universität
wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Efficient treatment of routing anomalies with the help of distance vector protocols“

verfasst von / submitted by

Florian Hermann BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2020 / Vienna, 2020

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von / Supervisor:	Univ.-Prof. Dr. Stefan Schmid
Mitbetreut von / Co-Supervisor:	Dipl.-Math. Dipl.-Inform. Dr. sc. ETH Dr. Klaus-Tycho Rüdiger Peter Förster

1 Acknowledgement

Many thanks to my supevisor, Univ.-Prof. Dr Stefan Schmid and co-supervisor Dr Klaus-Tycho Förster who supported me with knowledge and feedback during my work. In addition, you always took your time for me and were there for me if I had any problems or questions, thank you very much. I would also like to thank the expert Peter Paluch who helps me with the understanding and problems of the EIGRP protocol and the DUAL I have. But also many thanks to Vladimír Veselý and Jan Zavřel who helped me with problems with OMNET ++ and supported me with errors and questions. I thank my friends a who supported me in processing and where I did not find the motivation to continue working on it. Last but not least, I would like to thank my family, my mother Marianne, my father Gerhard and my brother Manuel. They kept me going and this work would not have been possible without their support.

Contents

Abstract

Computer networks have become a critical infrastructure. Today's networks should be very flexible and support fast updates to avoid undesirable configurations inside the network like routing loops. Current routing protocols only partially have a method to deal with such undesirable configurations. However, routing problems can still occur with these methods sometimes. This master thesis deals with a routing protocol called the Enhanced Interior Gateway Routing Protocol(EIGRP) which uses that Diffusing Update Algorithm(DUAL), which detects this misbehavior and has suitable mechanisms to guarantee loop freedom. By modifying EIGRP, various experiments are created in the simulators OMNET++ and GNS3 and compared with existing routing algorithms. Various properties are checked to determine the efficiency of the routing algorithms so that further properties such as waypoint enforcement can also be guaranteed. The goal of this work is to guarantee the property of waypoint enforcement by expanding EIGRP. We will use simulations to show that the extension allows the packages to include the waypoint in the route calculation.

2 Introduction

Computer networks can be seen as critical infrastructure. Because of the interconnectedness of the world and the fact that everyone can communicate with each other, there has been an increase in data. As a result, computer networks such as data centers have become a complex and sensitive infrastructure (see [10, p. 1]). With the increase in users, operating traditional networks has become a complex and error-prone task. Because more and more services are offered by different manufacturers and the network traffic increases, different properties such as availability, performance and correctness must be observed. Companies with a technical background such as GitHub or Amazon also report problems with misconfiguration in the networks, which can lead to routing anomalies such as routing loops [10, p. 1]. A story is told in [29] that tells of a bank failure in a data center. Every minute that the system is down, there is a loss of sales in the millions. Different teams were now trying to find out what the problem is or why the error occurred in this form. Since the network team was only equipped with a ping and traceroute and these tools only test end-to-end connectivity, problems can arise here because you cannot find out which network devices or packets are affected (see [29]). This story shows that when using traditional networks, it is important that you have a plan of how to proceed in the event of a fault. It is also important that you not only rely on the proven tools but also use monitoring tools to analyze where the errors occur. Based on various properties that the network has to offer, it is also necessary that networks have to be designed flexibly. This flexibility is necessary so you can react faster when properties such as increased network traffic or changes in the routers occur (see [10, pp. 1-2]).

2.1 Motivation

We all use the internet every day to communicate with friends or do something for work. We communicate with each other and that an email arrives where it should. Streaming is also very popular these days. If the picture of the film or series is in order, we hardly ask ourselves how the connection works and what happens there in the background. However, in the event of an error, we are already wondering why the connection is so bad at the moment or why the connection to the internet has been lost.

Networks like those in data centers have grown larger and more complex in recent decades. To cover different services, traditional networks have to be so busy and flexible that you can react quickly in the event of an update or error in the network, to lose a few packets as possible and to prevent routing loops from forming. Points are defined in [10, p. 2] which are important and must be taken into account in a network update.

- Flexibility: Flexibility is especially important for network updates and changes in the network.
  - Security policy changes: The data traffic in the network is to be routed through a certain point. This can be a firewall that checks packets for their content. This can be achieved through certain policies (see [10, p. 2]).
  - Traffic engineering: A network operator can decide whether to redirect the packets via a different path for certain reasons. That would improve the metrics. One reason for the change could be that one path is so overloaded that rerouting makes sense because another path is less loaded (see [10, p. 2]). For this purpose, the network operator can bring the packets to their destination faster with the help of load balancing. The strict guidelines of the federal states could be another reason for the network operator that the packets have to be rerouted (see [10, p. 2]).
  - Maintenance work: Due to technical maintenance work such as updates or a defect in a network device (router), it is important to redirect the routes and thus the network traffic (see [10, p. 2]).
  - Link and node failures: Link or node errors are very common and unexpected. The most important question here is that you find a suitable method to deal with the mistakes. Monitoring tools can discover the errors and counteract them in an emergency. Different reversals are needed so you are prepared for the mistakes next time. For this purpose, algorithms are used, which does something against occurring errors (see [10, p. 2]) .
  - Service relocation: A network is the interplay of several services that are executed simultaneously. If a service is changed, be it adding or removing, the network must be updated. The corresponding routers

have to be processed for a short time and therefore cannot forward packets for this specific time. Another decision that has to be made is that the corresponding services can also process the respective packets again (see [10, p. 2]).

- Maintaining Consistency: The network should maintain a certain consistency in the event of an update. Certain properties should be taken into account. A package should only be forwarded via a certain path during an update (see [10, p. 2]). There must be no mixing between the old and new path. Finally, it is also important that packets are never dropped or are part of a routing loop and are sent back and forth all the time (see [10, p. 2]).

So that you have to take which route the packets within a network, you use a so-called Interior Gateway Protocol(IGP). In the case of growing networks or when handling the services, the network operator must reconfigure the (see [33, p. 1] [34, p. 1]). IGP are divided into two different methods. On the one hand, there are distance vector and on the other hand, there are link-state-protocols. However, due to the advantages, link-state IGPs using Open Shortest Path First(OSPF) or Intermediate System to Intermediate System(IS-IS) are being used more. Only a few use distance vector protocols because these are not used as often due to some restrictions. In [33, p. 2] [34, p. 2] are reasons why the focus is more on link-state protocols. One reason for this is that these are used more often than distance vector protocols. Another reason is that with the help of link-state IGPs one can be configured in a flat or hierarchical mode. In flat mode, all routers know what the entire network looks like and send the packets to their next destination using the shortest path [33, p. 2] [34, p. 2]. The other mode would be the hierarchical mode where there is no guarantee that the shortest paths will be used. With the help of the hierarchical configuration it is possible to divide a very large network into zones called areas in OSPF or levels in IS-IS (see [33, p. 2] [34, p. 2]). This subdivision can ensure that network traffic can be kept to a minimum. In addition, monitoring in the event of an error or update in this area is also clearer than with a larger network. Other router types that are used in this mode are also presented and methods are used that are used in this context. After the topology has been set up, the routes that the packets should take are defined using the link-state algorithm (see [33, p. 2] [34, p. 2]).

That's why we decided to analyze and extend distance vector protocols to guarantee loop-free access and to make them faster when it comes to handling network updates. There are already options like EIGRP and its DUAL, but there are various ways to extend it. This means that the DUAL only guarantees freedom from loops, but network policy properties are not guaranteed.
For this reason, there are distance vector protocols used, because there are already a lot of papers that look at link-state protocols and their methods. With this thesis, the goal is to show that there can be distance vector protocols that can be used in today's area by using the right algorithm.

2.2 Goal of the work

In this master thesis, we deal with the topic of network update problem in distance vector protocols. We concentrate on the Diffusing Update Algorithm(DUAL), because it is already implemented in the Enhanced Interior Gateway Routing Protocol(EIGRP) and has implemented the properties of distance vector protocols. It was an exciting prerequisite that the DUAL offered the property of loop-free in every instance. However, the goal of the work is to extend EIGRP so that the algorithm guarantees that every packet goes through the firewall. Therefore, I have considered various questions that I will examine in this work. These questions are:

- Where is the extended EIGRP significantly better or worse compared to existing ones?
  - How does the extended algorithm work when you make various changes in the network?
  - How does it behave in the simulator?
  - How does it maintain itself in different large networks?

  An extension is created based on EIGRP and tried in a selected simulator called OMNET++. In this work the GNS3 simulator is also used so we can specifically address the update times of the routing protocols. Using different simulation results, there is an evaluation of how the extension relates to the existing EIGRP and to other existing routing protocols. Different large networks are simulated to show where the protocols is best received. There will be a comprehensive evaluation, where you can see in which areas the extended protocol has strengths to existing ones. The evaluation offers three experiments that analyze the different protocols. These points are described in the Section 7.

- Can you create an efficient and fast extension based on distance vector to limit the anomalies in the network?
  - What changes or exchanges are there to the current routing protocols?
  - Since EIGRP and his DUAL will be used, whether it can guarantee other properties such as network policy properties?
  - Can you just focus on the properties of distance vector protocols most of the time?

  The algorithm includes many properties of distance vector protocols. In Section 5 you can see the structure and which important points and decisions it contains. In connection with the research question, it is investigated whether distance vector protocol properties can be used or whether link state properties are required. By using EIGRP you have to take some properties of link-state protocols, because different tables are needed. In

addition, another property apart from loop freedom such as waypoint enforcement is to be guaranteed.

This work is for people who are interested in network update problems and how to counteract these problems using distance vector properties. An extension to EIGRP is to be presented to these people that converge to existing algorithms more efficiently and other properties than loop freedom can be guaranteed. The goal of waypoint enforcement is that every packet that goes through the network goes through the firewall. We want to achieve this property with our idea, but then it is often down to the configuration of the network. In this work a topology was created where this can be achieved if EIGRP is modified. As the networks get bigger, care must be taken to configure it to go through the waypoint.

2.3 Methodology

To answer the research questions mentioned above, an analytical instrument is required. In this case, defining and creating an experiment is the most appropriate method. The subject at hand has already been compared in a similar way, for example, by comparing EIGRP with OSPF and also recording the results accordingly.

An important part of the work comprises an experiment where the extension is compared with existing algorithms like EIGRP , OSPF or RIP. This experiment will be created with the help of the network simulator OMNET++ and GNS3. The results of the simulation are evaluated and assessed using defined metrics. Link weights are changed on various large networks and various link failures are simulated and analyzed how the corresponding routing protocols react to them. The routing protocols EIGRP, OSPF and RIP are compared with one another. The simulation is carried out for a certain time and then interpreted using various graphics and tables. For this purpose, it is examined whether each packet goes through the firewall specified in the experiment.

2.4 Delimitation

This master thesis deals with the planning and extension of a distance vector algorithm, which deals with routing anomalies within a network and prevents them. In addition, the extension should ensure certain properties such as taking into account a firewall and the packet should go through it. The implementation includes the existing DUAL, which is used by the routing protocol EIGRP. Since this has implemented some interesting functions, it builds on it. The algorithm should primarily relate to properties of distance vector protocols and it should not be examined here on how to build on link status protocols. There is also no reference to software-defined-networks(SDN) and how SDN can facilitate protocol configuration and network behavior. It is also not planned to imagine that there are several firewalls in the network. So there is no goal to solve network update problems that have to do with multiple waypoints. The extended protocol only works if there is one firewall in the network.

2.5 Structure of the work

The structure of the master thesis is described here and what is planned in each of these areas. This thesis is divided into six different sections. In Section 3 there is a simple overview of important routing methods, their protocols and important terms that occur in the master thesis and are important. It also explains why network updates can be such a problem. Section 4 also follows, which describes existing algorithms and technologies that are used in reality. The algorithm structure is described in Section 5. At this section a description of what problems there are and how that will solved with the planned idea. The most important properties that characterize the extension are mentioned here. In the next part, which you will find under Section 6, covers the information on how the extension is implemented.

Here different configuration details and why it was implemented the way, is being described. Interesting alternatives are offered and go into the core points of the implemented idea. An important part of this master thesis is described in Section 7. Here the simulation results and how the experiments are set up are described.Various tools that are used during the evaluation to get results are mentioned. Another point that is described is the parameters used, which are important to get the results. The thesis conclude with Section 9.

3 Basics

In this area, various terms and methods that build on the master's thesis are described. Different routing methods such as link-state and distance vector protocols have been mentioned. What effects network updates could have on a network such as a data center is also described.

3.1 Routing & Routing Protocol Methods

Each of us has information on the internet every day, to be always up-to-date or to communicate with someone. But how can you get the connection so quickly and get the information you need? There are many technologies behind it that need to work together to make this connection as fast as possible. However, when speaking of routing, the router is the most important device in this context. The simplest definition of why use a router is because you connect the networks with this device. The router uses a forwarding table to determine the best path to a specific destination and which router to send the packet to next [3, p. 44].

3.1.1 Static Routing

You have to manually add and delete the respective ones on the different routers. The entered routes are saved accordingly in the routing table [19, p. 79]. Is a basic way to tell a router where to forward packets to networks that are not directly connected. The best way to use static routes is if you have a smaller

network that has only one path to an outside network [3, p. 153]. A special case of the static route is a default static route. It is a special case of the static route. With this route, which can be learned both statically and dynamically, it is possible to forward all packets that otherwise have no match membership. These make sense if you are connected to the companies edge router of the ISP network. Another case would be if the router has no other connection to a router (see [3, pp. 156-157]).

3.1.2 Dynamic Routing

The problem with static routing is that you can not react easily to network changes. Since this can be very stressful for larger networks, algorithms for route determination are now used. Now we use dynamic routing so that we can make the path determination automated and this is done with the help of routing protocols (see [7, p. 59]). Routing protocols are used here to exchange information between the routers. This protocol comprises a series of processes, algorithms and messages to fill the routing tables to create the best routing paths [3, p. 233]. Another feature of this routing method is that the algorithm can also deal with existing problems in networks. In contrast to static routing, the size of the network is not relevant here. Imagine that you have to statically configure thirty routers and always monitor them (see [3, p. 233]). We often see that this does not work so well in "Network Update Problem". Routing protocols have been around for several decades and have always been further developed because of their complexity. This work will only concentrate on the most important routing protocols such as RIP, EIGRP and OSPF.

3.1.3 Administrative Distance

The AD is used to select the best route for a path. The router uses the AD to rate the trustworthiness of routing information sources. If different routing protocols are used for the same network, the router will use the administrative distance to save the best route in the routing table (see [19, p. 80]). The motto is, the smaller the value, the better the route. If there are several routes to the same network and with the same AD, then the metric is used for the best route [19, p. 80]. As you can see in Table 1, you see the different administrative distances that you see in the routing table when you implement the routing protocols. In this table it is important, the smaller the administrative distance, the better the route.

3.2 Link-State-protocols

Link-state-protocols calculate the paths by taking the shortest path available from source to destination by using the Dijkstra algorithm. This calculates the shortest paths in the respective domain. The algorithm uses the costs of the connections from the routers for the calculation [43, p. 3]. In an autonomous

Route Type	Administrative Distance
Directly connected	0
Static	1
EIGRP(internal)	90
IGRP	100
OSPF	100
RIP	120
EIGRP(external)	170

Table 1: The routing protocols and their Administrative Distance(AD) default value.

system, each router has its own database, where the connection states are located. Every router on the network has the same database, which is then used to describe the network topology. Each router executes the algorithm using its link-state database. First, create a tree with each router as root. From this, a tree is created that contains the shortest paths and is made available to every router (see [43, p. 3]). Link-State Advertisement(LSA) is responsible for the exchange of routing information between routers [43, p. 2]. The message exchange from the LSAs is sent using the flooding method. Each router floods its LSA with the network and then each router receives the LSA and processes it. In the event of a network topology change, the router sends LSA to the network. The other routers will soon be informed of changes in the network topology (see [43, p. 2]).

3.2.1 OSPF

OSPF is an open standard protocol that is defined as a link-state protocol. With the help of OSPF, the routers and links can be added to certain areas. The different areas must be balanced, as stability must be taken into account in this context. If the different areas are larger, the different routers must also handle more messages, which can cause slow processing of the messages (see [20, pp. 2-3]). Because the topology can change at any time and the network traffic is accordingly large because each router has to recalculate the different routers. OSPF routers exchange routes with neighbors that have established adjacencies. Such adjacency is called as a relationship between two routers that allows the exchange of routing updates [19, p. 91]. OSPF uses so-called LSUs for topology changes. If a link or router fails, the router that gets it will flood a status message to the neighbors as quickly as possible and this message will reach the entire network (see [19, p. 93]). OSPF is defined as a classless routing protocol that permits the use of variable-length subnet masks(VLSMs). It also supports equal-cost multipath load balancing and neighbor authentication. Multicast addresses are used to communicate between routers [7, p. 375]. With OSPF you have different router types that depend on the respective place in the network architecture. You can see them in [7, p. 379] and are also described in detail

there and how they are used.

3.2.2 IS-IS

The network is divided into routing domains [2, p. 1]. IS-IS uses different levels in a routing domain, whereby the routers can see themselves in the different levels. There is also a subdivision into two levels, with only routers are in level two exchanging data packets or routing information with external routers outside of this domain [2, pp. 5-6]. Level one routers route within an area so you can tell from the destination address in the packet whether or not the destination is within the area. If this is not the case, the packet is forwarded to a level two router [2, p. 7]. The Dijkstra algorithm is used, i.e. shortest path first.

3.2.3 Challenge with network updates

The challenge with link-state protocols when updating is that a new map of the topology has to be calculated. Since the routers have to recalculate the paths to calculate the path to the destination, inconsistencies can occur during the path. It may well be that a router updates faster, which can lead to routing loops for a certain time within the network. Only when all routers have the same plan in mind can a smooth exchange of messages be guaranteed. Only when all updates have been processed can the Dijkstra algorithm be run again and the shortest paths to the target can be calculated.

3.3 Distance vector protocols

With distance vector protocols, each router exchanges all known routes in the routing table with its neighbor. With the effect that each router should have a complete table for all subnets in the network. The routers combine their own routing tables from the various updates from the neighbors. The difference to link-state protocols is that a router does not know the entire plan of the network (see [19, p. 75]). Distance vector routing protocols use the Bellman-Ford algorithm. All known routes, the complete routing table, are sent to the neighbors and received from them (see [19, p. 76]). Messages that are continuously sent are used for this. This ensures that each router has a complete routing table for each subnet. After the neighbor received the routing updates, he would announce all routes learned by the neighbor via the respective interface received. Another step the Bellman-Ford has to take is if a router fails. Here, the router would delete the respective router from the routing table after an update has expired (see [19, p. 76]).

3.3.1 RIP

Routing Information Protocol(RIP) is one of the distance vector protocols and was developed for smaller and middle networks. With the help of the hop count, the best route is found in a subnet [19, p. 80]. A router that has been

configured with RIP sends its routing table to the neighbors every thirty seconds and receives a response message, where the routing tables of the neighbors are located [20, p. 1]. The maximum hop count at RIP is fifteen. This means that only fifteen routers can be visited for a particular route. If a sixteenth router has to be visited, it is marked as an infinite distance [20, pp. 1-2]. With RIPv1 and RIPv2 you have two different versions, whereby in IPv4 networks RIPv2 is more likely because of the extensions (see [20, p. 2]). In IPv6 networks, RIP has provided an extension with RIPng. RIP uses the UDP transport protocol to send individual messages. With the RIPv2 version, authentication for the messages is now also guaranteed (see [20, p. 2]).

3.3.2 EIGRP

EIGRP is an advanced distance vector protocol that implements some characteristics similar to link-state protocols [7, p. 345]. With the help of Hello packets, it is determined whether the connection to the neighbor is still working and the neighbor is not yet down [4, p. 4]. The DUAL algorithm is used to determine the respective routing paths. All route decisions are taken from all neighboring routers and the paths are determined. The algorithm then takes the route into the routing table based on the distance information. Important terms in route determination are feasible successor and successor. The successor is a neighboring router that is used to take the route that has the lowest cost to a destination. In addition, this path must guarantee no loops (see [4, p. 4]). In [44, pp. 2-8] you see how the algorithm works with diffusing updates. Below you will also find proof of why the respective implementation and properties of the DUAL work. If there is a change in topology, the DUAL will check for a possible feasible successor that has a route to the destination and successfully create a feasible condition. If there is a feasible successor, this becomes the new successor, otherwise, there is a reconfiguration. This reconfiguration affects convergence time. However, reconfiguration is avoided in most cases because it is computationally intensive. That is why you have the option with the Feasible successor to provide a backup route [4, p. 4][7, p. 347]. In the following, the points have taken from the RFC Standard 7868 to clarify the difference between successor and feasible Successor.

- Feasible Successor: A neighboring router that fulfills the feasibility requirement for a specific destination and thus provides a guaranteed loop-free path [32, p. 6].
- Successor: For a specific destination, a neighboring router that fulfills the feasibility requirement provides the most cost-effective path at the same time [32, p. 7].

3.3.3 Challenge with network updates

With distance vector protocols, the message about an update in the network spreads slowly because the messages are exchanged between the neighbors. So-

called triggered updates are exchanged, which are sent immediately when a change in the network is detected. Because the entire routing tables are sent, the routers must also process the respective information, which affects the convergence time (see [19, p. 75]). Since it takes longer for the network to reach a convergent state, inconsistencies can occur in the network. There is also the problem here that some routers process the messages faster and can lead to inconsistent paths because regular updates can be sent and received by neighbors at the same time (see [19, pp. 75-76]). This means that the routers that did not receive the triggered updates can send information about a route that no longer exists.

3.4 Network Update Problem

In this section, update problems and technologies that should solve this problem are described here. In this work, we refer to network updates in traditional networks, where the routing rules are calculated using routing protocols. The active links and metrics play an important role in this so that the paths for forwarding can be calculated. Various methods are now being used to avoid temporary inconsistencies due to changed topologies or configurations. These methods aim to bring in a certain order, so you stay converged through these updates or bugs.

3.4.1 Routing loops

Transient routing loops are described in [18, pp. 1-2]. In [18, pp. 1-2] routing loops are defined as inconsistencies in routing state. With these inconsistencies, routing loops can easily occur. When routing loops occur, it is usually characterized because a packet is always sent back and forth between two nodes. These are also shown as replicas of the original package which can be recognized by the time-to-live(TTL). Sometimes, routing loops can also be very difficult to find if they are not dealt with within an autonomous system(AS) and therefore extend over several AS (see [18, pp. 1-2]).

In Figure 1 you see a small network of three nodes. You can see in the initial state (Figure a) everything works without problems, R1 has a link to the next network and every packet goes via R1. However, it can now happen that the link between R1 and the next network fails or that an update to R1 has to be made. You can also see that R2 has a link to the next network. If the link from R1 fails, the routing traffic must now be redirected, so you now go over to R2. If R2 does not immediately notice the error, he will forward every packet to R1. However, R1 knows about his error and will send the packet back to R2 and a routing loop is the result (you can find this in Figure b) (see [18, p. 2]). The third example of this figure can be found under c. This deals with the scenario where R3 has already been updated, but the loop between R1 and R2 still exists. Now R3 forwards the packets, but R2 sends them to R1. Thus this loop exists until each router has a new plan and this could take several

milliseconds. In the worst case, the packets are thrown away by the router [18, p. 2]. So you can see how easily routing loops can be created. A small change or an update in the network is enough so that the different routers are no longer synchronized. If the different route states are no longer synchronized, it can happen that loops can occur in the network as in Figure 1 (see [18, p. 2]). However, we have shown the example with the routing loops on small networks so that everyone understands it. Now you imagine that networks like in data centers are not just three routers but much more. It can easily happen that a router is out of line or needs an update and the paths have to be reconfigured. Even more important are methods and algorithms that detect these anomalies and react to them with different properties.

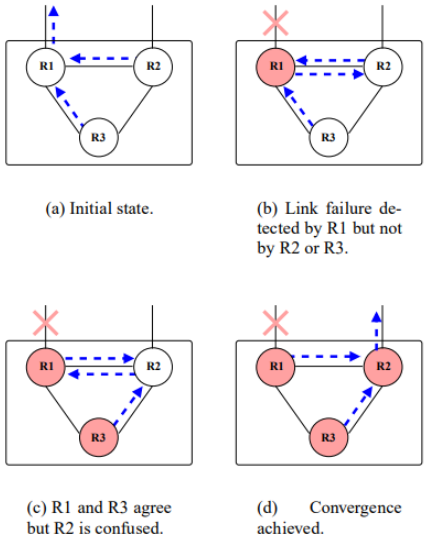


Figure 1: Scenario of transient loop [18, p. 2]. Scenario with three routers which show you some example what routing loop mean and how it can happen.

3.4.2 IGP reconfiguration

As mentioned earlier, this thesis focuses on traditional networks configured with IGPs. Routing tables and forwarding paths are computed in traditional networks using routing tables [10, p. 3]. Here you can decide whether you want to use link-state or distance vector protocols. Link-states are used in the frequent papers because of the better handling and some properties.The properties of the different routing protocol methods and how the respective methods calculate the paths in the network are already described before. If there is a change in the network, the network graph must be updated (see [10, pp. 3-4]). This is done by exchanging many messages on the network until the network is in a converged

state. However, until the network converges again, various problems can arise because the routers are not updated in a specific order. Some operations are listed in [10, pp. 4-5] that can deal with network update problems. Now follows an enumeration with such operations:

- Protocol Extensions: The point describes a simple solution for reconfigurations in traditional networks, but this method is very limited. In [10, p. 4] goes to the point protocol extensions. They used a modification of IGPs to avoid forwarding inconsistencies. The authors François and Bonaventure [13, pp. 1-3] propose oFIB that an IGP extension should be a method against the occurrence of forwarding loops through manually managed changes that have to disseminate information over a connection or a router, since these have to be shut down for maintenance. The main reason behind this method is that the synchronization between the routers is used to restrict the order so that the routing entries can be edited on it. For the providers and that they do not want to change the router software, protocol extensions are difficult to implement (see [13, pp. 1-3]). Since this is not workable in the long run, new approaches have been created based on different operations. Protocol changes are impractical for certain reconfiguration cases, such as for single-link errors, because in these cases the forwarding properties have to be predicted [10, p. 4].

- Coarse-Grained Operation Scheduling: The most important thing about a method is that you can calculate the correct order without affecting the other network traffic. In [10, p. 4] optimization techniques should get the correct order for the macro-operations.

- Progressive Link Reweighting: Change link weights to avoid disruptions during a reconfiguration [10, p. 5]. There will be a detailed description in the section "Related Work".

- Ships-in-the-Night(SITN): At Ships in the night, the initial and final IGP configuration will run concurrently on each router in a separate routing process. With the correct order you can influence how much the network traffic can be disturbed during a reconfiguration [10, p. 5]. There will be a detailed description in the section "Related Work".

3.4.3 Software-Defined Networks

Here update problems related to Software Defined Networks(SDN) are covered. SDNs provide a separation between the controller (control logic implementation) and the application on the decision of the packets. A central control system can implement path calculation, which offers new flexibility [10, p. 3]. In an SDN network, a controller calculates and installs the respective rule on the devices about how the packets should be handled and forwarded. This saves the control logic on the routers and switches. No more messaging or routing path calculation is required on network devices. This prevents the slow reaction to changes and

updates and makes redirection in these networks with the controller (see [10, p. 6] [41, pp. 1-3]). Here, the controller can also become a weak point because it has to meet many requirements such as performance, firewall or packet processing requirements. There may be inconsistencies when changes are made to the network so the routing protocols would normally be solved by reconfigurations [10, p. 6] [41, pp. 1-3]. Therefore, for each event that requires a network update, the controller must install certain forwarding rules on the routers and switches.

4 Related Work

This section describes methods of existing technologies that can be used in practice. In this part, the focus will now be on methods that can be used in link-state protocols and distance vector protocols.

4.1 Link State Protocols

In this part, different methods used by link-state protocols to discover or avoid routing loops in a certain way are described. In [33, pp. 1-3] [35, pp. 2-4] they use link-state IGPs because they can be configured either in a flat or in a hierarchical mode.

4.1.1 Progressive Link Reweighting

This method tries to set the link weights during the reconfiguration without causing problems or routing loops in the network. In [11, p. 5] it is described that many research contributions concentrate on this, to minimize the number of link weights during a reconfiguration. The problem is that, despite the number of changes, it is not too computing-intensive. So you can say that you can easily apply this method to any router. In [11, p. 5] mentions that it can be shown that it is always possible to calculate a specific sequence of link weight changes that have also been shown to prevent transient routing loops on a link. However, there is also the difficulty here, if several destinations depend on the link, that the sequence is calculated without disruptions.

Increment by 1
In [14, p. 3] is shown that a link is increased by one. This only works if the routers in the network have small metrics. If the metrics in the environment are too high, increasing it by 1 may cause no significant change, unless one router has made its update, routing loops may arise (see [14, p. 3]).

Increment the links with higher number
In the above part, we have explained that in [14, p. 3] the increment by 1 was explained and its proof also made. The downside of this is that the networks are larger, increment by 1 does not make much sense. Therefore, in [14, pp. 3-5] that is in such a case, it is better if one can take a larger number. This is also

recommended in larger networks and is also proven in [14, p. 3]. This example should show how to deal with link weight changes in a simple network and to check whether this method works.

Example
To make this visible, we will look at an example at this point, which can also be found in [11, p. 5]. In this example, you see four different network states (a-d). You can see the current network configuration under point a. Now we want to edit the link between v3 and d or it is incorrect and an update is required. We want to change the router so that there is no routing loop when a change is made, so we will continuously change the link weights of the link (see [11, p. 5]). After we set the first link weight to 31 in the example, v1 will save d as the next hop. This can be seen in Figure 2 status b and how v1 changed the next hop. Next, we set the value to 51 and force v2 to enter v3 as the next hop. This also prevents the loops that could occur within the network (see [11, p. 5]). After the changes have been made, the link between v3 and d can now be shut down.

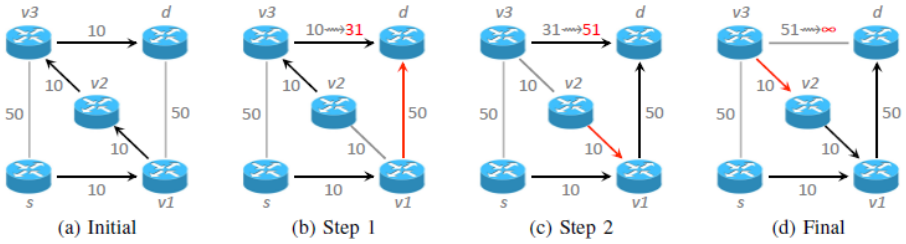


Figure 2: Link reweighting example. Intermediate IGP weights that enable a loop-free reconfiguration for the example [11, p. 5]

.

Proof of the algorithm
In [14, p. 4] you have the proof why incrementing of a link from A to B lead to loop-free convergence. It is also mentioned that you can also make a change and also can take a higher number for incrementing. Here, evidence was provided why changing in a low metric network does not cause problems within the network, although changes occur. It defines that routing loops can occur if the update is not carried out quickly enough. In [14, p. 4] you can see a detailed description of why this method can be proven and why it can always calculate a sequence that is also loop-free.

4.2 Distance Vector Protocols

In this part, different methods used by distance vector protocols to discover or avoid routing loops in a certain are described. With the help of distance vector

protocols, some methods are used to prevent loops within a network. One of the most important features in the routing protocol RIP is the "Counting to Infinity" method (see [7, p. 45]). In this, the routing protocols track the visited routers, which takes the packet from source to destination. If the defined maximum has now been reached, the packet is discarded. On the one hand, you want to see that there are no routing loops and that the packet does not stay too long within the network (see [7, p. 45]).

4.2.1 Split Horizon

With the help of Split Horizon, a routing update information is prevented from being sent back to the same interface where the route was learned. No route should be announced via an interface that the router itself sent to reach the destination. This prevents the hop count between two routers from being counted to infinity by the packet, always using the same connection [19, p. 77].

4.2.2 Route Poisoning

Route poisoning is a method in which a router marks networks that it can no longer reach with an infinite metric. This means that these routes are stopped immediately, instead of having to wait for the routers to expire. Therefore, routing loops can be prevented because of inconsistent routing updates via paths [19, p. 77].

4.2.3 Poison Reverse

This method is used when the network is stable and has no problems. If a connection fails and a router with an infinite metric route receives, then this router forwards to all connected interfaces. The special thing in it is also at the interface that is normally prevented by split horizon. The goal here is that every router receives this update of the route and no longer uses it, which is also called split horizon with poison reverse [19, p. 77].

4.2.4 Holddown timer

This timer is responsible if a router does not receive regular routing information from another router after a timer for a specific subnet has expired. In this scenario, this route is put into a so-called holddown status [19, p. 77]. Now the router ignores this period until the holddown timer has expired or until the respective router shows a better metric. Count to infinity problems can thus be solved since redundant paths can thus be prevented [19, p. 77].

4.2.5 Diffusing update algorithm(DUAL)

A loopless routing algorithm that is used with distance vectors or connection states and enables a diffuse calculation of a routing table. It works very well with multiple topology changes with little overhead [16, pp. 3-6]. Because you

want to achieve a loop-free connection to every instance in the network with DUAL, there is no need for another method such as the hop count of 16, which is used in RIP [5, p. 3]. In [17, p. 12] it is explained that in the event of node failures, the DUAL must involve the network nodes in diffuse calculations in such cases. One solution is that you have to wait a certain hold time to receive an update from your neighbors to determine the likely successor. During this hold-down time, you must choose the time so that the routers can receive and also manage the respective updates. To determine the hold downtime, it is necessary to estimate various factors such as the topology of the network (see [17, p. 12]). In [17, pp. 8-10] it is mentioned how to prove the correctness of the DUAL. It is mentioned how it's possible to guarantee that every instance is free of loops. The DUAL is used by the routing protocol EIGRP. If the topology changes, only the nodes that are affected will respond. The problem with this algorithm is that it is not open-source, but is used by Cisco in the EIGRP and only works on Cisco devices. Now the main features of the DUAL.

Routing States
A route can be in one of two states: PASSIVE or ACTIVE. These states describe whether a route guarantees freedom from loops (PASSIVE) or not (ACTIVE) [32, p. 10].

- Active state: A route is in the ACTIVE status if the neighbors do not meet the feasibility conditions. The lowest path cannot be achieved and there can be no guarantee that the path will be loop free. This router must coordinate with its neighbors to find the new loop-free path with the lowest total cost [32, p. 10]. If a router has a route with the status ACTIVE, it must not change its successor or its metrics until it changes back to PASSIVE. The updated information about the route that was sent during the ACTIVE status is temporarily stored or moved until it returns to the PASSIVE status [32, p. 11].

- Passive state: There is at least one neighbor who has a path with the minimum distance and the feasibility condition exists. With the feasibility condition, you can define that the path is loop-free [32, p. 10].

Feasibility Condition
This condition is required so that one can check the loop freedom of a specific path. Fulfillment of this check means that according to the definition it can be said that it is guaranteed to be loop-free. The feasibility condition is part of the DUAL and states that nodes that are not affected by the topology changes do not require a DUAL calculation (see [32, pp. 11-12]). After a change is made based on the result of the feasibility condition, the route can either remain PASSIVE or become active if no neighbor fulfills the feasibility condition and no inexpensive path has been found [32, pp. 11-12]. The feasibility condition is met if the neighbor has a shorter route than the router itself. In certain cases, take a neighbor who may not have the cheapest cost of the trail. This is possible if this path meets the feasibility condition and the current successor

can no longer be reached [32, p. 12].

DUAL Finite State Machine(FSM)
This mechanism is used to calculate all routes while tracking the shared routes from the neighbors. Distance information is used to find short and loop-free paths. DUAL selects the routes that are entered the routing table. This is based on a so-called feasible neighbor, which is used for inexpensive packet forwarding to a destination (see [32, p. 13]). If there are no such neighbors, but there are neighbors who reach the desired goal, then a recalculation must be carried out to determine who the new successor is. The DUAL tries to avoid recalculations since these affect the convergence time. If a topology change now occurs, the DUAL checks for feasible successors. If the algorithm finds one, it takes it to avoid unnecessary recalculations. The FSM is calculated for each destination in the routing table and a separate graph is also calculated to have a loop-free route to each destination in the event of an error (see [32, p. 14]).

Now different papers describe different ways to improve the performance regarding the convergence of the DUAL.

Diffusing Algorithm for Shortest Multipath(DASM)
With DASM, each router manages the entries for each destination that are stored in its routing table. These entries contain a series of information such as the next router and a loop-free route to the destination. A generalization of the diffusion calculation can guarantee freedom from loops (see [46, pp. 5-10]). A node has several next nodes for a specific destination and can also change them in the graph without creating a loop. In [46, pp. 1-10] it is shown that DASM has been defined, that you can act loop-free and based on simulations you show that you can keep up with the DUAL (see [46, p. 1]). This method prevents routing sleep by forcing routers to choose the next hop to destinations among neighboring routers that apply to all routers on the network. It is important that the designation of a specific destination by the neighbor is larger than your own (see [46, p. 3]). DASM offers similar performance for connection errors as DUAL and sometimes offers better performance. Results show that when connection costs change, it can perform significantly better than DUAL. You will find different definitions of different properties for the DASM algorithm in [46, pp. 2-9]. You can also see here why the algorithm remains loop-free in different situations and at the end, there is also a comparison with the DUAL where different measurement results are displayed.

Enhanced Distance Vector Algorithm(EDVA)
The paper [46] describes a new algorithm that DUAL does not use the loop-free mechanism to find the paths that are also loop-free. With the help of EDVA, the restrictions for a router are relaxed. Here the updates of the routing entry are not synchronized with its neighbors. If a router synchronizes its update activities with that of its neighbor, this level can be adapted to the network properties [46, p. 1]. EDVA is based on two different technologies. The DUAL will take

for the diffusing calculation for synchronization. Path finding algorithms are used to report the path information incrementally. The paper shows that there are quite a few advantages such as reducing convergence time and the number of packets for routing information time [46, p. 5].

More efficient DUAL
In [45] two different feasible conditions are presented, which improve the performance of the DUAL. The algorithm shows that it can be loop-free for every instance so that one does not have to do diffusing computations all the time when changing the network. The goal was to use the new algorithm to achieve a fast convergence time with less calculation. In [45, p. 1] they did some modification to the DUAL. The goal was to run a bit faster than the original one. A further proof shows that the modified DUAL will loop-free. The DUAL has already been described above and the most important components are local and diffusing computation. The selection of a successor, feasible successor and the calculation is also decided using the feasible condition [45, p. 2]. The paper defines two different feasible conditions such as RPFC and RAFC. There is a condition for each status, be it passive or active. One condition for passive router states is RPFC and this must be fulfilled by the successor to a router. The RAFC only comes into play when a router is active, so there is no neighbor who fulfills the feasibility condition [45, p. 2].

- Revised passive feasible condition(RPFC): If a router has to change the successor at a certain time, it can change it under a certain condition. This condition means that you choose a successor who has a minimum to achieve a specific goal (see [45, p. 2]). The exact definition of this condition can be found in [45, p. 2] and is described with an exact example.
- Revised active feasible condition(RAFC): If a certain router actively reaches the status at a certain time, it can define the neighbor as a successor if the feasible distance is smaller than before at a certain time. If there is no router in the area, the router remains a successor when it returns to passive mode (see [45, p. 2]). The exact definition of this condition can be found in [45, p. 2].

4.3 Ships-in-the-night(SITN)

Since we previously took care of the link weights and how to change them, we will now look at a method where we exchange the routing configuration. This method is called ships-in-the-night, where the initial and final IGP configuration runs simultaneously on each router and different routing processes (see [11, p. 5]). These routing processes are ranked based on administrative distances. You can see the definition of administrative distance in the upper area [11, p. 5] [33, pp. 5-6] [35, p. 4] . The route with the lowest administrative distance is saved in the forwarding table. An advantage of this method is that you have to configure it on the respective router. The routing protocols will then create the corresponding routes. The reconfiguration process in SITN is to swap the

preferences between the original and final configurations on each node. If we now want to use the final configuration, we set the administrative distance of the current configuration to 255. This means that we can also rule out that different processes are running. The next step is to wait for the network to converge again (see [11, p. 5] [33, pp. 5-6] [35, p. 4]). This can also lead to

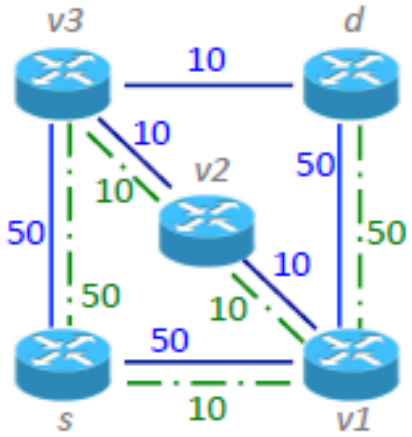


Figure 3: Ships-in-the-Night [11, p. 5] All routers run two routing processes, one with the initial configuration (blue) and the other with the final configuration (green).

inconsistencies, because if you have to keep in mind that the data traffic continues between the reconfiguration. And now it may happen that there are routing loops because the routers are updated without a certain order. Therefore, an order is required by converting the respective router into the final configuration without inconsistencies. Another way to do SITN is through the form of migration. In [42, p. 7] they have different reconfiguration strategies and theorem. This theorem handles the migration between two processes. An interesting theory that is mentioned is that it explains which different routing configurations can be used with SITN. Migrating from link-state to link-state (OSPF vs IS-IS) or distance vector (RIP vs EIGRP) works correctly. The interesting thing in [42, p. 7] that you can even reconfigure between the routing methods. In [42, p. 7] is also mentioned that the configuration can be exchanged between RIP to OSPF or EIGRP to OSPF. These different options can be migrated to SITN without difficulties and problems.

Figure 3 show the example which is mentioned in [10, p. 5]. You see the two different configuration options that run side by side. If there was an error, you can switch to the other configuration. However, be careful here that you

pay attention to the order, otherwise routing loops can occur.

Now two different algorithms responsible for calculating the order of the routers when making a change are described. This structured calculation of the sequence is important on the one hand, but the complexity of the algorithm is also very important. In [33, p. 7] [35, pp. 6-7] an algorithm is presented that takes longer to calculate in larger networks. However, [33, p. 7] [35, pp. 6-7] also offers an algorithm that should run more efficiently in larger networks.

4.3.1 Loop Enumeration Algorithm

The loop enumeration algorithm enumerates all the migration loops that can arise during migration. The output of this algorithm are constraints that ensure that no loop arise [33, pp. 7-8] [35, pp. 6-7]. The algorithm must find all migration loops for each destination. The algorithm builds a graph from the calculated paths. All different path combinations to a specific destination are calculated. Different cycles are then created, which can then be compared with the initial configuration and then influenced accordingly to the final configuration. In [33, pp. 7-8] [35, pp. 6-7] you can see how this algorithm was checked and why it works loop-free. In addition, rules are set up so you have a sequence to update the respective routers in the network without causing sleep. The algorithm has an exponential runtime. Because it depends on the different cycles in the graph and the cycles become more if more nodes are added (see [33, pp. 7-8] [35, pp. 6-7]).

4.3.2 Routing Trees Heuristic

The routing tree heuristic is the first step that the heuristic exploits a greedy procedure to compute a set of nodes that are guaranteed not to be part of any loop [35, pp. 7-8]. With greedy procedures it incrementally grows the set, adding a node at each iteration if and only if all the next-hops of the node in the initial and in the final configuration are already in the set. The algorithm is polynomial regarding the size of the input [35, pp. 7-8]. Three theorems are also shown in [35, p. 8] why this algorithm represents a more efficient option and how an order can be calculated.

4.4 Software-Defined-Network(SDN)

In pure SDNs you have a controller that calculates and installs the rules on the respective router so that the packets are forwarded over the network. This means that no messages are exchanged between the network devices that take place during routing protocols. With routing protocols, this message exchange is required to calculate the forwarding paths (see [10, p. 6]). This is solved with SDN. This also solves the problem that problems can arise if the network changes. This means that the controller can be provide different requirements

that can influence performance, security and packet forwarding. This also results in problems. With traditional networks, the routing protocols respond to updates or changes in the network with the help of routing messages. With SDN, the network operator would have to install the respective forwarding rules in the event of a change (see [10, p. 6]). Now it is crucial how the controller can react to a change so that the change is recognized and can be redirected to a new route using a specific mechanism.

4.4.1 Two-Phase Update

The two-phase update protocol is one way to get packets from an old to a new path. Failure to address this problem can cause path inconsistencies (see [24, p. 2]). The new configurations are installed on the internal ports of the network devices. The aim of this method is that the packages are tagged based on the version. With the help of the version number, you can now decide whether the package should take the old or new route [24, p. 2] [31, pp. 7-8]. To achieve this, the controller must install the new rules onto middle switches. After these rules have been installed, he informs the ingress ports of the network device so they sign the packets with the new version number (see [24, p. 2]).

4.4.2 ATOmic-MIP(ATOMIP)

ATOMIP is used to calculate a clear and overload-free migration plan so that SDN updates can be carried out. This means that congestion is to be minimized with the help of planned traffic management (see [23, p. 1]). The planning problem is formulated as a Mixed Linear Program(MIP), which is to create a schedule to prevent overload. The goal here is to find an order that causes minimal traffic loss [23, p. 1]. However, the calculation of the MIP is very time-consuming, especially for larger networks. For this reason, a heuristic called ATOMIP is proposed in [23], whereby the original update has to be broken down into smaller ones. Each of these minor updates is resolved by the MIP and is scheduled when it will be executed.

4.4.3 Architecture for consistent Network Updates

An architecture is proposed in [15, p. 8], which is intended to ensure that the consistency of rule updates is maintained. This is divided into three parts so that different problems can be dealt with like

- With the help of a generator, rules are created based on the guidelines (see [15, p. 8]).
- The update method selector: This part of the architecture comprises two steps. The old rules and collected data from previous updates are taken and a model of the current status is generated. Then several methods for applying updates are tested and simulated on the model of the network. The model is taken based on the results. There is also the possibility of

defining auxiliary rules that have not yet been defined in a ruleset (see [15, p. 8]).

- The update executor: The maximum value of updates that can be sent by the method is calculated here. Important details are the properties of the new, old rules and consistency. Faulty nodes only lead to a limited delay, whereby independent parts that are not affected can still be updated (see [15, p. 8]).

4.4.4 OpenFlow Safe Update Protocol

A protocol is described in [27] for updating OpenFlow networks. For this purpose, a delay in packet delivery is accepted so that the resources of the switch, in particular TCAM-space, are saved. There are different protocols that depend on the usage, which resources are the scarcest. You can then react to the respective situation. If compliance with the rules is important, this method is a good decision (see [27, p. 1]). The TCAM-space is saved because only one rule set is available on the switches. If there is a change and a network has to change the rule sets, the packets affected by the change are calculated and sent to the controller. The affected packages are buffered on the controller (see [27, p. 1]). After that, all switches with the new rule set are installed and only after all rules have been installed on the switches does the controller send the packets to the network again. The method, therefore, uses two different parts. It uses a technique to determine which packets are affected by a change and that of the controller which temporarily stores the packets and ensures the delivery of packets after the update (see [27, p. 1]).

4.4.5 Fast Lightweight Policy-preserving(FLIP)

This algorithm is supposed to guarantee the correctness of the forwarding and the input guidelines. In [40], the definition of FLIP, which describes an algorithm that creates sequences to replace and add rules, is nested. The goal is to ensure that updates are handled efficiently, securely and robustly. Different factors such as delays and changes in the network must be taken into account. FLIP calculates operational sequences to combine the efficiency of replacing the rules with the applicability of adding rules (see [40, p. 1]). There are restrictions when replacing and adding rules that should prevent each other from causing security breaches in the event of a change. A sentence is by exchanging the restrictions to prevent security breaches (see [40, p. 1]). After the algorithm is executed, a partial order of operations is returned that represents an operational sequence. In addition, further information is presented, such as replacing, marking and aligning rules. Another property is that the independence of the states with the help of the controller can reset to the previous status and the updates can stop (see [40, p. 5]). In addition, an assessment was made that FLIP reviews other approaches. It also shows that the WAN experiment created is more efficient in terms of memory usage than the 2-phase commit techniques. It also states that updates can be calculated faster (see [11, p. 15] [40, p. 13]).

4.5 Segment Routing

Multipath-Label Switching(MPLS) was one of the first alternatives to traditional weights and destination-based routing. With the help of MPLS it is possible to work with pre-configured routes and to work with push and pop. Over time, segment routing has emerged as a scalable alternative to MPLS networks (see [9, pp. 1-2]). With segment routing, no reservation of resources or states on the routers is required. This means that no virtual connections are created. In addition, segment routing has the advantage of being easy to deploy [12, pp. 1-2]. In [9] [12] one has now dealt with how to use segment routing networks with the help of rerouting algorithms to react to errors within the network. As mentioned before, re-computation after a network update is very slow and it affects performance. Now algorithms that were designed so you can react to link failures in segment routing are described.

Model
In this area, the model which is used and notation preliminaries which are necessary is mentioned. In this context, we will consider undirected graphs G = (V, E), which gives a certain order. The routing rules must be pre-configured and these cannot be changed during runtime [12, p. 4]. So you can't react during a mistake. Only routing rules that correspond to the next destination of the packet are allowed. The next thing would be to find a link that is already broken. Here the current node pushes certain labels on the top of the stack and acts as a backup path, so to speak. We now have to make sure that the backup paths have no forwarding loops (see [12, p. 4]).

4.5.1 Topology Independent Link-Failure Alternate(LI-MFA)

The concept of topology independence refers to the provision of a backup path after the failure of a link regardless of the topology [6, pp. 3-4]. The algorithm creates a data level for each destination within the network, whereby a switchover is activated in the event of a connection failure. Here we assume also that all link weights to be positive and have sub-paths of the shortest path which are also short (see [6, pp. 3-4]).

P and Q spaces
In this part there is the possibility to divide a path that is recalculated into a P and Q space. TI-LFA builds a so-called repair tunnel that comprises an outgoing interface and a list of the segments that are inserted in the header. This repair list has an explicit path to the destination. This path guarantees freedom from loops and a path that is independent [6, pp. 7-8]. The tunnel is formed if you connect the P and Q space with each other and then create the corresponding path from it. It is also ensured that neither of the two spaces uses a failure node or link (see [6, pp. 7-8]).

- P-Space: Is defined as the nodes whose shortest path from the source

which does not use the failure link. We used a list of the source and a point X that is not in the faulty route [6, pp. 6-7].

- Q-Space: Is defined as the nodes whose shortest path to the destination which does not use the failure link. Here we determine which nodes are going to the target after convergence, these are in Q space. This area comprises a list of members with defined links and nodes that lead to the goal [6, p. 7].

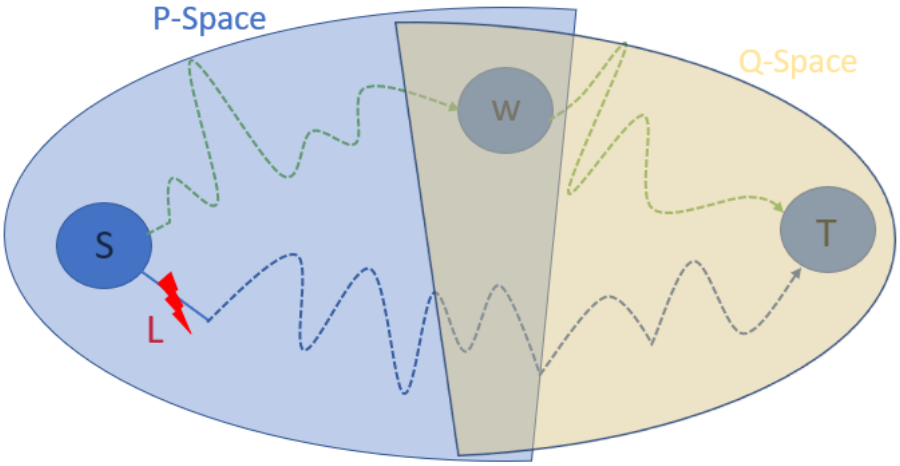


Figure 4: An example of a re-configuration and the division into Q and P space.

Figure 4 is supposed to show how one can make the division into a P and Q space. Here point w is used to separate the areas. An error occurs on the route L to T. In this case we calculate the routes in P and Q-space. The node q is now taken as the boundary and the P-Space calculates the shortest path to w. In the next step, the shortest path from w to destination T is calculated in Q-space.

TI-LFA loops indefinitely for two link failures
Now we have proven and shown that TI-LFA can handle an error very well. This area shows when TI-LFA is not operating correctly. In the Figure 5 an example is shown, which is also mentioned in [9, p. 3]. If we now imagine that the link between vm and t breaks. Then, thanks to the algorithm, vr is added as the next node on the stack. If it should break, as in this example, there is a loop between vm and vr. The problem with TI-LFA is that you cannot differentiate where the packet comes from, otherwise you could use the link el (see [9, p. 3]). To ensure that the packages still reach their destinations, a mechanism can be implemented that the link el is then taken. However, this requires that you increase the number of labels so you can push these links in the event of an error [9, p. 3].

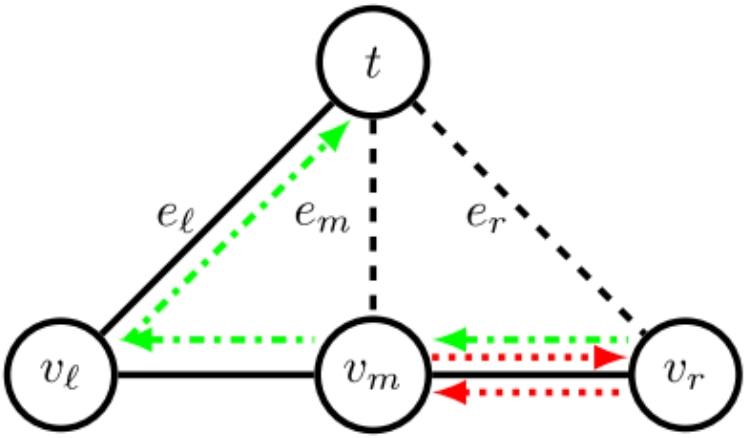


Figure 5: A network where you show TI-LFA with two link errors and a loop occurs. If the node between vm and em breaks, the backup path via er is used. But if this breaks too, there is a loop between vm and vr. TI-LFA will loop permanently between vm and vr and used the drawn dotted red line in the graphic [9, p. 3].

4.5.2 Topology Independent Multi-Failure Alternate(TI-MFA)

In [9, p. 4] describe how the algorithm works. The following steps are:

- Flush and empty the full stack except the destination node.
- Using the link errors in the packet header, a shortest path to the destination is calculated based on the current network.
- Finally, the corresponding label is saved in the stack. The node with the highest index on the route is calculated.

Correctness of TI-MFA
The correctness of TI-MFA is described in [9, p. 4]. The approach of Lakshminarayanan in [21], which describes Failure-Carrying Packets(FCP), is compared to TI-MFA. In FCP, a node decides which path to go next by taking the previous errors and the destination into account. It tries to make sure that the connection error is not taken twice [21]. This approach is like the hop-to-hop mechanism since you then decide the next point for each router. In the event of an error, TI-MFA behaves like TI-LFA by using the packet along a failure-free backup path. Now it is interesting how it is defined if the package detects x-1 errors on the path (see [9, p. 4]). There are two cases and have the assumption that the graph is connected:

- x = k: Because of the segment routing, the shortest path between G prime and G is identical again [9, p. 5].

- $x > k$: A link failure x + 1 is hit next, otherwise you can assume that you have reached the goal. Here, you call up the construction and therefore errors that have already been made can no longer be met. If all mistakes are made, we will reach the goal [9, p. 5].

5 Algorithm architecture

In this section, the problems which arise with the current EIGRP are described. In addition, what important properties the extension of EIGRP have to existing routing algorithms. The complexity of the architecture of the algorithm is also described. Another important point in this part is the description of what problem exists and what is the idea to solve the problem. The method already mentioned in the introduction is described in more detail and what you should pay attention to when setting up.

5.1 Problems statement

As already described above, we concentrate on the DUAL used by the routing protocol EIGRP. EIGRP is not the pure distance vector protocol but is rather referred to as a hybrid protocol. It uses different tables (neighbor, routing and topology table) to record the routing information. In addition, information is exchanged from the respective neighbors. An important feature of using this algorithm is that it guarantees loop-free access to every instance. This ensures that no package runs around in circles. To ensure this, the DUAL uses the status information and can be represent as finite state machine. Since the loop freedom is guaranteed, we have concentrated on testing another property and achieving this by expanding the protocol. To guarantee freedom from loops, other properties are neglected. Reliable packet transfer is not guaranteed, since packets can be lost. Other network policy properties such as waypoint enforcement are not supported. We have decided that with the help of extending the protocol with an idea that the property of the waypoint enforcement is guaranteed.

In the meantime, routing protocols are resolved in such a way that the metric decides which path to take. The shortest path is calculated and the packets are forwarded via this. If you want the packets to go through a firewall (waypoint) that is within the network, we have to do some reconfigurations. It may well be that the calculated path bypasses the firewall and thus no packet from it would be examined. When network security is important and damaged packets can have an enormous impact, it must be ensured that each packet is examined by the firewall.

5.2 Vision

Based on the problem, we want to ensure the waypoint enforcement network policy property based on EIGRP. Since the routing aims to take the shortest path, it is necessary to change the routing protocol to ensure that the packets go through the firewall. The goal of the extended EIGRP is now to guarantee the waypoint enforcement property. This is to ensure that the algorithm now calculates routes so that each of these routes goes through the firewall and can thus be ensured that every packet is checked. Every packet should go through the firewall when there is an update such as a metric change or link failure.

During the test phase, different routing protocols are configured using a network topology that has been created and tested using three experiments. In this phase, the data is recorded using PCAP-files or live recording with the Wireshark. This recorded data, which can be found in files, is visualized using Python and its libraries.

5.3 Conceptual contribution

To refer to the vision and to implement it, a corresponding concept is developed. This includes various decisions for implementation.

5.3.1 Use of distance vector

The title of this thesis also states that distance vector protocols should be used so that routing anomalies can be handled. We decided on EIGRP because this protocol is interesting of its properties and the algorithm. Therefore EIGRP is also defined as an advanced distance vector protocol or as a hybrid protocol. This protocol also offers a combination of distance vector and link-state protocols. The distance vector part is responsible for the exchange between two neighbors. This sends certain messages to the connected neighbors. These contain information about the networks and the costs at which they can be reached. As with link-state protocols, the routing information is stored in different tables and updates are only sent to the neighbors if there is a change in the network. This should reduce the network traffic somewhat and relieve the router's workload. All routes to the networks that were communicated by the neighbors are stored in a topology table. Only the best route is saved in the routing table.

5.3.2 Treatment of routing anomalies

We have already seen in the upper area that network anomalies can occur in the event of a network update. A common problem is that some routers process updates faster than others. This means that some update of the paths are faster than others. These differently processed updates can lead to routing loops. In this work, we have specialized because we want to guarantee the property of waypoint enforcement. By using EIGRP we have the property of freedom from loops, which is guaranteed by the DUAL. Because the extension that we are

planning is linked to metric changes and the connections to the links are set low and prioritized, freedom from loops can be guaranteed, since the paths that are calculated go through the waypoint. Another important fact, besides calculating the routes through the firewall, is that the packets, in the event of an update, do not mix the paths. This is to ensure that the packets do not take a different path when an update happens. With EIGRP, this is achieved by setting the router to active in the event of an update and continuing to use the old path until a new successor is recognized. This means that the paths do not overlap and when a successor is found who fulfills the feasibility condition, the system switches to the new path to the target.

5.3.3 Requirements analysis

To implement the vision and possible, an overview of the functionalities offered is provided based on a requirements analysis. The required requirements have been described, which are divided into functional and non-functional requirements.

Functional requirements
The functional requirements describe the functionalities offered by the extension.

- Simple extension: Since EIGRP is being built on and this protocol is being expanded, the aim was to offer an extension that can be achieved with the help of metric changes. The implementation can be expanded with further ideas. These extensions should then be able to be implemented on many simulators and real IGP that EIGRP have implemented.

- Handling of waypoint: The extension should extend the routing protocol so that a previously defined waypoint is included in the routing path. This node is defined before the configuration and is used to check the packets that are forwarded through the network. Here it is interesting to see how many of the routing protocols include this waypoint based on a network topology.

Non-functional requirements
With the help of non-functional requirements, various points are mentioned that can be measured during operation.

- Reliability: The extension protocol must bring the packets reliably to the prescribed destination without losing too many packets. Therefore, in the event of a change in the network, an appropriate path must be found and packet forwarding can be guaranteed.

- Transferability: Implementing the extension should be open so it can be extended and thus be able to see new ideas. The portability is now intended to ensure that the extension can be carried over to any simulator or real networks if they have implemented the EIGRP routing protocol.

- Efficiency: The efficiency of the modified protocol plays a big role in the comparison. By recording the network traffic, the results are analyzed after assessing defined metrics. Since the title of the work also refers to efficiency, but this can be tested using different properties, we tested how much a routing protocol uses the waypoint. We are less concerned with the speed of convergence or update time, but how many packets go through the waypoint. A trade-off is also entered because it can take a little longer to find a route to a destination, but the packages should be visited instead.

5.3.4 Use cases

Now that the requirements we have with our idea have been defined, use cases will now be described that produce certain results when this is carried out.

- Perform network experiment: A use case that is carried out for this work are simulations that are carried out based on certain experiments. In the beginning it is important that the created network topology is configured. This is done with the help of the knowledge gained and configured with different routing protocols. Three experiments are defined that are intended to provide answers to the different defined metrics. During the simulation, the data is recorded and repeated ten times so that corresponding outliers can be identified. With the help of an evaluation environment, the results are visualized and compared.

- Compare routing protocols: One goal is for routing protocols to be compared with one another using defined metrics. The extension of EIGRP is compared with the original EIGRP, OSPF and RIP and assessed based on the results.

- Record results: The results of the network simulation are recorded using PCAP-files and the results are recorded in files. These files can be analyzed by Wireshark or the files can be viewed by opening them. Several data are collected, as the individual experiments are carried out more often to analyze the differences in the results.

- Perform metrics analysis: Since we are comparing routing protocols with one another, it is important that a definition of different metrics are used to measure the data collected. It is important to determine which metrics are important for the respective experiments.

5.3.5 Design decisions

We used EIGRP because of the many properties that are also mentioned in this chapter. In addition, this protocol offers an interesting algorithm but also has a part as a distance vector protocol. As already described in the introduction, we have an idea how we would like to modified EIGRP so that waypoint enforcement is guaranteed. Metric changes are used, which are then used for the path calculation. This means that the network configuration would have to be made

so that the metric from the links to the waypoint are small. Because of the created network topology, it can be shown based on various experiments that EIGRP uses the waypoint for the path calculation with the idea used. Thus, changing metrics was decided, which means a little more effort for the network service provider, but can also be easily implemented.

5.4 Algorithm description

In this thesis we are interested on how packets going from A to B are checked by a node before they reach B. This node is called a waypoint, because the properties of the algorithm used are defined in such a way that they should take the waypoint. We need this because there is no routing protocol for tracking whether the sent packets have a corrupt content or cause problems. To achieve this, the forwarding paths must be calculated so that the paths send the packets via the waypoint. By using this waypoint it can be ensured that defective packets can be filtered out before they reach the destination. In addition, comparisons shown by simulations determine how individual routing algorithms include this waypoint.

An IGP is used within a network so that a packet can get from A to B. This regulates the forwarding to the next router with the help of the algorithms used. A link-state protocol uses a different algorithm than a distance vector protocol. Distance vector and link-state vector both aim to take the shortest route through the network. Because we want to research how routing protocols include a waypoint, there is often the problem that packets cannot find their way through the firewall node and arrive at the destination unchecked. That's why we researched a method for calculating the network paths so that the path goes through the waypoint.

In this work, we want to use distance vector protocols, because we are interested to look at network updates and their behavior on them. This is why the routing protocol EIGRP was considered because it is an interesting protocol from our point of view, because it offers the property of being free of loops for every instance. The DUAL is executed as it is implemented. The calculation of the shortest path, whereby the next router of this path is called successors. This means that the reported and feasible distance are important, as they are included in the path calculation. Since updates can also occur, the DUAL calculates a so-called feasible successor, which defines a backup path to a destination. The feasible successor must meet the feasibility condition, which states that networks are only accepted if the costs of my neighbors are lower than my costs to the destination. If this is the case, the respective node can be saved as a feasible successor in the topology table. Now we have taken EIGRP and its DUAL and tried that we offer the possibility that the paths that are calculated forward the packets via the waypoint. We said that this works best with prioritizing the links to this node. This is why the metric is set to a low level to ensure that the respective paths take the waypoint. Even in the event of

an update (metric change, link failure) it is possible to ensure that every packet is forwarded through the waypoint. To do this, the routing algorithm checks the respective metrics and then calculates the route to the destination. Because the links to the waypoint have a low metric, they are used for the route to the destination.

5.4.1 Properties of the algorithm

In this area, the most important properties that EIGRP and its extensions offer are described here.

Routing information tables
So that the information for routing has to be saved, EIGRP uses the following tables to record this information:

- Neighbor table: In this table, information is exchanged with the neighboring neighbors and stored in the table (see [37, p. 8]). To get this, Hello packets are exchanged to get the neighbor information. These also contain the IP address of the neighbor and the interface to which it is attached.
- Routing table: With the help of this table, the best routes to a destination are stored in the table from each source (see [37, p. 8]).
- Topology table: This table stores routing information learned by neighboring routers. This means that every router that can be reached by the respective node is saved. In addition, important data such as metrics, status of the route, successor or feasible successor are saved here. It is important to know that the topology table only stores routes are within the same autonomous system (see [37, p. 8]).

EIGRP metric
EIGRP calculates the metric from different values such as bandwidth, delay, load and reliability of a link. This data is required for the DUAL so that the metric of a connection between the nodes can be calculated. In [32, p. 42] you can see the calculation of the metric which is defined:

$$metric = [(K1 \cdot BW + \frac{K2 \cdot BW}{256 - LOAD} + K3 \cdot DELAY) \cdot \frac{K5}{REL + K4}]$$

The abbreviations of the formula are defined:

- BW: Designates the smallest interface bandwidth on the entire path (see [32, p. 41]).
- LOAD: The load describes how many packets go through this compared to others. This is shown as a value between 1 and 255 (see [32, p. 41]).
- DELAY: Describes the sum of the delays that occur on the interfaces along the path (see [32, p. 41]).

- REL: Reliability describes how reliably a link transmits the packets. This value is shown as a percentage between 0 and 100 (see [32, p. 41]).
- $K1$: Is used if the path selection is to be used based on the bandwidth that results along the path. EIGRP uses the bandwidth that is most available. However, this is adjusted through congestion-based effects (see [32, p. 40]).
- $K2$: If K2 is used, then the overload effect is reported as a measure of the load reported by the interfaces so that the maximum throughput is adjusted (see [32, p. 40]).
- $K3$: If K3 is used, then a path is selected that offers both delay and latency. EIGRP uses eigenvalues that are taken into account to calculate the bandwidth (see [32, p. 40]).
- $K4$ & $K5$: These values are used because there is a possibility that the path will be taken because of the connection quality and the packet loss. Loss of packets because of network problems or updates can lead to noticeable performance problems. This would lead to a problem, especially with streaming services, since packets are not transmitted quickly or not at all (see [32, p. 40]).

Finite state machine
EIGRP uses the DUAL to calculate the routing paths. Since topology changes can occur in a network, the algorithm can be described as finite state machine. With the help of the finite state machine, certain events are reacted to and a transition is made to the corresponding state (see [17, p. 135]). This comprises five states, one passive and four active states. These are necessary because routers can be changed in the network, even if they are active (see [17, p. 135]). A certain flag can ensure the status of the respective router. An overview of these different states and how to get to the individual states can be seen in Figure 6. In Table 2 you will find a description of how to get to the individual states and what the individual transitions mean. The o_j^i that appears in the graphic is the query origin flag and determines the current status of the node with the reply status flag. A further description can be found in [17, p. 135]. In Figure 6 you can see that if the network is convergent and runs without problems, the finite state machine remains in passive mode. The most interesting thing about this algorithm happens if there is a network change. The DUAL will perform the following actions:

- Recalculation of the metrics from the newly selected neighbor. The reported distance or connection costs were changed here (see [17, p. 134]).
- Check which of the neighbors offers a short total distance and fulfills the feasibility condition (see [17, p. 134]).

It is important to know before the router has been notified of the topology change that it was on a loop-free path. Because of the change, the router enters

the active status. Before doing so, he updates his reported and feasible distance to the current distance by the successor. The successor itself is not changed, as it is still guaranteed to be loop-free. Now maybe this path leads into a black hole, but because of the guaranteed freedom from loops it will still be used.

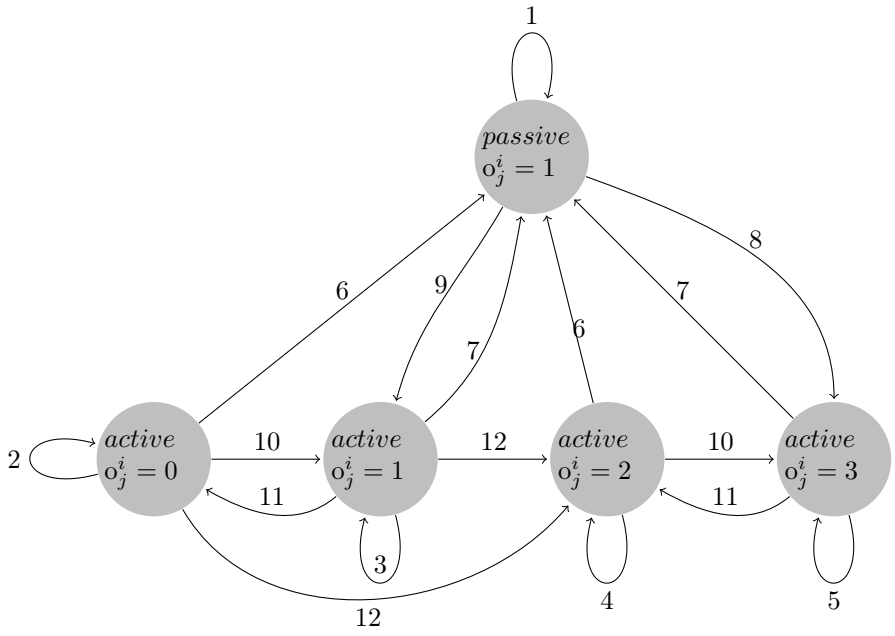


Figure 6: The finite state machine of the DUAL. It shows the different states the DUAL can have

The value of the feasible distance and the entries in the routing table could not be updated because of the active status. If all the answers arrive in the active states $o_j^i = 0$ and $o_j^i = 2$, then the search for a successor can be started, which is selected according to the smallest feasible distance (see [17, pp. 134-135]). By using the feasible distance, the topology changes that occur during the active state can be recorded and delayed until the active state is completed. This gives the router the opportunity to become passive if a successor is found or to carry out a diffuse calculation (see [17, pp. 134-135]).

Waypoint enforcement
With our expansion, we want to guarantee the ability of waypoint enforcement. The point where all packets are checked is defined in advance and serves as a so-called waypoint in the network. The exciting thing about this property is that by defining a point, every packet should go through this node (see [22, p. 275]). This node serves, so to speak, as a checkpoint for every packet that goes to a destination. In [22, p. 275] the route is abbreviated as π. In particular,

Input event	Description
1	Remains passive if the feasibility condition has been met. So a successor could be found.
2	Another change was detected in the network since the last response or query from the successor.
3	The network event differs from the reply, query from the successor or change of distance. Thus there was a change that occurred in the network, but different from the previously known.
4	A different change was detected in the network than the last answer showed.
5	The network change is different from the last response sent or involves increasing the distance to the destination.
6	All sent messages have been confirmed with a response and with the current feasible distance, the feasibility condition n is also met.
7	All messages have been confirmed with a reply and the feasible distance is set to infinite.
8	A request was received from the successor and the feasibility conditions are not met with the current metrics.
9	There is a change in the network as the query of the successor, where the feasibility condition is not met.
10	A response was received to all messages, but the feasibility condition for the current feasible distance is not met. So to speak, the current metrics do not meet the feasibility condition.
11	The distance to the destination has increased.
12	The query was received from the successor.

Table 2: Describes the input event of the various statuses and the description of the events. (see [17, p. 135])

the routes are defined as $\pi 1$, $\pi 2$ and stand for old ($\pi 1$) and new route ($\pi 2$). To achieve the principle of waypoint enforcement, both the old and the new route must be calculated so they pass through the waypoint in the network (see [22, p. 275]).

To achieve this, the links to the waypoint are prioritized and the metrics are adjusted accordingly so that the router takes the route to the waypoint. The user sets the metric correspondingly smaller so that the DUAL uses it for the calculation.

No packet change
We wanted to extend the algorithm in such a way without having to change the

package itself. That would mean too much effort, since every packet would have to be viewed and the header used. To prevent this from happening, we didn't want to change the package because that would mean additional work. Because by expanding the header, each packet must be expanded and then searched for the respective field in the header. Because every router would have to check it. This can also result in a delay in the packets. That would slow down network traffic.

5.4.2 Complexity

This area describes the time it takes for the algorithm to analyze how much effort is required for the network to reach a convergent status. Since it can happen in the network that the nodes execute the algorithm differently, the measurements are based on the assumption that the nodes execute the algorithm synchronously (see [17, pp. 140-141]). The respective nodes in the network execute the algorithm at a specific point in time. When the algorithm is executed, packets are sent and processed on the respective node. If a node detects a change in the network, it sends this change to the neighbors in a packet. This is used to calculate the performance of an algorithm, since it calculates the number of steps until convergence (see [17, pp. 140-141]). It also takes communication complexity into account, which dictates how many messages are needed during a change. In the event of a change (failure of a link, metric change), the time complexity can be reduced to O(n), where n is the number of affected routers. With DUAL, not all routers are included in an update process, but only those that are directly affected. The communication complexity is O(6*D*n) after a change in the network. D stands for the level of a node that sends different messages (query, response and update) to the neighboring nodes (see [17, pp. 140-141]). The comparison with different algorithms is also shown in [17, pp. 140-141]. It is also mentioned that adding a link or decreasing the metric leads to a deterioration in complexity. This mean that a feasible successor must be found who fulfills the feasibility condition.

5.5 Summary

In this chapter we have described the problem of routing protocols with the property of waypoint enforcement and the solution we want to counteract this problem. In addition, the most important properties of the routing protocol are described and their details are discussed. The current properties such as the description of the DUAL by a finite state machine and why there are so many states were discussed here. Further properties such as the extension of the algorithm and how we wanted to achieve this are also described. We also describe the method with which we would like to achieve the results. A main part of the work is the simulation and comparison of different routing protocols and how they react in the event of an update. Another point is the complexity of the algorithm, because it is important that a network converges as quickly as possible and in the event of an update, converges as quickly as possible

without routing anomalies. The next chapter of the implementation deals with implementing the changes to the algorithm presented in this part and details are given why this was adopted. The implementation of the EIGRP routing protocol in the ANSARouter module was built upon.

6 Implementation

This area describes the most important points of the implementation. There are different highlights that make up the implementation given. Below, the aforementioned properties of the implementation are summarized. In addition, different implementation decisions are discussed and why the implemented idea is so advantageous to the solution which are already existing. The disadvantages that arise from the implemented idea are also dealt with. Details on the configuration are mentioned and what must be taken into account when using the extension. There are also alternatives to the chosen method and why there might be problems with its methods named.

6.1 Overview

The EIGRP router module of OMNET++ is now to be expanded by building on the EIGRP implementation and implementing the method of prioritization mentioned in the architecture through metric changes.

A network topology is created that is configured with an IGP, in our case it is EIGRP. Then we configured the router using an example that you can find under the Figure 10 and made it run. After we have got it running, the corresponding routing paths are calculated using the DUAL and depending on the setting, ICMP packets are sent from PCA to PCB at a certain interval. Now you can see which packets the respective routers exchange with each other. The routing paths are determined and the packet can take a path to its destination. We have implemented it in such a way that both ping and one-way UDP packets are sent so they can also be analyzed later.

To achieve this, the various tables are now created on each router and with the help of the Hello packets, these are now initialized and can now calculate the paths. How this happens is described under "Functionality & Core points".

6.2 Implementation decisions

In this subsection, the description of the decision made for the implementation is given. Important implementation decisions are explained. Since different simulators are used, the technical aspects of those that were used in the implementation are described. There are also alternatives mentioned we thought through while working on the thesis.

6.2.1 Technical details

To set up the network topologies and evaluate the results, certain tools such as OMNET++ and the ANSARouter module were required. You can see these details about OMNET++ under the Table 3. Since GNS3 was also used to

Technical aspect	Description
OMNET++ version	5.0
ansainet version	3.4.0
EIGRP router type	ANSA_EIGRPRouter
OSPF router type	OSPFRouter
RIP router type	Router

Table 3: Technical decisions made for implementation and simulations with the OMNET++ simulator

evaluate the results and also offer an implementation of the EIGRP routing protocol, we also used this simulator to get and analyze the results. Details on GNS3 can be found under the Table 4.

Technical aspect	Description
GNS3 version	2.2.8
Wireshark version	3.2.3
Cisco router version	c2600

Table 4: Technical decisions made for simulation with the GNS3 simulator

6.2.2 Structure in OMNET++

We therefore decided on GNS3 and OMNET++ because they have a configuration for the routing protocol EIGRP. That's why we could plan, test and analyze our planned methods here. This area is now about the EIGRP structure that was implemented in OMNET++ with the help of the ANSARouter module. You can see this structure in Figure 7. This structure provides the various tables that EIGRP uses to store routing information. In addition, the authors have implemented a configuration for both IPv4 and IPv6 in [37]. In this thesis we have only focused on the IPv4 area. Besides the tables already described, there are other sub-modules:

- eigrpIpv4Pdm: This module is called the protocol-dependent module(PDM) and is used so that EIGRP messages which contain important routing information are exchanged. It is also important for the exchange between the routing table and the topology table (see [37, p. 8]).
- eigrpRtp: The RTP is used so that the routers configured with EIGRP can communicate (see [37, p. 8]). EIGRP uses its own solution with RTP

instead of UDP and TCP to exchange messages with each other. There are different packet types with Update, Hello, ACK, Query and Reply. This ensures that reliable transmission can be guaranteed (see [37, p. 8]).

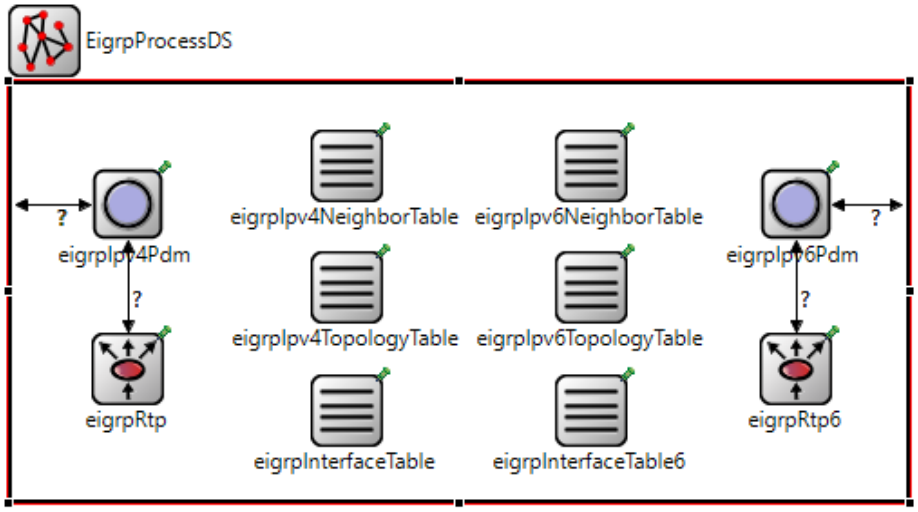


Figure 7: EIGRP model structure of the ANSARouter module [37, p. 8]

6.2.3 Configuration in OMNET++

We mentioned in the previous point that the configuration is loaded from a config file. In Figure 8 you can see an example of an excerpt from a config file. In Figure 8 you can see how to configure the individual interfaces, so it is advisable to make an interface plan beforehand so you know how to configure the topology correctly. Now the config file begins with a root element, as in my case devices. Then every router can be called up with the help of an id, as with s1 under 10.0.1.0. Then the interface name, for example eth0, is used and this is provided with an IP address and subnet mask. In this case, we decided on a /30 (255.255.255.252) subnet mask between two routers because we didn't want to waste the address space with the help of subnetting. The values of the metric can also be set, such as a delay in this case, where it is said during configuration that the value of the delay assumes a value such as 1001 in this case.
Figure 9 shows the second excerpt from the configuration of a router. This shows the routing process that determines the IP address using the routing protocol EIGRP, in this case IPv4. Here you can see how the various networks are listed and the associated wildcard mask. It is also possible to configure a passive interface eth2. This means that no routing updates will be sent there, as it is going to a PC that cannot manage routing updates.

```
<!-- s1 -->
    <Router id="10.0.1.0">
        <Interfaces>
            <Interface name="eth0">
                <IPAddress>192.168.0.1</IPAddress>
                <Mask>255.255.255.252</Mask>
            </Interface>
            <Interface name="eth1">
                <IPAddress>192.168.0.5</IPAddress>
                <Mask>255.255.255.252</Mask>
                <Delay>1001</Delay>
            </Interface>
            <Interface name="eth2">
                <IPAddress>10.0.1.1</IPAddress>
                <Mask>255.255.255.0</Mask>
            </Interface>
        </Interfaces>
```

Figure 8: Shows the inteface configuration in OMNET++ from the router s1

6.2.4 Further decisions

The choice of the programming language came about quickly, because OMNET++ is based on C++, the choice for the extension also fell on C++. There was no reason to add any extra classes because it was built on the implementation that is already implemented in the ANSARouter module.

There was also the possibility of an updated OMNET with the connection of the ANSARouter module, but there were a few compilation errors and stayed on OMNET5.0 because there weren't any big differences.

Another decision that was made is that only IPv4 for the routing through the routing protocols during configuration is used. The inclusion for IPv6 would not be a problem, since the implementation also has a module for IPv6.

Another decision that was made for the implementation is that the links that go to s3 as the waypoint are kept low so they are taken into account for the paths. These paths to s3 are treated with a higher priority and are therefore selected by the DUAL as intermediate path destination.

6.2.5 Alternatives

In the previous area, the selected method was chosen to expand the algorithm, but alternatives to this implementation have also emerged. This alternatives in the area and were the following ideas:

- Package change: Here we add a flag to the packet header, which is set when the packet has visited the waypoint or not. If the flag is not set,

```
<Routing>
    <EIGRP>
        <ProcessIPv4 asNumber="1">
            <Networks>
                <Network>
                    <IPAddress>192.168.0.0</IPAddress>
                    <Wildcard>0.0.0.3</Wildcard>
                </Network>
                <Network>
                    <IPAddress>192.168.0.4</IPAddress>
                    <Wildcard>0.0.0.3</Wildcard>
                </Network>
                <Network>
                    <IPAddress>10.0.1.0</IPAddress>
                    <Wildcard>0.0.0.255</Wildcard>
                </Network>
            </Networks>
            <PassiveInterface>eth2</PassiveInterface>
        </ProcessIPv4>
    </EIGRP>
</Routing>
</Router>
```

Figure 9: Shows the network configuration in OMNET++ from the router s1

then it must not go to the destination. This brings with it two problems, each packet and its header have to be extended to one. Every router must check this packet for the respective header. If the routing paths are calculated incorrectly, delays or packet loss can occur, as the packets are only allowed to arrive at the destination after they have visited the waypoint.

- Delete unnecessary neighbors: To achieve that the waypoint is reached, neighbors cannot be inserted in the algorithm if they would bypass the waypoint. In this way you can influence the calculation of the routing paths and ensure that the waypoint is used. This means that no packets go through these neighbors and are not taken into account when calculating the routing paths. The problem with this method is that although the paths can go through the waypoint during the calculation, errors can occur in the event of an update if no alternative can be calculated.

- Run the algorithm twice: Here the DUAL is done in duplicate. The goal would be that the algorithm first calculates the path to the waypoint and then calculates another path from the waypoint to the destination. Another solution would then be found, since each router would look for the shortest path to the waypoint and thus send each packet through. One problem with this would be that the runtime for the algorithm would be twice as long as it would have to be executed twice.

- Using a controller: The idea here would be to use a controller, similar to SDN, to regulate packet forwarding and ensure that every packet goes through the waypoint. The controller then sets the paths so that every packet from the source to the destination is checked.

6.3 Functionality & Core points

The individual configurations, which are saved in an XML format, are called up at the beginning. After the config-file of the respective router has been read in, the tables (interface, neighbor, topology) are initialized and processed according to the configuration in the config-file. After the initialization has ended and every router knows which interfaces and networks it has, messages are now exchanged, so-called Hello messages, which are used to get information from the neighbor and the respective tables are updated. These messages are used to ensure that every router knows that the neighbor is still alive and to exchange routing information in the event of an update. With EIGRP, a Hello message is sent every fifth second. After these have been exchanged and every router now knows who the respective neighbors are, routes to every destination can now be saved to the topology table, with only the best routes saved into the routing table.

6.3.1 Core points

After we have described the functionality of the EIGRP process, now come the most important points important for using the routing module:

- Tables: With the help of the tables (interface, neighbor and topology), different information can be exchanged, which is important for the path calculation.

- Path calculation & updates: The path calculation and the case when an update occurs is also a feature of the DUAL. Because the DUAL can be displayed like a finite state machine and offers different states for the active mode, it is very interesting to analyze when a change occurs in the network. The feasibility condition also checks whether the path can be declared as loop-free or not. Only neighbors who meet this condition may also serve as successors.

- Metric calculation: EIGRP takes various values for calculating the metric of a link. These values have already been mentioned in Section 5. We used that to help and put the links that go to the waypoint small accordingly. We want the DUAL to calculate the paths so that this waypoint is included. This serves us the method that we support the property of waypoint enforcement.

6.4 Advantages & Disadvantages

In this subsection a description of the advantages and disadvantages from the extension of the idea which is used for this thesis. These emerged from the implementation.

Advantages

- Easy implementation: Prioritizing the links that go to the waypoint, which occur through metric changes, guarantees a simple implementation of this method. To do this, see where the waypoint is and with which neighbors it is connected.
- Applicability is given: Because no major changes were made to the implementation, it can be used easily with no problems.
- No major changes required: Because some links to the waypoint are prioritized based on the metric change, no serious changes to the implementation of the routing protocol are necessary. The existing modules were used in the implementation of OMNET++ and did not need to be expanded with other modules.
- Waypoint enforcement: We have found a method for how we can achieve the property of waypoint enforcement in that each packet goes through a waypoint and can do this for both new and old paths.

Disadvantages

- More effort for the administrator/ISP: The administrator/ISP is responsible for the network configuration and that the packets are efficiently forwarded from A to B. Now it is the case that a node of the waypoint is to be included in the route calculation. Because changing the metric, the network administrator/ISP has to set the links so that the route goes through the waypoint. This can lead to increased effort, as the network administrator has to take this into account.
- Not always efficient: The main goal of a routing protocol is that the lowest/best path from the source to the destination is calculated. Because we want the path to go through a certain node, it is not always guaranteed that it is the best path.

6.5 Summary

In this chapter we mentioned the most important points about the implementation and why the decisions were made as they were implemented. It was described how the network topology was configured and how the implementation is carried out. In addition, alternatives were named and the advantages and disadvantages of the chosen extension. By changing the metric, the property of the waypoint enforcement can be ensured, as the links are prioritized around

the waypoint and set accordingly low. This makes it very certain that the route goes through that point. To ensure this, the simulations are now carried out and analyzed. The change in metrics from EIGRP is compared with other routing protocols and the results are checked using three experiments. The description of the experiments and the evaluation of the results follows in the next section.

7 Evaluation

In this area of work, the results of the work are described. Here the simulators and which ones were used for implementing for the results of the work are mentioned. There is also an analysis of the different results and compare them with the existing routing methods.

7.1 Structure of the test environment

This subchapter describes a created network topology that you can see in Figure 10. You will see seven nodes where the nodes of s1-s5 should represent routers and the other two PCA and PCB are normal user devices or laptops that want to communicate with each other. It is important that point s3 is defined as a waypoint, i.e. as a firewall and that the packets are forwarded via the paths. The paths are calculated by the individual routing protocols themselves and forwarded to a destination using the shortest path. In the created experiment

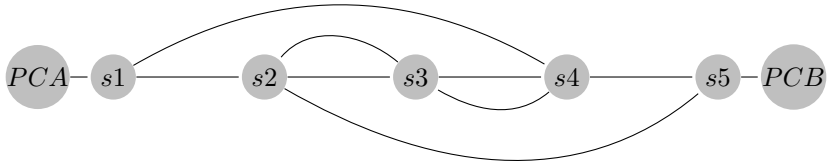


Figure 10: Graph with target PCB where point $s3$ represents the firewall and the packets from PCA start and are forwarded via the best calculated path.

run the various routing protocols (EIGRP, OSPF, RIP) and check them against the various metrics. There is a description of the methods used to check the respective metrics. The results of the various measurements that occur during the simulation are recorded using tables. The metrics that are used for the measurements of the simulation:

- How many packets go from the source (PC1) to the destination (PC2)?
- How many packets of the sent packets are lost?
- How many packages go through a previously known waypoint?

- How long does the routing protocol take until the update has been processed?
- Do the link-state protocols process the updates faster and more efficiently than distance vector protocols?

7.1.1 Configuration of the experiment

The graph from the experiment has already been shown and described in the upper area. This point describes the configuration of the individual routers, which can be reconfigured accordingly. This means that every user can understand the results and try out for himself how the packages work.

Table 5 shows how the experiment is configured and the individual routers. It shows which addresses and subnet mask we used to configure the network. You can now configure the interfaces on the respective routers as shown in the Table 5. In addition, the various routers and which interfaces they use are displayed. It is important that a /30 (255.255.255.252) subnet mask is used for every hop to hop configuration, as two IP addresses are needed for these respective connections. This is especially important in larger networks for saving IP addresses. This can be solved with the help of subnetting, where a large address space can be divided into smaller address spaces.

No.	eth0	eth1	eth2	eth3
s1	192.168.0.1 255.255.255.0	192.168.0.5 255.255.255.252	10.0.1.1 255.255.255.0	
s2	192.168.0.2 255.255.255.252	192.168.0.13 255.255.255.252	192.168.0.9 255.255.255.252	192.168.0.30 255.255.255.252
s3	192.168.0.17 255.255.255.252	192.168.0.14 255.255.255.252	192.168.0.26 255.255.255.252	192.168.0.29 255.255.255.252
s4	192.168.0.18 255.255.255.252	192.168.0.6 255.255.255.252	192.168.0.21 55.255.255.252	192.168.0.25 55.255.255.252
s5	192.168.0.10 255.255.255.252	192.168.0.22 255.255.255.252	10.0.2.1 255.255.255.0	

Table 5: Router configuration of the experiment

7.1.2 Methodology

As already mentioned in the introduction, various measurements are carried out on a network topology and different routing protocols are compared with one another. When the simulation is started, it is checked how the individual protocols and their algorithms calculate the paths through the network and how the packets are forwarded. A network topology is set up in GNS3 and OMNET++, which you can see in Figure 10. After you have built the topology,

you can configure it using the router configuration of Table 5 to understand the results of my thesis. A corresponding scenario will also be developed to view the behavior of the routing protocols in the event of an update and how long they need for processing to achieve a convergent state again. An important detail here is that the extended routing protocol also takes the waypoint and it is important to consider how many packets go through s3. The simulation is now observed by recording the network traffic on the corresponding nodes and can be analyzed. Then the numbers are assessed and the results of the routing protocols are compared with each other, explaining why they came about in this way. Various metrics can now be examined based on the results. These deal with packet forwarding in the individual routing protocols and the behavior of this forwarding in the event of an update. The results of the simulation are recorded in tables. Since simulations can lead to different values in the results, the individual simulations are also carried out several times.

7.1.3 Data acquisition

So that the various metrics can be checked, we used the simulators GNS3 and OMNET++ for this thesis to show by simulations whether the metrics can be adhered to. Data is checked with Wireshark and OMNET++ in there are also different ways to see, for example, whether packets are passing through a certain node. Because the simulations run, we get a lot of data that is recorded with the help of PCAP-files or analysis of the links. The routing protocols (EIGRP, OSPF, RIP) will run on the same topology and test which path the packets take and whether all packets arrive successfully and how many packets go through the waypoint. By logging the packets, the individual metrics can now be researched, so individual packets can also be researched using Wireshark and with the help of that tool, can also be filtered for packets. The various mentioned metrics are recorded in percentages and tables so you can see how the individual routing protocols behave. In addition, these tables show comparisons to the other routing protocols and how they react to a waypoint in the network.

From my point of view, the measurements and comparisons have a practical relevance, since they offer comparisons of the various routing protocols and how they relate to an update and a waypoint in a network. In networks it is important that it forwards the packets to a destination using the cheapest and shortest path, but it is also important that in the event of an update this happens efficiently and without anomalies. Here it is also important that the update is done as quickly as possible, without routing anomalies such as routing loops or forwarding on the wrong path within the network.That is why it is important for us to know which packages that are sent arrive and how long it takes to process an update and how a waypoint is included in the path calculation.

7.2 Simulators

A simulator is used to implement the master thesis. Two simulators were used to test the routing protocols and evaluate the results. These were GNS3 and OMNET++. This two simulators were selected from the large pool because they implemented EIGRP and its DUAL. To extend the protocol and the program code, OMNET++ was used, where one could see the implementation of the routing protocol. In addition, it offers various options for testing scenarios such as link changes and link failures during the simulation. In addition, the extended EIGRP is compared with the existing EIGRP and other routing protocols.

7.2.1 Graphical Network Simulator-3(GNS3)

With the help of virtualization, people are not necessarily dependent on physical hardware. Now you no longer have to rent a room or network laboratory, which on the one hand can become impractical and can also restrict resources for planning a network. Earlier software simulation programs are often restricted to the network design and can only be used to a limited extent(see [30, p. 3]). GNS3 lets you create network topologies based on requirements and you can add unlimited objects [30, p. 2]. GNS3 is a graphical network simulator that runs on many platforms (Windows, OS X and Linux). With GNS3, a MIPS processor simulation program is also used called Dynamips. The router operating systems can be run in this simulation program. In addition, GNS3 brings a graphical user interface where a network can be set up by simple drag-and-drop. The simulator is also freely available for every platform. With GNS3 you can design and test virtual networks on your PC [30, p. 1]. With the help of Dynamips-Hypervisor-Program different router hardware from Cisco can be emulated. These different network devices can thus be configured and the connections tested [30, p. 5]. VirtualBox virtual machines that can represent a host or server can also be added. These virtual hosts can be connected to the network devices of the topology. This means that a complete end-to-end network can be created and tested [30, p. 5].

Problems with GNS3
The networks can be designed and configured easily. Link-stat-protocols such as OSPF or distance vector protocols such as RIP or EIGRP can also be configured. The exciting thing is that you need real router images to configure the router. So you work with real images, which is also very good to experience. With the help of Wireshark, you can also see the packet flows and lost metrics. We worked with pings and traceroute, because when changing the network, we see how long it takes for the network to converge. You can also see lost packages with the help of Wireshark. The problem with GNS3 is that it has implemented the existing routing methods, however, since we want to modify EIGRP but cannot extend the routing protocols in GNS3.

7.2.2 Objective Modular Network Testbed in C++(OMNET++)

OMNET++ is a C++ based simulator for creating networks. It is an open-source program which can be used free with the help of the Academic Public License. An attempt is made to provide an interface between free and commercial simulators. OMNET++ is provided for different platforms such as Linux, Microsoft. Simulation models from different areas can be developed, including different IGP routing protocols [39, p. 1]. With the help of a NED-file, the user can define a structure of the model. This can be created with two options the graphical editor and the NED file itself. If the graphical editor is used, the NED source code is generated automatically [8, pp. 27- 28]. Different components of the network can occur, such as the gates, parameters to connect the individual components. The various network devices such as PCs, routers and switches are also defined here [39, p. 2]. Since the NED files can be saved in an XML representation, they can be converted into an XML document. In this way, different manipulations of NED files can be programmed and thus stored in SQL databases and reused [39, p. 3]. Besides a .ned file where the topology is described, a configuration file is also required that defines how the simulation is carried out. A configuration file called "omnetpp.ini" should be created for each simulation. Without this file, the simulation would not start. Here you can specify how the different network devices are to be configured and when using a ping you can also specify here [8, pp. 31- 33].

Automated Network Simulation and Analysis(ANSA)
ANSA is a project created and implemented by the Brno University of Technology in the Czech Republic. This project tries to develop a variety of simulation models and extends the IP network framework INET. It is important to mention that INET offers an extension for OMNET++. With this extension, different protocols and models are offered and are interesting for people who work with communication networks [1]. The concept of components means that they can be easily understood and further programmed. With this extension it is possible to simulate different routing or switching protocols and to examine certain properties like faulty configurations or network states [1].

One aim of the ANSA-project was to offer several dynamic routing protocols through router modules. Thus it is possible to implement a variety of routing protocols using ANSARouter that simulate the behavior of Cisco routers [38, p. 4]. Because an EIGRP module was developed with this project, it was used to simulate the networks that were created. Another important detail of this simulator is that analyzes of the network configuration can be carried out [38, p. 1]. You can get more information from the ANSA source code at [36]. Here you can see the implementation of the different routing protocols and also other different network structures such as MPLS and many more.

7.3 Tools evaluation

This area summarizes the different tools which are used to test the configuration and check the results. These tools are used to get the results and to assess them. The following methods were used with the help of the OMNET++ simulator to get the results.

7.3.1 Ping

Ping is used to checking the availability of a specific address. ICMP-echo messages are sent to the destination. For this purpose, the packets are forwarded along with the respective router, along the shortest path. To complete the ping, a reply is sent so that the accessibility can be checked (see [25, p. 12]).

Figure 11 shows how to ping a source to a destination with the OMNET++-simulator. PingApp module, which offers the functionality of a ping, only that several target addresses can also be pinged, was used. Here the ping to go from PC1 to PC2. The destination is set with the parameter "destAddr". This ensures that not all points ping, only PC1. With the help of PingAPP the packet loss and the round trip time are calculated. This allows various statistics to be created and then analyzed. Last but not least, we chose the "printPing" parameter so we can track which specific routers take the ICMP packets.

```
*.PC1.numPingApps = 1
*.PC1.pingApp[0].destAddr = "PC2"
*.PC1.pingApp[0].printPing = true
```

Figure 11: Configuration of a ping in OMNET++ from PC1 to PC2

7.3.2 Traceroute

With traceroute a packet with a TTL of 1 and the next time it is sent will always be sent one. A packet is sent to the first router with a TTL of one. The router subtracts the TTL by one and recognizes that the packet has expired and sends it back to the source (see [26, p. 2]). Then a TTL of two is taken and sent the packet to the second hop of the path which then recognizes again that the packet has expired and is sent back to the source. This process is then repeated until the specified goal is reached. Traceroute is used to track the user on the path taken by the package (see [26, p. 2]).

Traceroute is not explicitly implemented in OMNET++, but different properties that traceroute offers can be copied. On the one hand, the PingApp module can be used to define the "hopLimit" property to track the packets step by step using the number of the router. The path itself can be followed with

the help of the simulator by visualizing the path and the respective packages and thus making them visible.

7.3.3 One-way UDP packets

Another way to track the path to a specific destination is to use UDP packets without sending a response packet. This allows further statistics to be kept on how many packets run over a particular router or not. With this method, you can ensure that some routers are visited more than others and that there are not too many packets in the network, since additional responses are expected, such as with ping or traceroute.

Figure 12 shows how a packet is sent from PC1 to PC2. We use UDP packets for this, since they also do not require an answer. With the help of OMNET++ this can be done with UDPBasicAPP. UDPBasicAPP allows packets to be sent to a determined port and like with PingAPP, the interval for how the packets are sent can be set. You can see how packets are sent from PC1 to PC2 and which path they take. The UDP-module UDPSink on PC2 is also used because it prints the messages that are sent via this module. Here it is important that both destination and local port are the same so that the respective UDP packets can be tracked.

```
**.numUdpApps = 2
**.udpApp[0].typename = "UDPBasicApp"
**.udpApp[0].destPort = 1234
**.udpApp[0].messageLength = 32 bytes
**.udpApp[0].sendInterval = 0.1s
**.PC1.udpApp[0].destAddresses = "PC2"
**.udpApp[1].typename = "UDPSink"
**.udpApp[1].localPort = 1234
```

Figure 12: Shows the one-way configuriation of packet tracking with the help of OMNET++

7.3.4 Packet capture(PCAP)

PCAP-files are a common file format and we use it to record, analyze and review network traffic. With the help of INET, it is possible in the OMNET++ simulator to record the network traffic with every simulation via PCAP-files and to view it like Wireshark using a network number [39, p. 88]. PCAP-files can easily be created in the simulator OMNET++ using the "PcapRecorder" module. The recordings capture the traffic at the network level and the traffic

that occurs in a network. It is possible with this simulator to view different protocols and to filter the recordings to certain protocols through filters to reduce the file size [39, p. 88].

```
*.s3.pcapRecorder[*].pcapFile ="results/s3.pcap"
*.s3.pcapRecorder[*].moduleNamePatterns = "eth[*]"
*.s3.pcapRecorder[*].verbose = true
```

Figure 13: Configuration of a PCAP file created and recorded

Figure 13 shows how to record a PCAP-file from a specific network device in the simulator OMNET++. It also shows how router s3 network traffic is recorded. All traffic running over the s3 is recorded and saved in a PCAP-file. A file is created where the network traffic is saved, which can be set in the "PcapRecorder" module under the property "pcapFile". You can set other properties, the verbose setting was chosen so that packets should also be logged. The other interesting feature is "moduleNamePatterns" is used, so you can specify which interfaces you want to listen to. In addition, more properties would be possible, for example, a filter could be set, which can be found under "packetDataFilter". Here you can filter for certain packets under all traffic.

Wireshark
Wireshark is a packet sniffing software used to analyze network traffic. The advantage of this software is that it is freely accessible and available for every operating system. If Wireshark is installed, then it is necessary to know which interface you want to listen to receive the data traffic that goes through this interface (see [28, p. 40]). All data traffic can now be viewed and analyzed by every user. It should be noted that network devices exchange several types of packets with each other to communicate [28, p. 40]. In summary, Wireshark offers an excellent opportunity to record and then analyze the network traffic. Another feature is that you can look into the respectively sent packets [28, p. 40]. The user interface can also track related packets and you can also search by searching for a specific IP address or packet. For example, in a three-way handshake, each TCP-packet can be tracked and analyzed individually.

Since it was already discussed in the upper area at the PCAP section that OMNET++ can create PCAP-files and thus with the help of Wireshark it can be read in that this software can also track and analyze network traffic. The different network traffic that occurs in my simulation and which packets are exchanged between the routers with an EIGRP configuration can thus be tracked.

7.4 Results and observations

In this part of the work the results of the work are described and assessed based on their results and the reasons they came about are described. These results

are described with the help of tables and graphics for a better overview. Three experiments were carried out that deal with different scenarios and describe the behavior of the routing protocols. EIGRP, extended EIGRP, OSPF and RIP are run in the simulator and the results are compared with one another.

Overview of the experiment topology
The network topology shown in Figure 14 is the starting point of every experiment. In these experiments different scenarios are carried out, which are supposed to solve the previously defined metrics. These experiments are also used to collect the data so that the routing protocols can be compared with one another.

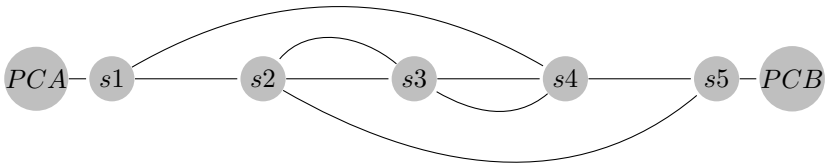


Figure 14: This graph forms our network topology, which we use for the individual experiments. In the respective experiments, packets are sent from PCA to PCB.

7.4.1 Comparison to other algorithm with experiments

In this area, the results are recorded in tables that record various data on metrics. Now the experiment is configured shown in Figure 10 with the routing protocols EIGRP, OSPF and RIP. We chose this because the routing protocols mentioned use a different algorithm to calculate the paths to the destinations. Then these are compared and analyzed using the metrics which are mentioned before. This metrics should show whether the packets arrive at the destination and which path they take. How long the updates need for processing with the respective routing protocols. Both comparisons and differences of the simulators to the respective results are discussed. These comparisons were divided into three different experiments and will be compared with one another below.

Experiment 1
In this experiment, the respective routing protocols were executed and the paths were taken as they would normally be calculated. The simulation is carried out for 150 seconds and an update is carried out after 50 seconds. After the simulation has ended, it is checked how many packets sent by ping and UDP packets go through the waypoint. The aim of this experiment is to show which path the routing protocols take through the network and how many packets of it go through the waypoint.

OMNET++
Table 6 and Figure 15 were made with the simulator OMNET++. This simulator offers a wide variety of routing protocols and was run in sequence. The routing protocols are checked at different intervals between the ICMP packets. The sent packets are compared to the received packets and the packet forwarding is also viewed. What is of particular interest here is how many of the packets sent by waypoint s3 from the created topology that can be seen in Figure 10. In Table 6 you can see the comparison of routing protocols where the ICMP

Routing protocol	Interval between packets in milliseconds(ms)	Send packets & Lost packets	Received packets
EIGRP	10	14999, 59 lost	14940
EIGRP	100	1516, 7 lost	1509
EIGRP	500	299, 1 lost	298
EIGRP	1000	149, 2 lost	147
Modified EIGRP	10	14999, 1515 lost	13484
Modified EIGRP	100	1499, 145 lost	1354
Modified EIGRP	500	299, 26 lost	273
Modified EIGRP	1000	149, 12 lost	137
OSPF	10	15273, 3517 lost	11756
OSPF	100	1499, 344 lost	1155
OSPF	500	301, 62 lost	239
OSPF	1000	152 , 35 lost	117
RIP	10	15000, 2290 lost	12710
RIP	100	1535, 214 lost	1328
RIP	500	299, 39 lost	260
RIP	1000	149, 19 lost	130

Table 6: The routing protocols in comparison when simulating sent and received packets in 150 seconds in the simulator OMNET++

packets are executed at different intervals. It could be seen here that EIGRP performs best when running the simulation with an update to 50 seconds when it comes to the relationship between sent and received packets. With EIGRP and extended EIGRP, more packets arrive and also more are sent to PCB. Compared to RIP and OSPF, fewer packets are lost with the extended EIGRP.

At Figure 15 you can see how the values with the lowest error rate were taken from Table 6. So that the values are in a certain context, we then decided on the values at 500ms. This table should show how much the routing protocols take into account a waypoint in the network. As you can see, no routing protocol takes a path through the waypoint, but calculates the shortest path to the destination. The waypoint is only taken into account when we extend a routing protocol in this way. As with the extended EIGRP, you can see that the packets go through the waypoint (green bar). However, if we make comparisons to the

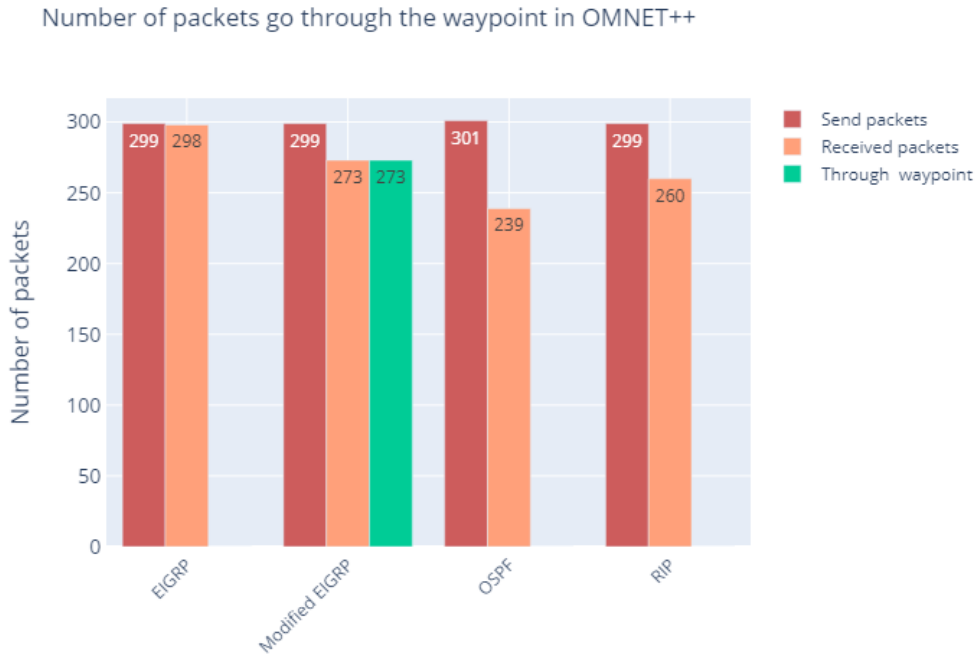


Figure 15: Routing protocols in comparison and how many packets go through the waypoint. The values with the lowest error probability are used. The results from the simulations of OMNET++ were taken.

original EIGRP, then 288 fewer packets are transmitted because the paths are specially calculated because they should go through the waypoint.

GNS3

In Figure 16 you can see how the experiments in GNS3 are treated and what the results of the simulation are. ICMP packets are sent to the destination and how many are sent and received are counted. It is also interesting here which path is considered the best by the respective routing algorithms. Another important detail is that it is also observed how many packets the waypoint s3 defined in the experiment are taken.

Evaluation of experiment 1

The Figure 16 shows the results that come from the GNS3 simulator. 7000 packets were sent to the destination via ping and it was checked how many of the packets were received even after an update was performed. It is interesting to see how the routing protocols calculate the new path. Since GNS3 cannot set the intervals of the ICMP packets, we ran several runs and as with OMNET++, to use the values that have the lowest error rate. Using the table,

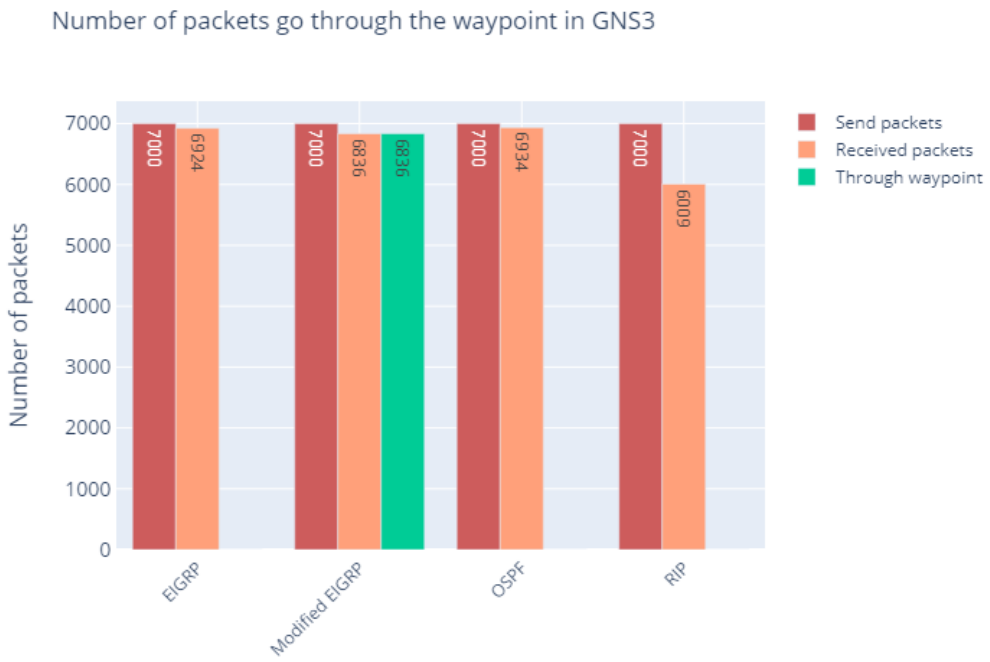


Figure 16: Routing protocols in comparison and how many packets go through the waypoint. The values with the lowest error probability are used. The results from the simulations of GNS3 were taken.

.

you can now see that, as with OMNET++, OSPF, EIGRP and the extended EIGRP performed well in terms of the error rate and how many packets were received. However, here the extended EIGRP has more packet losses than the original EIGRP and OSPF. You could also see here that only the extended EIGRP calculates the paths through the waypoint and forwards the packets through it. This explains why more packets are lost because the paths have to be calculated accordingly. From the last column you can see that only the extended EIGRP uses the waypoint and ensures that the packets are checked by it.

In summary of the first experiment, it can be said that the extended EIGRP and EIGRP are best suited for our created experiment. OSPF does not perform badly either, but has little more packet loss. However, the extended EIGRP is the only one that takes the waypoint into account, this was confirmed with both simulators. The other routing protocols (EIGRP, OSPF and RIP) calculate the shortest path to the destination without including the waypoint.

Experiment 2 - update time of the routing protocols
Since changes in the network can also occur while the packets are being forwar-

ded, an update of the routing algorithm is required. Routing anomalies should not occur during such updates. Each routing protocol treats changes in the network (metric changes, link failure) differently. We find it interesting to consider how long each routing protocol needs to process these updates. We therefore include these in the simulation and compare the routing protocols with one another. The aim of this experiment is to show how long the routing protocols need to process an update that happens in the network. The results of the second experiment can be seen in Figure 17 and in detail Figure 18.

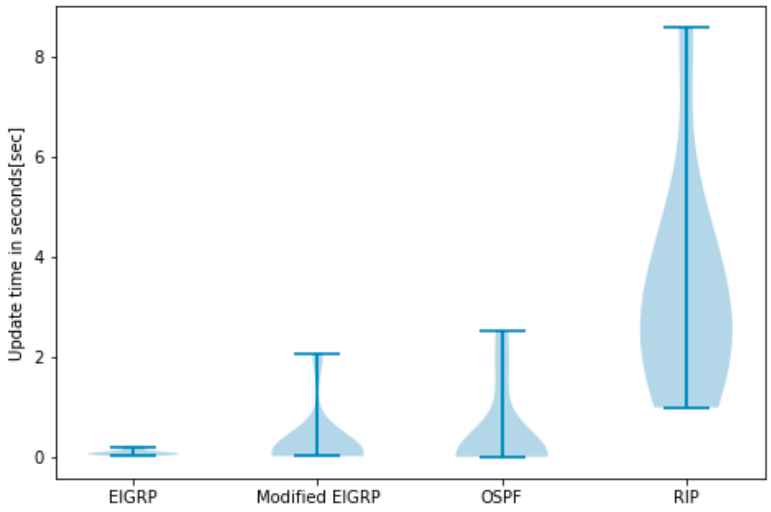


Figure 17: Routing protocols in comparison and how long it takes for the update.

.

Evaluation of experiment 2

The results were collected based on ten test runs and analyzed accordingly and visualized using a violin plot to see how the individual routing protocols are distributed. For this purpose there are two figures created, which only decided that the update times for RIP are still shown in Figure 17. You can see here that RIP delivers different values and also higher values such as 8.59 seconds, whereby this is an outlier and therefore the mean values are also entered the graph. It can be seen here that with ten attempts, RIP takes a lot longer than the other routing protocols. We wanted to zoom in on the result and Figure 18 shows how EIGRP, modified EIGRP and OSPF are compared with each other, as these are very good in terms of update time. EIGRP delivered the best results here, where the slowest time was 0.20 seconds. Otherwise it was quick. The modified EIGRP was worse, which sometimes took a little longer to manage the update. The modified protocol performed somewhat better overall than OSPF. You can see this in Figure 18. For this purpose the mean value is also shown in the graphic so that the results show what the distribution of the

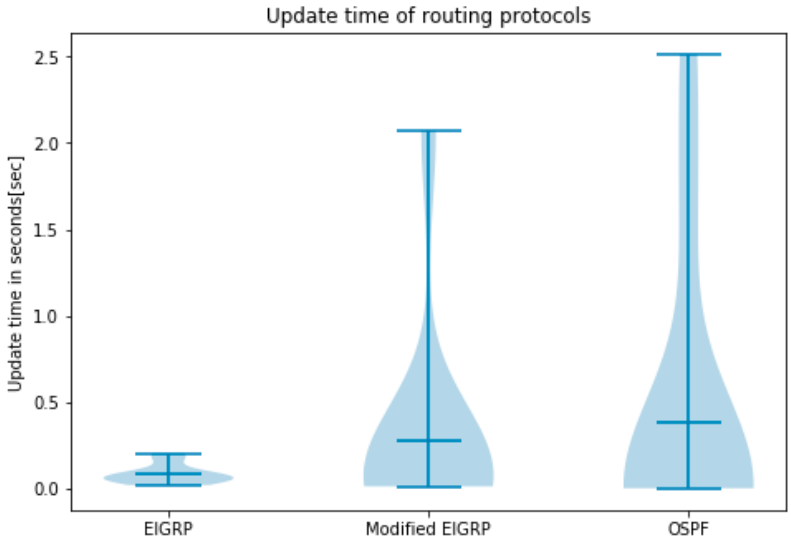
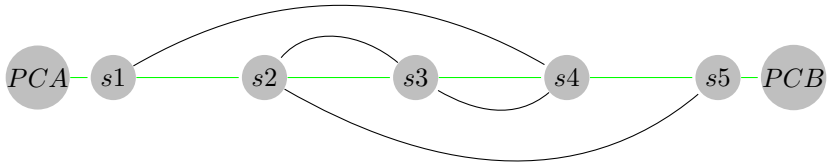


Figure 18: Routing protocols in comparison and how long it takes for the update. Here we inspected the routing protocols EIGRP, Modified EIGRP and OSPF, because you can follow the values better here than in Figure 17.

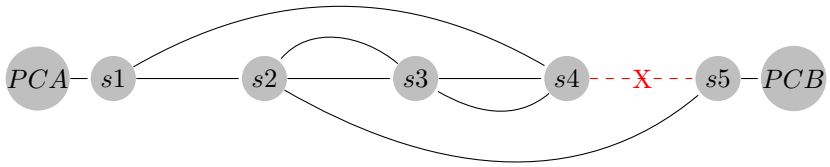
routing protocols looks like. This shows that the values in the modified EIGRP and OSPF have varied somewhat and sometimes took longer to update.
In summary, EIGRP works best in connection with the update times. Both OSPF and the modified EIGRP are somewhat worse, but still deliver good results. Thus, updates of these protocols can be processed quickly, as the graphics in the experiment have shown.

Experiment 3 - waypoint path analysis
In Figure 19 you can see the planned third experiment. This contains two different graphics that describe the topology and the experiment. In Figure 19a you can see the starting position of the experiment. The path is guided through the waypoint by taking the path PCA-s1-s2-s3-s4-s5-PCB which is show in green. In the simulation's course, a link failure occurs between s4 and s5, which is show in red. This behavior is shown in Figure 19b. Now the behavior of the different routing protocols is compared and analyzed. Here, it is interesting how many packets then go through the waypoint after the update has been processed.

Evaluation of experiment 3
In Figure 20 and Figure 21 you can see the results of the third experiment. These results are got when the link between s4-s5 fails and when a ping of 2500 packets checks how many packets arrive at the destination. Since the example is done with GNS3 and the ping was made by s1, the command "ping 192.168.0.18 repeat 2500" was executed. To get the results, the experiment is carried out



(a) The packets are forwarded between PCA-s1-s2-s3-s4-s5-PCB. Every routing protocol uses this path at the beginning and exchanges the packets.



(b) Link failure between s4 and s5 forces the routing protocol to update

Figure 19: The path through the waypoint is shown in 19a, which is displayed in green. After a while, a link failure between s4-s5 is shown in 19b.This serves as a representation for the third experiment of the master's thesis

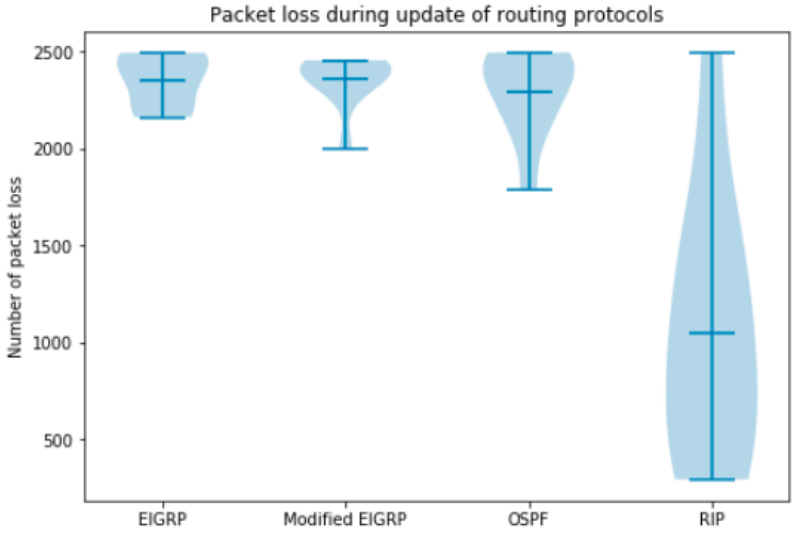


Figure 20: Routing protocols in comparison and how many packets they loss after a update is made. The results were created with the GNS3 simulator.

ten times. We created the results with the help of a violin plot for each routing

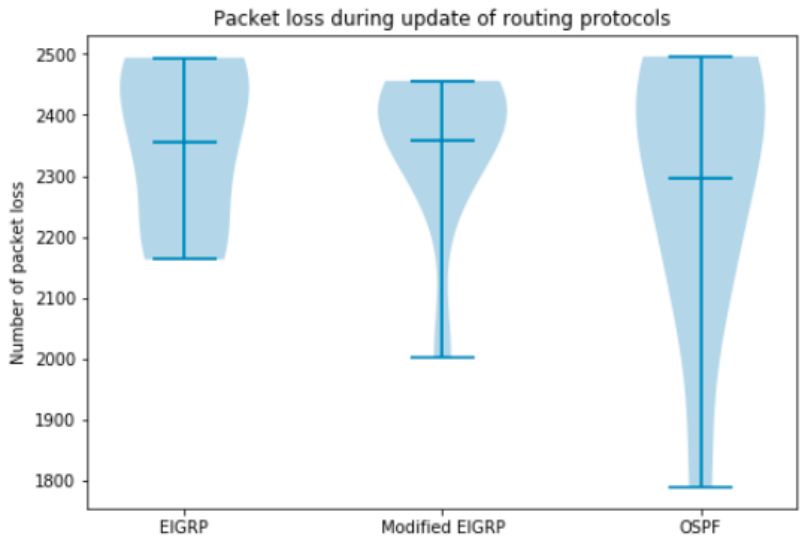


Figure 21: Routing protocols in comparison and how many packets they loss after a update is made. The results were created with the GNS3 simulator. The results of RIP have been omitted here so that the values can be better assessed.

protocol. You can see that RIP delivers different results and therefore the violin plot is drawn longer. The results of RIP scatter from almost all packets arriving up to a lot of packets loss. OSPF has delivered better results, where the violin plot is shorter than that of RIP. You can see from the Figure 20 that with OSPF more packets arrive after the link failure than with RIP. Because RIP loses a little more packets after the update is performed, we have concentrated in Figure 21 on the routing protocols that give the best results for this experiment. As you can see, EIGRP gives the best results, but the modified EIGRP and OSPF also have very good results for forwarding the packets to the destination after the update. However, it is also interesting to see here, even if every packet goes through the firewall before the update, then the packets only go through the firewall from the modified EIGRP after the update.

7.4.2 Differences between the simulator results

Some changes can be determined from the tables. With OMNET++, ICMP packets can be sent in different millisecond ranges with the help of the PIngApps and the UDPBasicAPP module. This enables a better analysis because the simulator also shows a lot of analyzes and metrics. In comparison, GNS3 does not offer as much metrics in advance, but the individual links between two nodes can be analyzed with Wireshark. This makes it possible to track the communication between the routers and analyze which packet types are exchanged by

the nodes.

7.4.3 Test duration

An important part of the thesis comprises the simulation and the analysis of the results. To achieve this, the simulation has to run for a certain period of time in order to collect data and learn how packets are forwarded. The simulation will log every packet on every router and examine the metrics discussed above. Special attention is paid to s3, which should represent the waypoint, i.e. the firewall in the network and which checks the individual packets. During this time, so many ICMP packets are sent from PC1 to PC2 and it is checked which paths the packets are taking and analyzed accordingly. Since I use the simulators GNS3 and OMNET++ for the simulation, the results are recorded in tables and compared.

With OMNET++ the interval between the pings or one-way UDP packets can be set accordingly. We use the intervals 10ms, 100ms, 500ms and one second for the simulation and see how the packets behave and how many arrive at their destination. With OMNET++ the simulation will run for 150 seconds so we can get the data accordingly and see how it is handled in the network. The PingAPP and UdpBasicApp provided by OMNET are used here. At 50 seconds, a scenario will take place that forces a change in the routes and an update of the router. It is also possible to record how the routes change and how long the update of the routing protocols takes until the network is convergent again.

At GNS3, packets are forwarded from s1 to s5 and a check is made to see what the calculated paths are. The "ping destination-address repeat 7000" command is used, which generates ICMP packets and forwards them through the network with the help of the routing protocols. Then tracerroute can check whether the network has changed after an update. Here a change in the network (metric change) is carried out during the ping, which forces the routing protocols to update.

7.4.4 Description of the results

During this work, the results were got with the help of the simulators OMNET++ and GNS3. The data are collected and analyzed with the help of three experiments. Different metrics are considered, which aim at how successful the routing algorithms are and how the updates are processed. In addition, a waypoint is included in the network where it is also checked to what extent this is included in the path calculation. After running the results, it can be said that the routing protocols EIGRP, modified EIGRP and OSPF gave the best results for the network topology created. With these protocols, most of the packets were delivered to the destination and update processing also takes not too long. Exciting here was that the metric changes were processed faster than a link failure. If the link failed, the routing protocols took longer to calculate a

new path. The results of RIP have shown that an update takes longer and that many packets are lost compared to the other protocols. That is why the results of the other routing protocols were more discussed. The results of the updates have shown that a metric change can be processed quickly by the protocols and only individual packets are lost. In contrast to this, with a link failure, all protocols take a little longer to calculate a path again. This is characterized by a higher error rate because packets using the old path are discarded. I was really surprised that EIGRP could keep up with OSPF, be it update time and lost packets. It did the best in terms of the network topology created when performing the individual experiments.

7.5 Discussion and evaluation

In this area the most important recognition gains that have arisen from this thesis are described. The applicability in practice and how the metrics behaved in the simulation's course are discussed.

7.5.1 Relation to the metrics

In this subsection I refer again to the metrics that were analyzed during the experiments. The results that were presented in the tables have already been discussed, but here it will summarize again and reflect on why they came about in this way.

Packets through the network
The result tables of both GNS3 and OMNET++ show that the routing protocols EIGRP, modified EIGRP and OSPF do lose packets, but still have a low error rate in order to forward the packets from source to destination through the network. Because updates were made during execution, some packets cannot be forwarded to the destination until the new path is calculated. It must also be taken into account that initially ICMP packets where the network was not yet converged. This made it possible to determine which protocol sends how many packets to the destination and how many are lost. It is interesting that EIGRP performed best in our experiment, even better than the modified EIGRP. At least the extended protocol lost more packets in the same runtime.

Packets through waypoint
As you can see in the results of the tables, no routing protocol would go through the waypoint. Only by modifying EIGRP is it now possible for the waypoint to be taken into account for the route calculation and for the packets traveling through the network to cross this node and thus the packets to be checked. One reason a routing protocol does not include a waypoint is that the node in our example is not on the shortest path. Since the routing protocol, be it link-state or distance vector, takes the shortest path to the destination, in our example these go past the firewall. The extension of EIGRP ensures that the packets are checked, but the path calculation takes a little longer because a specific

path through the waypoint has to be found. This can explain the higher packet loss, since care is taken that the destination is reached and that the path goes through the waypoint.

Reaction after update
Since an experiment aimed to analyze the update behavior of the routing protocols and how long they need to carry out the update. It was particularly interesting that the routing protocols EIGRP, modified EIGRP and OSPF have the best update times. This data was collected with the help of Wireshark and analyzed accordingly. To collect multiple data, the experiment was carried out ten times. With EIGRP and OSPF, this behavior could be followed using the packet types, which made it more difficult for RIP. So the two interfaces to the destination on s5 and measured how long the update took was monitored. It was exciting to see how long it took for updates to take place because the network converges pretty quickly. The most exciting thing is the behavior of the routing protocols in the event of an update. The third experiment aims to have the starting position of each network pass each packet through s3. Now a link failure occurs and only the extended EIGRP calculates the path so it continues to use s3 after the update.

7.5.2 Knowledge gained

The results did not surprise me as I expected that EIGRP and OSPF could give similar results. As we could see in the first experiment, the two protocols were very reliable and lost few packets. It was also exciting to see how the respective routing protocols react in the event of an update. Compared to the existing protocols, the extended EIGRP achieves similar results, but slightly more packets are lost than the EIGRP. The metric was confirmed that the extension calculates the paths so they go through the firewall. In Figure 15 and Figure 16 you can see the results of the simulators, which shows that otherwise no routing protocol takes the waypoint into account. The following findings from the master thesis were recognized:

- Packages only use one path: The packages are only sent on a defined path. There is no mixing of two paths where the packets are forwarded to another path in between.
- Packets go through the waypoint: The extension now calculates the paths so that the packets are forwarded through the firewall. This ensures that the packets that go through the network are checked.
- Not pure distance vector protocol: EIGRP used a distance vector protocol, but this protocol also uses properties of link-state protocols. This protocol is therefore classified as a hybrid protocol. It uses both types of routing protocol to exchange routing information between neighbors.
- Comparison of the routing protocols: Since the routing protocols were tested and compared with the help of experiments, they could be assessed

using various metrics. The metrics were defined in such a way that they concentrate on sending/receiving the packets on the destination and updating the network.

- Update experiment with OMNET++: An attempt was made to simulate an update with the help of a defined scenario in OMNET++, but the network did not react immediately and therefore I had to carry out experiments two and three in GNS3.

7.5.3 Differences to other routing algorithm

Routing protocols, be it distance vector or link-state vector protocols, have the goal of calculating a path that is the most favorable for the respective algorithm used. No primary goal of such algorithms is to see if there is a waypoint in the network responsible for checking the packets. As a result, it can happen that defective or faulty packets are sent to the destination, as these are not checked by the firewall, but that the shortest path does not go through the waypoint with the installed firewall. That is why we found it interesting to find out whether we can guarantee this property without losing the properties of EIGRP.

7.5.4 Applicability

Because the expansion of the DUAL was kept simple, there is no problem to use that, as an existing routing protocol with EIGRP was used. Because it was built on an existing algorithm, it can be used on any router where EIGRP is implemented. It was already explained in the upper part which different changes have to be made in order for the extension to be applied.

Based on the scenario that was examined for this thesis, the extension can be used very well and worked without problems. By prioritizing the paths to the waypoint, it is possible to expand the scenario and apply it to larger networks. The scenario we have developed comprises five points and shows how tricky the integration of a waypoint can be, since the routing protocols are primarily designed for the property that the packets take the shortest path. Now it is possible that not only one destination is used and that the topology is extended by several nodes. You can also define several waypoints in the network. We have set ourselves the goal of concentrating only on one waypoint in the network in this work. We used the GNS3 and OMNET++ simulators as they gave us the ability to configure different routing protocols. Because EIGRP is not offered in every simulator, we used the simulators mentioned here.

7.6 Summary

In this chapter the results of the simulations and the evaluation of these results have been analyzed. At the beginning we described the simulators used and what technologies we used to get the results. We compared different routing protocols, whereby one protocol extends EIGRP that works with metric

changes. Three experiments were carried out for the results, which contain different scenarios so that the routing protocols are compared with one another. It also describes how the experiments and simulation looks like and how long it runs. We also ran the simulation several times to get different results. We wanted to check how the routing protocols behave when we insert a defined waypoint. The results have shown that if the routing protocols were running, the packets would bypass the waypoint. That was because the routing algorithms try to find the shortest and best route to the destination and ignore whether there is a firewall in the network. In the next chapter the details and difficulties arose during the implementation and evaluation phase are described. It also explains what can be dealt with or what can be done better next time.

8 Discussion

This chapter describes again the important points and a conclusion from my point of view. This explains what difficulties that have arisen during the drafting process and what can be done better next time.

8.1 Problems with the simulators

In the master thesis the simulators OMNET++ and GNS3 are used to get the results and to analyze them. Both simulators were used because both had an implementation for EIGRP and with OMNET++ you could still inspect the program code and expand it. When using it, there were quite a few problems while editing. With GNS3 I had the problems that this simulator uses the hardware of my laptop and gives a maximum load when using many routers. This can affect the results somewhat and lead to somewhat worse results. To prevent this, we carried out the experiments at multiple points in time so we could compare the results and see the differences. Another problem that arose was that the Dynamips emulation software caused problems when starting the devices. This could be solved by restarting the program. In addition, there were difficulties in configuring the network topology, as only a limited number of serial interfaces could be configured. So we couldn't add any second links between the routers. But that had no side effect because the paths are still calculated accordingly. Thanks to the various manuals, OMNET++ has a good description so that this simulator can be installed successfully. It is important to use the installation manual for the operating system used. Since the ANSARouter module has included here, it is important to use the specified OMNET++-simulator with the version for the ansainet module. If this is not taken care of, some classes will not be loaded and errors will occur when compiling. However, using the two simulators is not difficult because there is a lot of documentation on them and they are interesting to use. Once you have created and configured your network topology, you can now test it using various metrics and experiments. Initially, the topology was created, configured and executed in OMNET++. There was a problem with OMNET++ when an update occurs that took the simulator a

long time to send an update package to the neighbor. Several seconds passed with no reaction to the changes and nothing happened. That's why it was the switch to GNS3 here, because you can easily follow the update times and their update packages here.

8.2 Problems with the implementation

Here GNS3 referring at first. With this simulator it is possible that the devices can be configured with Python. However, the routing protocols cannot be expanded because there is no insight into the implementation. The network topology can thus be created, configured and then checked. OMNET++ can be expanded with the help of the ansinet module, which has implemented various routing protocols. This allows you to see the implementation and expand it. One problem that happened is, that when changing the current implementation, errors occurred. During compiling the files, there were errors that I couldn't assess. The mistake was that the makefile had to be rebuilt so that the individual classes had to be reloaded. Another difficulty was understanding the respective classes and their context. Since there are many classes, the meanings must be looked at, that was made with the help of debugging the program and see the configuration steps. There is also the difficulty of which idea the implementation of the master thesis will achieve. It was then prioritized the links that lead to the waypoint. This is done with the help of metric changes.

8.3 Relations to the goals

In this subsection, a description of the answers to the questions which are mentioned in the introduction is given. These answers are based on the results that were made while working on the thesis.

8.3.1 Extended algorithm compared to existing ones

Based on the results got through the simulations, it can be determined that the modified EIGRP is worse than the original EIGRP. This was analyzed using metrics such as packet loss and update times. The modified EIGRP is a bit worse because it deals with the property of waypoint enforcement and the path calculation for this can take a little longer than with the original EIGRP, which is only designed to calculate the shortest route to the destination. Paths are thus checked through a previously defined node and if this is not on the shortest path, the packets pass. This can also be seen from the table which explains how many packets go through this waypoint. Using the network topology created in Figure 10, no protocol calculates the path through s3 except for the modified EIGRP. By choosing the method of changing metrics, we could extend EIGRP so it can be executed successfully in both simulators. In the simulators you could see that good results are achieved and can keep up with OSPF and EIGRP. Because the DUAL is executed when the network topology changes (metric change, link failure), it has the same properties for a change as the original EIGRP. The

properties of the DUAL and their various states in active mode handle the updates that occur well. Thus, the modified protocol reacts just as efficiently to the updates, but the choice of the path is designed so that the waypoint is included. Because our planned topology is a smaller one, one can conclude that the ISP would have more effort to configure in larger networks, since it must ensure that the links to the waypoint are prioritized and the metrics are kept correspondingly small.

8.3.2 Algorithm based on distance vector mechanism

In this thesis an extension of the algorithm based on EIGRP was made. So we didn't just concentrate on distance vector protocols. So that routing anomalies can be prevented, we ensure that the packets are only forwarded on one path and that a node is installed as a firewall. We have expanded the implementation of EIGRP by helping each other with the help of the metric change and the prioritization of the links to the waypoint. The ISP of the network is asked to set the metrics for the waypoint so low that the path goes through this node. This means an increased effort for the ISP, but it can be ensured that every packet that goes through the network is checked. With the help of the DUAL used by EIGRP, the freedom from loops is guaranteed in advance. With this work we wanted to dedicate a network policy property like waypoint enforcement. This defines a node in the topology that should be in the path so that the packets are checked before they are forwarded to the destination. However, in this work we have defined that there is only one waypoint in the network. This would be a weak point when implemented in larger networks, because this node would have to examine every packet. This could cause delays or flooding of the node. Since EIGRP was used, a hybrid protocol is used that also uses different tables to exchange routing information. So the focus is not only on distance vector properties, but a mixture of both protocol types. The information is stored as a link-state protocol would do. However, the messages are only exchanged between the surrounding neighbors.

8.3.3 Guarantee for other network topologies

In this thesis, the results are checked using a network topology. You can find this topology under the Figure 14. The results have shown that by configuring the network, the modified EIGRP calculates the paths so that the packets received from the destination go through the waypoint. It was a good example to show that the routing protocols take the shortest path on the way to the destination. Because the waypoint is not on the shortest path, help is required to ensure the property of waypoint enforcement. Now it is interesting to know whether this technique can be used for any network topology. To change metrics, each network topology could be configured in such a way that they use the waypoint when calculating the route. Depending on the size of the network, this means a greater effort for the person responsible for the network. The network can have been constructed in such a way that the path to the firewall can be very

long, where it's difficult to guarantee the property of waypoint enforcement. In addition, there can be a link failure on the way to the firewall, a new route would have to be calculated on this basis and depending on the construction of the network, this can reach the destination without checking.

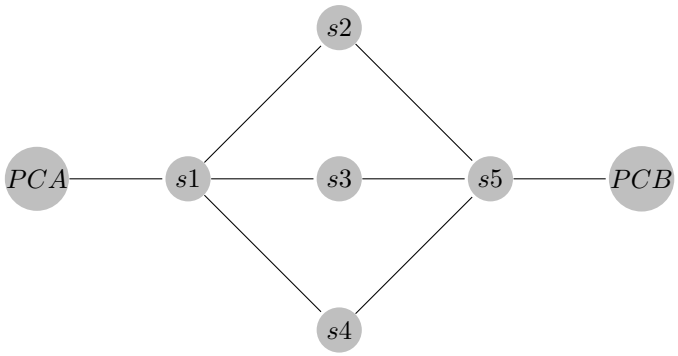


Figure 22: An example where the researched idea can have problems with the packets going through the network going through the waypoint.

Figure 22 shows a considered example where the mentioned idea might not work. Here, s3 is again defined as a waypoint. It could initially be configured so that the path goes from PCA-s1-s3-s5-PCB and thus uses the waypoint. The case could arise that the link between s1-s3 fails and a new path that goes past s3 has to be calculated. This is only one example of a network topology where problems could arise. Further topologies can be created where problems can arise with our idea to guarantee waypoint enforcement.

9 Conclusion

In this part, the most important details of the master thesis are summarized. An outlook of what can still be taken into account when working on the topic are described. For this purpose, the most important results are summarized again and how they solve the problem.

9.1 Summary

The purpose of this master's thesis was to provide researchers which are interested in the field of routing protocols and network updates. At the begin the routing protocol methods and how they work were described. Afterwards an overview of the current status and already examined work was received. At the end, the routing protocols are compared with one another through three experiments. The aim of this work was to extend a routing protocol and its algorithm that is also based on distance vector properties. We show that an existing routing protocol with EIGRP is extended in such a way that it supports the property of waypoint enforcement. This extension concerns the calculation

of the paths with the help of the metric. Since EIGRP was used, the path calculation could be influenced by setting the metric. The waypoint is known before the paths are calculated. This allows the routes to this node to be prioritized by keeping the metric small and taking the route through the waypoint. This means that the problem that packets are forwarded to the destination without being checked can be addressed. We offer a way to solve this problem and to include the waypoint in the route calculation. The experiments clarified that, with our network topology, the routing protocols EIGRP, modified EIGRP and OSPF show similar results for ensuring that the packets arrive at their destination through the network even if the network changes. The updates to these logs were also similar, as it often only takes milliseconds to manage the updates. An interesting point is that the experiments show that only the extended routing protocol uses the waypoint based on the created topology. To measure and analyze the results, two simulators were used here. These are OMNET++ and GNS3, which have implemented different routing protocols and were the best solution for the defined problem. With this work we have shown that the routing paths can be calculated so that the packets that go through the network from A to B are checked. This topic opens a wide landscape of further challenges for IGP to deal with policy properties. Further metrics and questions can be examined and it can be seen whether it can be solved more efficiently.

9.2 Outlook

Since we have inserted a waypoint in an example during our work, it is now possible to check this on other network topologies. In addition, the insertion of several waypoints would also be very interesting and how the routing protocols can handle this. It is an exciting topic because it has not yet been researched that routing protocols network policies such as including a firewall in the routing path.

Nowadays, large companies will look less at the IGP configuration and will increasingly rely on SDN. Since it can calculate the paths with the help of the controller and the controller takes over the path regulation and determines how the path through the network looks and can thus work through network policies properties such as waypoint enforcement. Other companies that do not yet use SDN can put even more effort into it and build on this idea of waypoint enforcement of the routing protocols. Thus there would be some security, since the packets from the waypoint have to be checked before they go to the destination.

By adding the feature of checking the packages, a security feature can be built in, as this ensures that the packages are not defect or fault. Compromises have to be made for this, since a waypoint is built into the network which should also appear in the routing path it may be possible that the shortest path is no longer used.

10 List of abbreviations

Glossary

AD Administrative Distance. 11, 12, 75

ANSA Automated Network Simulation and Analysis. 42–44, 52, 68, 74

AS Autonomous system. 15

DASM Diffusing Algorithm for Shortest Multipath. 22

DUAL Diffusing Update Algorithm. 5, 7–9, 14, 21–23, 31, 32, 35–41, 44–46, 51, 67, 70, 74, 80

EIGRP Enhanced Interior Gateway Routing Protocol. 5, 7–9, 11, 12, 14, 21, 24, 31–37, 41–43, 46, 48, 50–52, 55–61, 63–72, 74, 75, 80

FLIP Fast Lightweight Policy-preserving. 3, 27

GNS3 Graphical Network Simulator-3. 3, 5, 8, 9, 42, 49–51, 58, 59, 61–65, 67–69, 72, 75, 81

ICMP Internet Control Message Protocol. 41, 53, 57, 58, 63–65

IGP Interior Gateway Protocols. 7, 16, 17, 23, 33, 35, 41, 52, 72

IP Internet Protocol. 36, 43, 49, 52, 55

IPv4 Internet Protocol Version 4. 14, 42–44

IPv6 Internet Protocol Version 6. 14, 42, 44

IS-IS Intermediate System to Intermediate System Protocol. 7, 13

LSA Link-State Advertisement. 12

MIP Mixed Linear Program. 26

MPLS Multipath-Label Switching. 28, 52

NED Network Description. 52

OMNET++ Objective Modular Network Testbed in C++. 3, 5, 8, 9, 41–45, 47, 49–55, 57–59, 63–65, 67–69, 72, 74–76, 80, 81

OSPF Open Shortest Path First. 7, 9, 11, 12, 24, 34, 42, 48, 50, 51, 56, 57, 59–61, 63–66, 69, 72, 75

PCAP Packet capture. 32, 34, 50, 54, 55

PDM Protocol-dependent module. 42

RAFC Revised active feasible condition. 23

RIP Routing Information Protocol. 9, 11–14, 20, 21, 24, 34, 42, 48, 50, 51, 56, 57, 59, 60, 63, 65, 66, 75

RPFC Revised passive feasible condition. 23

RTP Cisco Reliable Transport Protocol. 42

SDN Software Defined Network. 2, 9, 17, 25, 26, 46, 72

SITN Ships-in-the-Night. 17, 23, 24

TCAM Ternary Content-Addressable Memory. 27

TCP Transmission Control Protocol. 43, 55

TTL Time-To-Live. 15, 53

UDP User Datagram Protocol. 14, 41, 43, 54, 56, 64

List of Figures

List of Tables

References

[1] What is inet framework? `https://inet.omnetpp.org/Introduction.html`.

[2] Use of OSI IS-IS for routing in TCP/IP and dual environments. RFC 1195, Dec. 1990.

[3] Academy, C. N. *Routing and Switching Essentials V6 Companion Guide*. Cisco Press, 2016.

[4] Albrightson, B., Garcia-Luna-Aceves, J., and Boyle, J. Eigrp - a fast routing protocol based on distance vectors.

[5] Albrightson, B., Garcia-Luna-Aceves, J., and Boyle, J. Eigrp - a fast routing protocol based on distance vectors.

[6] Bashandy, A., Filsfils, C., Decraene, B., Litkowski, S., Francois, P., Voyer, D., Clad, F., and Camarillo, P. Topology Independent Fast Reroute using Segment Routing. Internet-Draft draft-bashandy-rtgwg-segment-routing-ti-lfa-05, Internet Engineering Task Force, Oct. 2018. Work in Progress.

[7] Bruno, A. *CCIE routing and switching exam certification guide*. Cisco Press, 2002.

[8] Chamberlain, T. *Learning OMNeT++*. Packt Publishing, 2013.

[9] Foerster, K., Parham, M., Chiesa, M., and Schmid, S. Ti-mfa: Keep calm and reroute segments fast. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (2018), pp. 415–420.

[10] Foerster, K., Schmid, S., and Vissicchio, S. Survey of consistent software-defined network updates. *IEEE Communications Surveys Tutorials 21*, 2 (Secondquarter 2019), 1435–1461.

[11] Foerster, K., Schmid, S., and Vissicchio, S. Survey of consistent software-defined network updates. *IEEE Communications Surveys and Tutorials 21*, 2 (2019), 1435–1461.

[12] Foerster, K.-T., Parham, M., Schmid, S., and Wen, T. Local fast segment rerouting on hypercubes.

[13] Francois, P., and Bonaventure, O. Avoiding transient loops during the convergence of link-state routing protocols. *IEEE/ACM Transactions on Networking 15*, 6 (2007), 1280–1292.

[14] Francois, P., Shand, M., and Bonaventure, O. Disruption free topology reconfiguration in ospf networks. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications* (May 2007), pp. 89–97.

[15] Förster, K., Mahajan, R., and Wattenhofer, R. Consistent updates in software defined networks: On dependencies, loop freedom, and blackholes. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops* (2016), pp. 1–9.

[16] Garcia-Luna-Aceves, J. A unified approach to loop-free routing using distance vectors or link states. vol. 19, pp. 212–223.

[17] Garcia-Lunes-Aceves, J. J. Loop-free routing using diffusing computations. *IEEE/ACM Transactions on Networking 1*, 1 (1993), 130–141.

[18] Hengartner, U., Moon, S., Mortier, R., and Diot, C. Detection and analysis of routing loops in packet traces. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment* (New York, NY, USA, 2002), IMW '02, Association for Computing Machinery, p. 107–112.

[19] Hoong, Y. C. *CCNA Complete Guide: The BEST EVER CCNA Self-Study Workbook Guide*, 2nd ed. CreateSpace, Scotts Valley, CA, 2010.

[20] Kalyan, G. P. S., and Prasad, D. V. V. Optimal selection of dynamic routing protocol with real time case studies. In *2012 International Conference on Recent Advances in Computing and Software Systems* (2012), pp. 219–223.

[21] Lakshminarayanan, K., Caesar, M., Rangan, M., Anderson, T., Shenker, S., and Stoica, I. Achieving convergence-free routing using failure-carrying packets. *SIGCOMM Comput. Commun. Rev. 37*, 4 (Aug. 2007), 241–252.

[22] Ludwig, A., Dudycz, S., Rost, M., and Schmid, S. Transiently secure network updates. *SIGMETRICS Perform. Eval. Rev. 44*, 1 (June 2016), 273–284.

[23] Luo, L., Yu, H., Luo, S., and Zhang, M. Fast lossless traffic migration for sdn updates. In *2015 IEEE International Conference on Communications (ICC)* (2015), pp. 5803–5808.

[24] Luo, S., Yu, H., and Li, L. Consistency is not easy: How to use two-phase update for wildcard rules? *IEEE Communications Letters 19*, 3 (2015), 347–350.

[25] Lynch, D. C., and Jacobsen, O. J. A Glossary of Networking Terms. RFC 1208, Mar. 1991.

[26] Malkin, G. S. Traceroute Using an IP Option. RFC 1393, Jan. 1993.

[27] McGeer, R. A safe, efficient update protocol for openflow networks. In *Proceedings of the First Workshop on Hot Topics in Software Defined Networks* (New York, NY, USA, 2012), HotSDN '12, Association for Computing Machinery, p. 61–66.

[28] Mishra, C. *Mastering Wireshark*. Packt Publishing, 2016.

[29] Networks, B. The world's fastest and most programmable networks. `https://barefootnetworks.com/resources/worlds-fastest-most-programmable-networks/`, 2016. White Paper.

[30] Neumann, J. C. *The Book of GNS3: Build Virtual Network Labs Using Cisco, Juniper, and More*, 1 ed. No Starch Press, 2015.

[31] Reitblatt, M., Foster, N., Rexford, J., Schlesinger, C., and Walker, D. Abstractions for network update. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication* (New York, NY, USA, 2012), SIGCOMM '12, Association for Computing Machinery, p. 323–334.

[32] Savage, D., Ng, J., Moore, S., Slice, D., Paluch, P., and White, R. Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP). RFC 7868, May 2016.

[33] Vanbever, L., Vissicchio, S., Pelsser, C., Francois, P., and Bonaventure, O. Seamless network-wide igp migrations. In *Proceedings of the ACM SIGCOMM 2011 Conference* (New York, NY, USA, 2011), SIGCOMM '11, Association for Computing Machinery, p. 314–325.

[34] Vanbever, L., Vissicchio, S., Pelsser, C., Francois, P., and Bonaventure, O. Lossless migrations of link-state igps. *IEEE/ACM Transactions on Networking 20*, 6 (Dec 2012), 1842–1855.

[35] Vanbever, L., Vissicchio, S., Pelsser, C., Francois, P., and Bonaventure, O. Lossless migrations of link-state igps. *IEEE/ACM Transactions on Networking 20*, 6 (Dec 2012), 1842–1855.

[36] Veselý, V. Ansainet extends inet framework for omnet++. `https://github.com/kvetak/ANSA/`.

[37] Veselý, V., Rek, V., and Rysavy, O. *Enhanced Interior Gateway Routing Protocol with IPv4 and IPv6 Support for OMNeT++*. 01 2015, pp. 65–82.

[38] Veselý, V., Rek, V., and Rysavy, O. *Enhanced Interior Gateway Routing Protocol with IPv4 and IPv6 Support for OMNeT++*. 01 2015, pp. 65–82.

[39] Virdis, Antonio; Kirsche, M. *Recent Advances in Network Simulation. The OMNeT++ Environment and its Ecosystem*. Springer International Publishing, 2019.

[40] Vissicchio, S., and Cittadini, L. Safe, efficient, and robust sdn updates by combining rule replacements and additions. *IEEE/ACM Transactions on Networking 25*, 5 (2017), 3102–3115.

[41] Vissicchio, S., Cittadini, L., Bonaventure, O., Xie, G. G., and Vanbever, L. On the co-existence of distributed and centralized routing control-planes. In *2015 IEEE Conference on Computer Communications (INFOCOM)* (2015), pp. 469–477.

[42] Vissicchio, S., Cittadini, L., Bonaventure, O., Xie, G. G., and Vanbever, L. On the co-existence of distributed and centralized routing control-planes. In *2015 IEEE Conference on Computer Communications (INFOCOM)* (April 2015), pp. 469–477.

[43] Wijaya, C. Performance analysis of dynamic routing protocol eigrp and ospf in ipv4 and ipv6 network. In *2011 First International Conference on Informatics and Computational Intelligence* (2011), pp. 355–360.

[44] Zaumen, W. T., and Garcia-Luna-Aceves, J. J. Loop-free multipath routing using generalized diffusing computations. In *Proceedings. IEEE INFOCOM '98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No.98* (1998), vol. 3, pp. 1408–1417 vol.3.

[45] Zhao, C., Liu, Y., and Liu, K. A more efficient diffusing update algorithm for loop-free routing. In *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing* (2009), pp. 1–4.

[46] Zhengyu Xu, Sa Dai, and Garcia-Luna-Aceves, J. J. A more efficient distance vector routing algorithm. In *MILCOM 97 Proceedings* (1997), vol. 2, pp. 993–997 vol.2.

A Zusammenfassung

Computernetzwerke sind zu einer kritischen Infrastruktur geworden. Heutige Netzwerke sollten sehr flexibel sein und schnelle Updates unterstützen, um unerwünschte Konfigurationen innerhalb des Netzwerks wie Routing-Schleifen zu vermeiden. Gegenwärtige Routing-Protokolle verfügen nur teilweise über ein Verfahren, um mit solchen unerwünschten Konfigurationen umzugehen. Bei diesen Methoden können jedoch manchmal immer noch Routingprobleme auftreten. Diese Masterarbeit befasst sich mit einem Routing-Protokoll namens Enhanced Interior Gateway Routing Protocol(EIGRP), das den Diffusing Update-Algorithmus(DUAL) verwendet, der dieses Fehlverhalten erkennt und über geeignete Mechanismen verfügt, um die Schleifenfreiheit zu gewährleisten. Durch die Erweiterung von EIGRP werden verschiedene Experimente in den Simulatoren OMNET++ und GNS3 erstellt und mit vorhandenen Routing-Algorithmen verglichen. Verschiedene Eigenschaften werden überprüft, um die Effizienz der Routing-Algorithmen zu bestimmen, sodass auch weitere Eigenschaften wie die Durchsetzung von Wegpunkten garantiert werden können. Ziel dieser Arbeit

ist es, die Eigenschaft der Wegpunktdurchsetzung durch die Erweiterung von EIGRP zu gewährleisten. Wir werden Simulationen verwenden, um zu zeigen, dass die Erweiterung es den Paketen ermöglicht, den Wegpunkt in die Routenberechnung einzubeziehen.

B Access to the results

You can view them in the Git repository so you can understand and reproduce the results that have arisen during your work. This offers the installation steps and the data for the following examples:

- GNS3 files
- OMNET++ files

You can access the results on your device with the following command:

- Repository: https://gitlab.cs.univie.ac.at/ct-projects/routing-anomalies
- HTTPS: git clone https://gitlab.cs.univie.ac.at/ct-projects/routing-anomalies.git
- SSH: git@gitlab.cs.univie.ac.at:ct-projects/routing-anomalies.git