time needed to exchange LSPs and on the latency are analysed.

4.1 IS-IS protocol modification

In order to use TCP in IS-IS there are few modifications to bring to the protocol. As IS-IS runs above layer 2, there is technically no need for the interface to have an IP address. This is of course something which needs to be modified in order to be able to open a TCP connection between two routers. The first modifications is that on every interfaces on which a TCP connection needs to be brought up, the interface requires an IP (IPV4, IPV6 or both) address.
Second, the hello message must be modified to announce it supports TCP to exchange LSPs. This is discussed in Subsection 4.1.1

4.1.1 Hello modification

To inform the neighbor that TCP is supported by this router on this circuit, a new TLV needs to be introduced. This TLV is called *TCP availability advertising* with the following Type, Length, Value:

- Type: 15
- Length: At least 4 bytes.
- Value : two sub-TLVs are defined in the following.

If this TLV is present in the Hello message, this means the router is able to handle TCP connections. In order to open a TCP connection two things must be known: the IP address to connect to and a port. As both IPV4 and IPV6 are supported, we define three sub-TLVs to carry this information. Adding this TLV allows different versions of IS-IS to be compatible. Indeed, if the *TCP availability advertising* is missing in the IIH of one of the two routers trying to bring an adjacency up, this is the classical IS-IS which is used and all packets are exchanged on layer 2.
On the opposite, if the *TCP availability advertising* TLV is present in both IIH, then only hello messages are sent above layer 2 and a TCP connection is opened.

Port number sub-TLV

This sub-TLV is used only in the *TCP availability advertising* TLV and carries the port number on which the router has started a TCP server. This sub-TLV has the following Type, Length, Value field:

- Type: 1
- Length: 2 bytes
- Value: the port on which the server is listening.

IPV4 address sub-TLV

There is also a need to provide the IP address to which the connection must be established. There are two ways to provide it:

1. If the *TCP availability advertising* is present in the hello message, the sender might add the IP interface address TLV (type 132) containing the IP address of the interface.
2. Or the sender might add the IP interface address sub-TLV in the *TCP availability advertising* TLV with the following parameters:

(a) Type: 2

(b) Length: 4 bytes

(c) Value: the IPV4 address of the interface

If none of these are present and there is no IPV6 address sub-TLV as described after, the *TCP availability advertising* must be considered invalid.

IPV6 address sub-TLV

The IPV6 address sub-TLV is very similar to the IPV4 one. Except that if IPV6 is supported by the router to establish a TCP connection, this sub-TLV must be present in the *TCP availability advertising* TLV as there is no other to provide it.

1. Type: 3
2. Length: 8 bytes
3. Value: the IPV6 address of the interface

4.1.2 Open TCP connection

The method to open a connection depends on if IS-IS is configured to do a two ways handshake or a three ways handshake. But in order to choose which router acts as a server and which as a client, the decision process is the same in both cases.
The hello message are still sent directly above the layer two, this is not modified compared to the IS-IS standard implementation.
The first step for the router is to look if both routers have the same IP version by checking the sub-TLV in the TCP availability advertising TLV. If it's not the case, no TCP connection can be established and every LSPs are sent above layer 2 like in the standard implementation.

Establishing a TCP connection

The decision process to determine which router is the server and the client is quite simple. It is only based on the IP address advertised in *TCP availability advertising* TLV of the hello message. The router which has the lowest IP address in its hello packet is the server and hence the other has the responsibility to open the connection.

Two ways handshake

In the case of a two ways handshake, as soon as the router receives the IIH, it runs the decision process described above in order to determine which router acts as a server. The TCP connection is opened as soon as the decision process ends.
One corner case that can arise with this method is directly related to the drawback of the two ways handshake discussed previously. The client might attempt to open a TCP connection while the server did not already notice the presence of the client because of a client's hello lost, for example.
In this case, the connection is refused by the server and the client has to delay his next attempt by the hello message interval time.

Three ways handshake

In the three ways handshake, the connection is opened only once the three ways handshake is completed. The port number and the IP address must be included in all the hello messages exchanged during this phase. The decision process is then run at the end of the three ways handshake.
In this case the client router is sure that the server router has seen it.

Security consideration

In the context of this work, the IS-IS daemon launches a TCP server on each interface when it was setting up each circuit. If it turns out that one of the routers does not have the TCP extension capability, the TCP server must be killed in order to avoid any connection to it which could lead to some kind of LSP injection.
The first step to avoid this situation is to only allow the IP advertised in the hello message to connect to the server. However, this does not protect from a spoofed IP. To prevent this issue, the Generalized TTL Security Mechanism (**GTSM**)[8] could be used. The idea is, in IPV4 (resp IPV6), to set the time to live (**ttl**) (resp hop limit) to 255 to be sure that the packet comes from an adjacent node.
One other possible method can be to open the TCP server only once the decision process has determined if the router is the server. This the router is then sure to not have an unnecessary TCP servers running. The client in this case must delay its connection attempt at the end of the Hello exchange by a while in order to let the server router setting up is TCP server.
The drawback of this method arises when routers use the two ways handshake. Indeed, as said before, in this case the client router cannot be sure that the server router has seen it and thus has opened its TCP server. This can result in a refused connection due to the absence of a TCP server listening on the server router.

4.1.3 Loss of TCP connection

Once a router detects the loss of a TCP connection, it has to switch back to exchange every LSPs and SNP on layer 2 like in the classical IS-IS. There might be two reasons why the TCP connection was lost:

- It was interrupted due to some events but the router remain active.
- The router is down due to for instance a hardware failure.

If both routers are still working, the procedure to open a connection can be started again as Hello messages are still exchanged. But in order to avoid any time loss waiting for an hello to arrive while the adjacency is still up, all packets are sent above layer 2.
If one of the two routers is down, the other still sends every packet above layer 2 but as now hello is received anymore, the adjacency will be declared down after the timeout expired.

4.1.4 Acknowledgment of LSPs

Once LSPs are sent above TCP there is obviously no need to acknowledge LSPs received anymore as this is done by TCP itself. Thus, once LSPs are sent above TCP the sending of PSNP is no more useful. This means that for a p2p circuit, only one packet other than LSP has to be sent once the handshake had been completed and the TCP was opened : the CSNP.

4.2 Broadcast circuit

All the discussion above and results that are shown later consider IS-IS on p2p circuit. This section is a theoretical discussion of what happens with the TCP extension when IS-IS is running on a broadcast circuit.
As explained before, when IS-IS is running on a LAN, in order to decrease the size of LSPs in the LSDB and the number of LSPs exchanged when there is a route modification, a DIS is elected. The DIS is responsible for the synchronization of LSDBs across the LAN. As TCP is connection oriented, the extension does not work well in case of a LAN. Suppose a LAN with N routers attached to it. Indeed, with the TCP extension, the DIS should have to maintain $N - 1$ TCP connections. And in addition, instead of sending one LSP to a multicast address, the DIS should have to send multiple times the same LSP across its different connection. This could result in a massive peak in the traffic on the LAN.
The TCP extension proposed in this work does not fit in the case of LAN.
The solution for this problem is beyond the scope of this work. One way to address

it could be with a reliable multicast protocol such as TRMP [17]. However in case of a small LAN this could not be a problem. But if routers which belongs to the LAN have a lot of adjacency, this could be an issue. The exact point at which the TCP extension become inefficient is not determined.

4.3 Fragmentation

Another point of discussion with this TCP extension is the IS-IS fragmentation mechanism. As IS-IS runs directly above layer 2, it cannot benefit from the IP fragmentation mechanism.
To deal with LSPs that can be larger than the MTU, there is one byte in the LSP-ID which is called the fragment ID as shown on Figure 4.1. If an IS-IS router

Figure 4.1: The LSP-ID one byte dedicated to fragmentation.

has to originate an LSP which is above the MTU, for instance three time bigger, the router splits the original LSP in three segments with the fragment-id bit equals to 00,01 and 02 respectively, before transmission on the wire. The receiving router then simply installs the three fragments in its LSDB.
When the TCP extension is used by a router, there is no need for IS-IS to fragment a larger LSP. This is handled by TCP.
However, this is in the best case scenario, when all routers of a same area have the TCP extension enabled. If at least one router in the area does not support the TCP extension, LSPs can be sent above TCP but they must remain fragmented by the IS-IS mechanism as only the router which originates the LSP can fragment it.

4.4 Experimentation

Now that the protocol extension had been explained, this section shows some results on its performance.
The experimental setup is the same as the one described in Section 3.1.

4.4.1 Methodology

The methodology used to perform the analysis is very similar to the one used in Section 3.4. As now the IS-IS LSPs are sent above TCP, thsark is not able to

extract the LSP id anymore. In order to analyse the LSP exchange, a new tool was written in python to extract from binary data representing IS-IS packets the LSP id. The way to obtain the sending LSP pattern and the time needed to exchange all LSPs from the pcap is now divided in 3 phases:

1. Thsark is used to extract the time and the payload of the TCP layer and puts them in a CSV file.
2. Running the python tool on the CSV file to extract LSP-id and write in another CSV file with the time it was sent.
3. Running the python tool used in Section 3.4 to extract statistics from the capture.

The tests were run in the same conditions than the ones used to run the test on the standard implementation of IS-IS explained in Chapter 3. Again, multiple tests were made in multiple link capabilities. Results are shown for those three different link situations:

- No modification added on the link, called **no delay** in the following.
- $60ms$ of delay added on the link.
- 1% of packet losses.

4.4.2 No delay

In this case, there is no delay added on the link. The mean time needed to synchronize all the LSPs can be seen on Figure 4.2. The first noticeable element, is that the time needed to synchronize is much lower in this TCP solution than the basic implementation of IS-IS. Where in the IS-IS basic implementation, the time needed to flood all the LSPs is in order of seconds, when using the TCP solution this time drops drastically to only a few tenth of milliseconds.
Then as expected, the time needed grows linearly with the number of LSPs. This is a big difference with the classic implementation and can simply be explained by the fact that in the TCP solution we do not need to wait for five seconds for a PSNP or CSNP to consider an LSP acknowledgement. And the retransmission is directly managed by the TCP stack which vanishes the five seconds delay between retransmissions as was seen in the basic implementation.
Finally, the big variance seen for 200 LSPs and 300 LSPs can simply be explained by some instability on the link or in the routers CPU load .
The other point of interest is the sending pattern of the LSPs. This sending pattern can be seen on Figure 4.3. Only the sending pattern for 100 LSPs and 800 LSPs

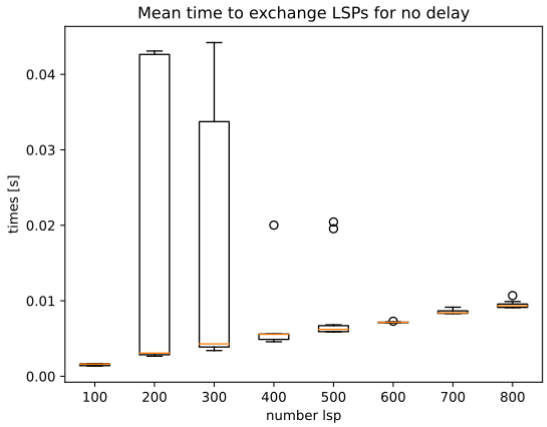

Figure 4.2: Mean time for exchanging LSP when no delay.

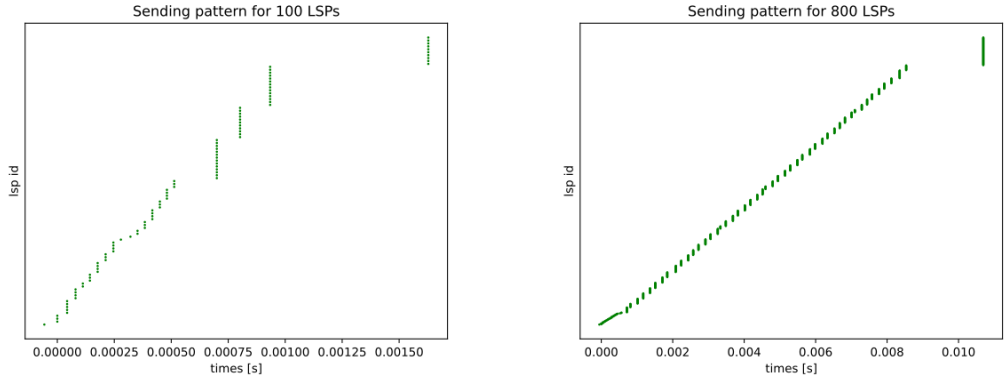

Figure 4.3: Sending pattern for 100 LSPs (left) and for 800 LSPs (right).

are shown as there is not a lot of variation between the different sending patterns of different numbers of LSPs. The sending pattern observed is almost the one expected, with almost a straight line and all LSPs sent once. One observation is surprising. A vertical line displayed at the end of the transmission means that several LSPs are sent at the same time. This is due to the TCP stack using the Nagle's algorithm [15] to aggregate some LSPs together in the same TCP packet to decrease the number of packet send over the wire. This can however be disabled by using the *TCP_NODELAY* option. In this experimentation the *TCP_NODELAY* was not enabled.
This behavior of TCP implied some modifications in IS-IS as the way it is implemented does not expect to receive multiple LSPs in only one packet. Another point of attention with this Nagle algorithm is that it can split up the LSP across

multiple packets. The receiver may have to reconstruct the LSP before processing it.

4.4.3 Modification on the link

The TCP extension was, like the classic implementation, also tested in two different scenarios with different link capabilities. The different scenario include:

- Adding loss on the link. Two different loss rate were tested: 1% and 5% packet loss.
- Adding some delay on the link. The delay range tested start from $10ms$ of delay to $100ms$ of delay.

Loss

In this case, there is no big difference in the behavior seen between the two percentage of losses, so only the 1% loss is shown.
When adding loss, the meantime for exchanging LSPs is not expected to increase a lot but the variance can increase a bit and there might be an increasing number of outliers far from the mean. Figure 4.4 shows the result. It can directly be

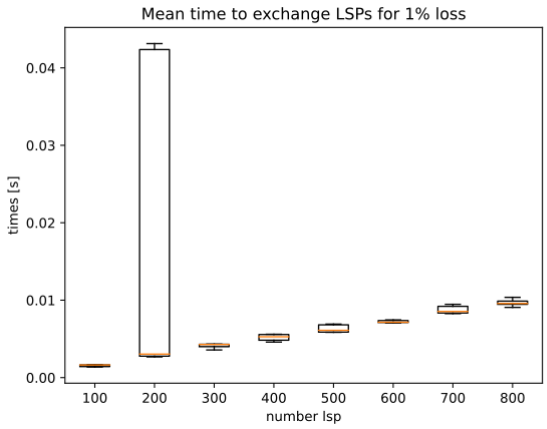

Figure 4.4: Mean time to exchange LSPs with 1% loss.

seen, that unlike when the IS-IS uses the standard implementation, the mean time to exchange all LSPs does increase drastically. This is of obviously due to the retransmission mechanism of TCP, such as the fast retransmit heuristic [25] and the Selective Acknowledgement (**SACK**) [14]. In this specific case, the good performances are mainly due to the fast retransmit heuristic because of the high

number of packets sent and relatively low losses. As the sender sends a lot of packets almost at the same time, it is quickly noticeable that a packet was lost due to the acknowledgment mechanism of TCP. On the opposite, in the basic implementation of IS-IS, if the packet is lost, the sender has to wait 5 seconds before transmitting it again. TCP improves a lot the meantime of LSPs exchanged in case of packet loss.

Delay

When delay is added, the behavior observed is very similar for the different delay values but with different value mean.
The first point to notice is that even with delay, the TCP extension performs much better than the standard IS-IS implementation. Then, one point which might be surprising is that there is always a step in the meantime to exchange LSPs when 500 LSPs are reached as shown on Figure 4.5. In fact this is the result of a combination

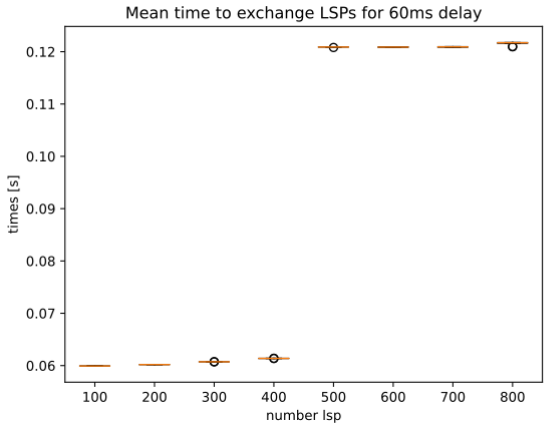

Figure 4.5: Mean time exchange for 60 ms delay.

of multiple mechanisms implemented by TCP:

- TCP retransmission timeout
- the TCP receiving windows

The TCP retransmission timeout should be set accordingly with the round trip time (**rtt**), in order to have maximal performances. In this case the retransmission timeout is larger than $60ms$ before considering a packet is lost. The TCP sending windows represent the maximum number of bytes a sender can send without waiting for acknowledgment. The behavior of those TCP mechanisms is clearly shown while looking at the sending pattern for 200 LSPs and 700 LSPs on Figure 4.6. Different

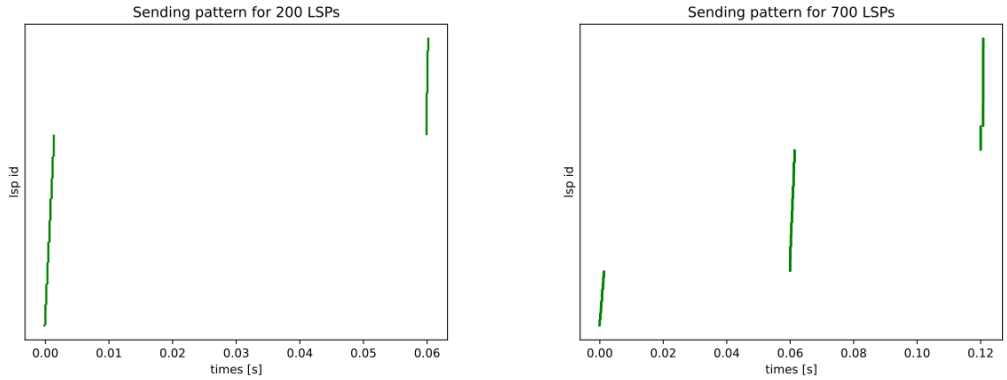

Figure 4.6: Sending pattern for 200 LSPs and 700 LSPs with $60ms$ of delay.

bursts can clearly be seen. Every almost vertical line represents the sender fulfilling the receiving windows. As there is a $60ms$ of delay, no acknowledgment is received for those TCP packets and sender as to wait as the receiving windows is full. Then $60ms$ later an acknowledgment is received by the sender, the remaining LSPs can thus be transmitted. In the case of 200 LSPs, all the LSPs are transmitted before the receiving windows is already full. For 700 LSPs, the receiving window is already full before the sender was able to send every LSPs. Another waiting of $60ms$ is necessary to finish the sending of all LSPs.
In conclusion, when there is some delay on the link, the TCP extension needs a time multiple of the rtt to achieve the LSP exchange. Nevertheless, this time remains far below the time needed in the standard implementation of IS-IS.

4.5 Latency

One other factor of interest in case of a router running some routing protocol, is the latency. The latency is the time needed for a router receiving a routing information on one of its interface to send it on its other interface running the routing protocol. The latency is a crucial parameter as the higher it is, the longer the network converge and for an IGP this is directly translated to a money loss.

4.5.1 Methodology

In order to measure the latency the setup used is shown on Figure 4.7. Three routers are put in line. The latency is calculated on router R2 as the difference between the incoming LSP identified by its LSP-id on interface $enp16s0f0$ and the same LSP sent on the outgoing interface $enp1s0f1$. The IS-IS daemon is running

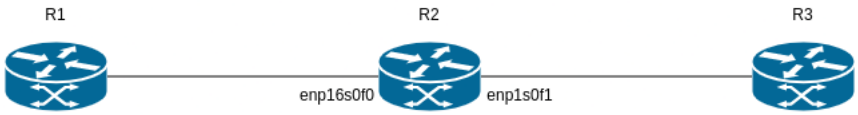

Figure 4.7: The topology used to measure the latency.

on R3 and R1.
On R1, in order to have a full control on LSPs sent, scapy [2] is used to send LSPs to R2.
Scapy is a tool written in python which simplifies a lot the crafting of packets by proposing a simple interface to change various field of any protocols. To calculate the latency, scapy was used instead of running the IS-IS daemon because of the much more important flexibility it offers on the field value.
The python script running on R1 simulates a real IS-IS process from the point of view of R2 except that it takes not into account any message sent by R2.
The behavior of this script is divided in two parts:

- First sending an IIH with the holding time set to 600 seconds to R2. R2 considers an adjacency up on interface enp16f0s0 upon reception of this IIH and then processes the LSP received on this circuit.
- Then every 2 seconds sending LSP to R2. LSPs sent can be of two different types:
  - A refreshed LSP. This is an LSP with an LSP-id which is already in the R2 LSDB but with an incremented sequence number.
  - A new LSP. This an LSP whith an LSP-id that R2 does not have in its LSDB.

There are 3 different configurations of connection between routers in which latency has to be tested:

- When all routers run the standard version of IS-IS and there is no TCP connection between them. This configuration is referred to the **raw-raw configuration** in the following.
- When R2 and one of the two other router run the IS-IS with the TCP extension. The third router run the standard version of IS-IS. This configuration is denoted by the **raw-tcp configuration** in the rest of this work.
- When all routers run the TCP extension version. This is called **tcp-tcp configuration** in the following.

In order to simulate a router with the TCP extension to run the test in the tcp-tcp configuration, scapy was extended to support the new TLV described in Section 4.1.1.
To have a good overview of the latency in the different situation, each latency mean is calculated on 250 LSPs sent.

4.5.2 Results

As Figure 4.8 shows, the latency when the TCP extension is enabled on R2 increases a bit, by a few micro seconds. This little increase can be due to the TCP stack which may increase a bit the processing time used to decode the receiving packet and to encode the LSP before sending it. Another observation, is that there is no

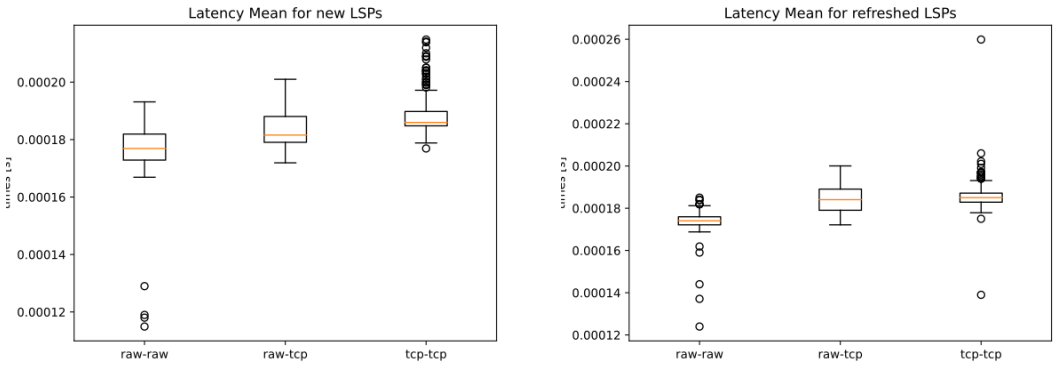

Figure 4.8: Mean time latency for the two different type of LSPs

real difference in the latency between a refreshed LSP and a totally new LSP as summarized in Table 4.1.

	raw-raw	raw-tcp	tcp-tcp
New LSP	176μs	183μs	188μs
Refreshed LSP	173μs	184μs	185μs

Table 4.1: Mean time for the three configurations and both type of LSP

4.6 Conclusion

In this chapter, a new TLV was proposed to extend the IS-IS protocol to use the efficiency of TCP to exchange LSP. This extension is only suitable when IS-IS is

running on a p2p link.
Using this extension, the time used to exchange all LSPs is much lower than the time needed in the classical IS-IS implementation. Table 4.2 summarize the mean time to exchange 800 LSPs in both implementation for three different conditions. The difference between the two implementation is due to the difference between

	No delay	1% loss	60ms delay
Basic implementation	9 seconds	14 seconds	14 seconds
TCP extension	0.01 second	0.01 second	0.12 second

Table 4.2: Time needed to exchange 800 LSPs in three different condition

the implementation of the reliability between TCP and IS-IS. When looking at the latency, which is defined as the difference between the time an LSP is received on an interface before being sent on another, the TCP extension increase by a bit less than a microsecond the latency
In conclusion, the TCP extension on p2p circuit, performs much better than the classical implementation of IS-IS.

Chapter 5

Further work

Even if it was shown that the TCP extension performs better at exchanging LSPs than the classical IS-IS protocol there is some unknown remaining.
This extension could be tested and compare to the classical protocol in a real datacenter environment to measure the convergence time of the network. However, this would not be very practical. The other solution could be to simulate a datacenter with a tool like IPmininet [26]. Even if the network is simulated, this could give a good overview on the time needed for the network to converge with this extension compare to the classical protocol. Or using the same methodology as [7] to compare the IGP convergence in large IP networks.

Another point of interest could be to determine why the classical IS-IS implementation loses packets. As explain before, there is for now, no clue about why IS-IS does not treat all LSPs received. This directly result in a poor LSPs exchange performance.

Finally, a last point of further investigation might be how to extend the use of this TCP extension to a LAN circuit. One could also be interest to know for how many routers on the LAN the TCP extension is suitable. Or also see how some mechanism implemented by IS-IS like the fragmentation could be modified to take advantage of the TCP/IP mechanism.

Chapter 6

Conclusion

In this master thesis, we analysed the performance on LSPs exchange of the IS-IS routing protocol. We showed that the retransmission mechanism and the way to exchange LSPs in order to synchronize the LSDB is not optimal. The time to exchange LSPs is in order of seconds even ten of seconds and increase by stages regarding the number of LSPs to exchange. Performances were tested in different link capabilities.

We showed how it is possible to extend the IS-IS protocol to allow the use of TCP for exchanging routing information while staying compatible with the standard IS-IS protocol. We added a new TLV with his corresponding sub-TLV for the router to be able to inform his neighbor that it support the TCP extension.

We studied the performance of the protocol with the TCP extension in terms of time needed to exchange LSPs. We showed that, in every link capabilities tested, the TCP extension improve by a factor 100 the time needed to exchange routing information. This is mainly due to the congestion control and the retransmission mechanism of TCP.

Although there are still some untested cases (Chapter 5), the extension proposed shows that it is possible to improve the information exchange between IS-IS routers. This extension could improve the convergence of the network and allow ISP to reach their SLAs.

Bibliography

[1] Number of internet of things (IoT) connected devices worldwide in 2018, 2025 and 2030. URL: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/.

[2] Scapy. URL: https://scapy.net/.

[3] tshark. URL: https://www.wireshark.org/docs/man-pages/tshark.html.

[4] Neil Briscoe. Understanding the OSI 7-layer model. *PC Network Advisor*, 120(2):13–15, 2000.

[5] Bruno Decraene, Jayesh J, Tony Li, Gunter Van de Velde, and Chris Bowers. IS-IS Flooding Parameters advertisement. URL: https://tools.ietf.org/html/draft-decraene-lsr-isis-flooding-speed-04.

[6] Editor. Internetwork Packet Exchange (IPX). URL: https://networkencyclopedia.com/internetwork-packet-exchange-ipx/.

[7] Pierre Francois, Clarence Filsfils, John Evans, and Olivier Bonaventure. Achieving sub-second igp convergence in large ip networks. *ACM SIGCOMM Computer Communication Review*, 35(3):35–44, 2005.

[8] V. Gill, J. Heasley, D. Meyer, and C. Pignataro P. Savola. The Generalized TTL Security Mechanism (GTSM). RFC 5082, RFC Editor, October 2007. URL: https://datatracker.ietf.org/doc/html/rfc5082.

[9] Walter Goralski Hannes Gredler. *The Complete IS-IS Routing Protocol*. Springer, 2005.

[10] T. Li, P. Psenak, L. Ginsberg, H. Chen, T. Przygienda, D. Cooper, and L. Jalil. Dynamic Flooding on Dense Graphs draft-ietf-lsr-dynamic-flooding-06. URL: https://datatracker.ietf.org/doc/html/draft-ietf-lsr-dynamic-flooding-06.

[11] T. Li and H. Smit. IS-IS Extensions for Traffic Engineering. Standard, International Organization for Standardization, October 2008.

[12] Linux Foundation Collaborative Projects. FRRouting. URL: https://frrouting.org/.

[13] Athina Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, and Christophe Diot. Characterization of failures in an ip backbone. In *IEEE INFOCOM 2004*, volume 4, pages 2307–2317. IEEE, 2004.

[14] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgment Options. RFC 2018, RFC Editor, October 1996. URL: https://datatracker.ietf.org/doc/html/rfc2018.

[15] Bradley Mitchell. An Overview of the Nagle Algorithm for TCP Network Communication. URL: https://www.lifewire.com/nagle-algorithm-for-tcp-network-communication-817932.

[16] John Moy. Ospf version 2. STD 54, RFC Editor, April 1998. http://www.rfc-editor.org/rfc/rfc2328.txt. URL: http://www.rfc-editor.org/rfc/rfc2328.txt.

[17] Y. Nagasaka and S. Kajiyama. A reliable multicast protocol, trmp, for data acquisition systems. In *2006 IEEE Nuclear Science Symposium Conference Record*, volume 2, pages 982–985, 2006.

[18] Information Sciences Institute University of Southern California. INTERNET PROTOCOL. RFC 791, RFC Editor, September 1981. URL: https://datatracker.ietf.org/doc/html/rfc791.

[19] D. Oran. OSI IS-IS Intra-domain Routing Protocol. RFC 1142, RFC Editor, February 1990.

[20] PACKET(7) Linux Programmer's Manual. URL: https://man7.org/linux/man-pages/man7/packet.7.html.

[21] David A Patterson et al. A simple way to estimate the cost of downtime. In *LISA*, volume 2, pages 185–188, 2002.

[22] Quaggua community. Quagga Routing Suite. URL: https://www.quagga.net/.

[23] The European Commission's science and knowledge service. Telework in the EU before and after the COVID-19: where we were, where we head to.

[24] Robert Sedgewick and Kevin Wayne. *Algorithms Fourth Edition*. Addison Wesley.

[25] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC 2001, RFC Editor, January 1997. URL: `https://www.rfc-editor.org/rfc/rfc2001.txt`.

[26] Olivier Tilmans. IPMininet. URL: `https://github.com/cnp3/ipmininet`.

Appendix A

Json format

This is the exact json format that is generated by the tool and loaded by FRRouting after modifications.

```
"LSP":= {
  "hdr": HDR,
  "tlv": TLV
 }
```

The *HDR* and *TLV* are defined thereafter.

```
HDR := {
    "pdu_len": 1,
    "rem_lifetime": 1142,
    "seqno": 1,
    "checksum": 1,
    "lsp_bits": 1,
    "lsp_id": [0,2,3,4,5,6,0,0]
}
```

```
TLV := {
protocol_supported": Protocol_supported,
    "ipv4_address": [IPV4_Address],
    "extended_ip_reach": [ Extended_IP_reach ],
    "extended_reachability": [ Extended_reachability ],
    "TE_router_id": "10.10.2.1",
    "router_cap": Router_cap
}
```

The *Protocol_supported* json object represent TLV 129. the *protocols* field is an array containing all the protocol supported in hexadecimal as define by IS-IS.

```
Protocol_supported := {
    "count": 1,
    "protocols": [204]
}
```

The *IPV4_Address* is the json object representing TLV 132.

```
IPV4_Address := {
    "addr": "10.10.2.1"
}
```

The *Extended_IP_reach* object represent the TLV 135.

```
Extended_IP_reach := {
    "metric": 10,
    "down": false,
    "prefix": "10.10.2.0\\/24"
}
```

The *Extended_reachability* object represent TLV 22.

```
Extended_reachability := {
    "id": [1,2,3,4,5,7,0],
        "metric": 10
}
```

The *Router_cap* object represent TLV 242.

```
Router_cap := {
    "router_id": "10.10.2.1",
     "flags": 0
}
```

Appendix B

Common Header

The structure of the IS-IS common header which is the same for all IS-IS packet is shown on figure B.1 The different fields are explained hereafter:

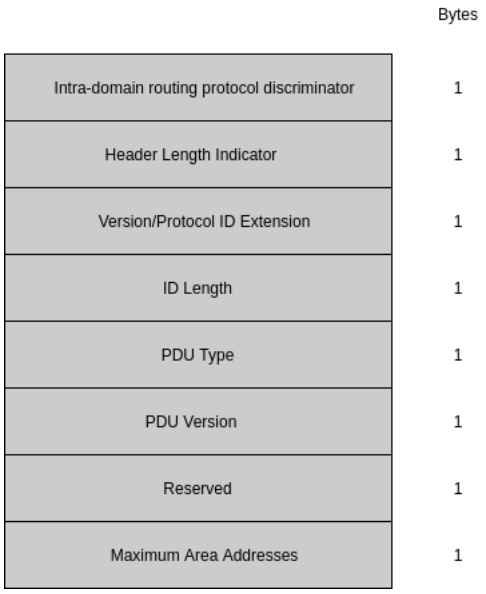

Figure B.1: The common Header of all IS-IS packet.

- *Intra-domain routing protocol discriminator* is always the same value $0x83$.
- *Header Length Indicator*, give the length of the common header plus the length of the header of the PDU.
- *Version/Protocol ID Extension* give the version of the protocol, should always be 1.

- *ID Length* give the length of the ID. It is usually 6.
- *PDU Type* give which type of PDU is encapsulated in this header: IIH, SNP or LSP.
- *PDU Version* should always be 1.
- *Reserved* should always be 0.
- *Maximum Area Addresses* give the maximum area address, it is in general 3.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**