Julius-Maximilians-Universität Würzburg
Institut für Informatik
Lehrstuhl für Kommunikationsnetze
Prof. Dr.-Ing. Phuoc Tran-Gia

Optimization and Design of Network Architectures for Future Internet Routing

Matthias Hartmann

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Bericht 2/15

Würzburger Beiträge zur Leistungsbewertung Verteilter Systeme

Herausgeber

Prof. Dr.-Ing. Phuoc Tran-Gia
Universität Würzburg
Institut für Informatik
Lehrstuhl für Kommunikationsnetze
Am Hubland
D-97074 Würzburg

Tel.: +49-931-31-86630
Fax.: +49-931-31-86632
Email: trangia@informatik.uni-wuerzburg.de

Satz

Reproduktionsfähige Vorlage vom Autor.
Gesetzt in LaTeX Computer Modern 9pt.

ISSN 1432-8801

Optimization and Design of Network Architectures for Future Internet Routing

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius-Maximilians-Universität Würzburg

vorgelegt von

Matthias Hartmann

aus

Würzburg

Würzburg 2015

Eingereicht am: 21.03.2014
bei der Fakultät für Mathematik und Informatik
1. Gutachter: Prof. Dr.-Ing. Phuoc Tran-Gia
2. Gutachter: Prof. Dr.-Ing. Andreas Mitschele-Thiel
Tag der mündlichen Prüfung: 29.04.2015

Meinen Eltern in größter Dankbarkeit gewidmet.

Danksagung

Ohne vielfältige Unterstützung wäre diese Arbeit nicht möglich gewesen.

Ich danke meinem Doktorvater Prof. Phuoc Tran-Gia, der es mir möglich gemacht hat, die vorliegende Arbeit zu verfassen. Seine fachliche Expertise und seine Vernetzung mit Industrie- und Forschungspartnern erlaubten mir, von Anfang an auf hohem Niveau zu forschen, und meine Theorien innerhalb internationaler Forschungsprojekte und auf Konferenzen und Workshops zu diskutieren und weiterzuentwickeln. Die sehr angenehme Arbeitsatmosphäre an seinem Lehrstuhl mit Einbindung in die Lehre und Verantwortungsübertragung für Projekte bildeten die Grundlage für eigene Ideen und Arbeiten.

Ganz besonders bedanke ich mich auch bei Prof. Michael Menth, mit dem ich schon fast seit Beginn meines Informatikstudiums zusammenarbeiten konnte. Er hat seine wertvollen Erfahrungen mit mir geteilt und mir durch seine intensive Betreuung das wissenschaftliche Arbeiten vermittelt. Die vorliegende Dissertation, basierend auf zahlreichen gemeinsamen Veröffentlichungen, ist in enger Zusammenarbeit und ständiger Diskussion mit ihm entstanden.

Bedanken möchte ich mich auch bei Prof. Andreas Mitschele-Thiel, der die vorliegende Arbeit begutachtete, und bei Prof. Alexander Wolff und Prof. Samuel Kounev, die als Prüfer bei meiner Disputation fungierten.

Meinen ehemaligen Kollegen danke ich für die tolle Gemeinschaft und die kooperative und inspirierende Zusammenarbeit. Ich habe mich immer sehr wohl gefühlt. Für fast jedes fachliche Problem wurde gemeinschaftlich eine Lösung gefunden und durch zahlreiche Unternehmungen am und außerhalb des Lehrstuhls wurden aus Kollegen gute Freunde.

Ein besonderer Dank geht auch an Gisela Förster und Alison Wichmann

für ihre vielfältige administrative Unterstützung, insbesondere bei Dienstreisen und bei der Projektorganisation. Ich konnte mich immer auf ihre herzliche Tatkräftigkeit verlassen.

Prof. Kurt Tutschku danke ich dafür, dass er mir einen Forschungsaufenthalt an der Universität Wien ermöglicht hat, und den Wiener Kollegen und Studenten für die herzliche Aufnahme und die gute Zusammenarbeit.

Besondere Freude hat mir während meiner Zeit am Lehrstuhl die Zusammenarbeit mit 'meinen' Studenten bereitet. Ich bedanke mich für tolle Ideen und großes Engagement in Abschlussarbeiten, Projekten und auch als Hilfskräfte, wodurch meine Arbeiten unterstützt und erleichtert wurden.

Zu guter Letzt danke ich ganz herzlich meinem Kollegen Dr. Thomas Zinner, der mich beim Zusammentragen der Einzelveröffentlichungen und Ideen motiviert, die Arbeit Korrektur gelesen und mich bei der Vorbereitung auf die Disputation besonders unterstützt hat.

Danke!

Contents

1 Introduction

Coming up with ideas is the easiest thing on earth.
Putting them down is the hardest.
(Rod Serling)

The Internet has evolved from pure scientific use to a central infrastructure for many businesses and recreational services. Today's society highly depends on this global network of networks. Even more so as many services have converged from dedicated networks to the Internet. Services like general telecommunication, live television and radio broadcasting, financial transactions, or telemedicine make high demands on the networks in terms of reliability and availability.

Most traffic in the current Internet is transported using the Internet protocol (IP). It is a packet switched transmission protocol, which provides best-effort packet delivery. Decentralized routing inside and between autonomous systems (ASes) steers packets towards their destinations. Most failures in this system can be recovered by slow but reliable distributed routing protocols. These mechanisms make the network very robust in terms of general restoration of connectivity, but no guarantees can be made for data transmission with specific quality of service (QoS) requirements. Unexpected events like network failures or flash crowds can lead to very low network performance. Constantly, new ideas are discussed to improve the Internet. Its modular architecture facilitated the replacement and addition of protocols, which enabled the Internet to support new requirements. This is one of the reasons for the success of the Internet. But every new protocol and mechanism must be carefully analyzed, and the network must

be adjusted accordingly to fully utilize the potential of each protocol. This monograph analyzes and improves routing both inside and between ASes. We optimize parameters and settings for plain IP routing and new add-on protocols, and propose a new naming and routing protocol to make the Internet more scalable and reliable.

Inside a typical AS that is using a link-state routing protocol like OSPF (Open shortest path first) or IS-IS (Intermediate system to intermediate system), the paths through the network are determined by decentralized routers, which calculate shortest paths according to administrative link costs. These link costs are the only way to influence the routing and, thereby, determine the load distribution inside the network. Link costs also influence less obvious properties like for example the ability to recover quickly from certain failures using additional protocols. Often, the implications of a specific link cost setting on the network are not obvious and might even have counter-intuitive effects. Optimizing the link costs in one area or for one protocol can deteriorate the performance of other mechanisms. Thus, the appropriate selection of an objective function that is used for network optimization is of utmost importance. In this monograph we investigate traffic optimization techniques for IP networks. Based on pure IP routing and also including newly developed fast reroute and admission control protocols, we show tradeoffs between different solutions and provide guidelines for network administrators, which help them to improve their network according to their goals.

To route data between ASes, every router in the Internet must be able to forward packets towards their destination addresses. Many access networks simply use a default route, which forwards all packets to unknown destinations towards their larger upstream provider. In the core of the Internet, the so called default-free zone (DFZ), routers must know a path to each destination. To make this task easier, groups of addresses are aggregated to larger blocks. Internet service providers usually assign their customers addresses from their own large blocks. This should create hierarchical and scalable structures for Internet routing. The number of users, devices, and services connected to the Internet is constantly

growing. In addition, a reliable connection is of utmost importance and, therefore, customers often use more than one provider for a resilient access to the Internet. When multiple connections are set up, the customer network needs provider-independent IP addresses to use these links. These are two of the main factors that currently lead to fast increasing routing table sizes in the DFZ, which is a major concern for the scalability of the Internet and a threat for its effective operation in the future. This monograph provides an overview over several different techniques that can be used to alleviate this growth. We propose a new naming and routing architecture that makes the Internet more scalable, and has many benefits, even for early adopters, while maintaining compatibility to the remaining Internet. A detailed description of the scientific contribution in this monograph is given in the following.

1.1 Scientific Contribution

This section summarizes the contributions of the author to the field of routing in the Internet. It gives an overview of the content of the studies presented in this monograph and explains their relations. Afterwards, it provides a summary of the author's contributions beyond this monograph. All of the studies are based on scientific publications of the author.

This monograph focuses on two different areas to improve routing in the Internet. In the first part, the optimization of IP routing and protocols using link cost optimization is examined. We compare different objective functions for the optimization and analyze optimization results for regular routing and for IP fast rerouting. In addition, we examine the pre-congestion notification (PCN) mechanism for admission control and flow termination, and optimize protocol parameters and link costs using appropriate objective functions. The second part of this work proposes a new routing and addressing protocol for the Internet, which can handle future growth while providing many additional benefits. We address challenges that arise during the development of such protocols and develop and propose solutions.

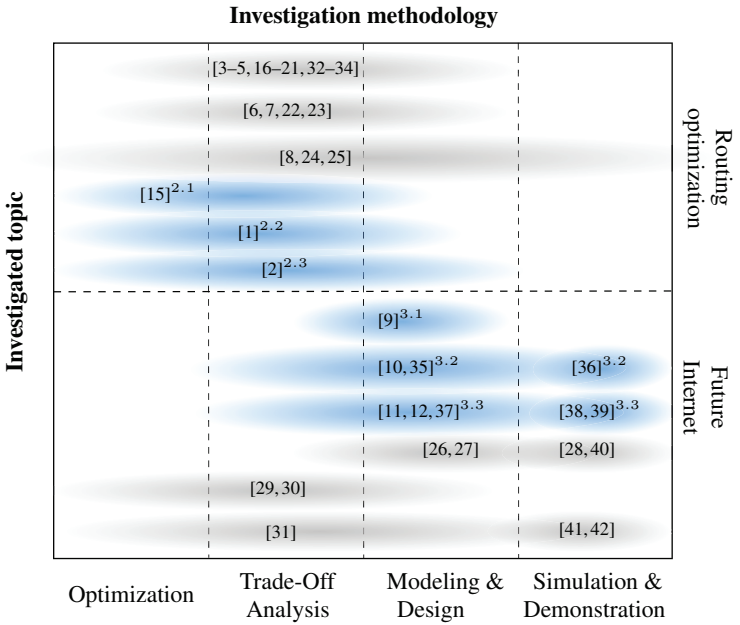


Figure 1.1: *Cartography of scientific contributions of the author on routing optimization and future Internet routing. The notation [x]^y in the blue areas indicates that the scientific publication or software demonstrator [x] is discussed in Chapter y of this monograph.*

Figure 1.1 presents a classification of the author's publications according to the investigated topic on the y-axis and the investigation methodology on the x-axis. The explored main topics are resilient routing optimization and future Internet protocol design, which are two essential approaches to improve the design and performance of networks. The methodologies comprise theoretical methods as modeling of networks, routing, and objective functions, heuristic optimization and trade-off analysis, protocol design and design guidelines, and practical parts as protocol simulations, proof-of-concept implementations, and demonstrations.

The first chapter in this monograph covers optimization of IP routing. Its first section is based on [15], which presents a study on the optimization of general intradomain routing in IP networks. Packets that are routed through the network generally follow least-cost paths according to administrative link costs. Routing optimization modifies these values to minimize an objective function for a network with given link capacities and traffic matrix. We investigate and compare various objective functions and evaluate the resulting routing.

The second part of the first chapter is based on [1]. Here we continue the study of intradomain IP routing optimization by extending our focus to a fast reroute (FRR) mechanism using loop-free alternates (LFAs). It is a simple mechanism that has been standardized recently and is already offered by several vendors. However, LFAs have two major drawbacks: They often cannot provide failure protection against all single link or node failures in spite of physical connectedness, and some LFAs cause routing loops in scenarios with node or multiple failures. We classify LFAs in three different categories according to their ability to avoid loops and propose several objective functions for IP routing optimization that can be used for different use-cases of LFA application. We show that the maximization of LFA coverage can lead to high link utilization and present a Pareto-optimization, which offers several routing options to the network administrator. With these options, the human operator can choose a tradeoff between protection and high utilization and select the setting he considers best for the network.

The third part of this chapter is based on [2,43] and presents a study on routing optimization and protocol analysis. It examines a new pre-congestion notification mechanism in IP networks, which uses packet metering and marking within a PCN domain to notify its egress nodes whether link-specific admissible or supportable rate thresholds have been exceeded by high priority traffic. Based on this information, simple admission control and flow termination is implemented. The latter is a new flow control function that is useful in case of high priority traffic overload, which can occur in spite of admission control, e.g., when traffic is rerouted in failure cases. Resilient admission control admits only as much traffic

as can be rerouted without causing congestion on backup paths in case of likely failures, e.g., single link failures. To achieve these functions, PCN requires additional bits in the packet header, which are used for marking packets. Two different architectures are proposed, one that uses only a single marking and another one that uses dual marking, but is more flexible. We propose algorithms to configure the link-specific PCN rate thresholds so that resources are utilized efficiently and fairly by competing traffic aggregates while meeting resilience constraints. This is done for the single and dual marking PCN architecture whereby the single marking case is more demanding since it requires that the supportable rate is a fixed multiple of the admissible rate on all links within a single PCN domain. Furthermore, we derive objective functions to optimize the underlying routing system for both cases. Our performance results for various network types show that the dual marking PCN architecture leads to significantly better resource efficiency than the single marking PCN architecture.

The second chapter of this monograph presents the design and implementation of a new naming and addressing protocol. The first part of the chapter is based on [9]. We explain reasons for the fast increase of the routing table size in the default-free zone, and show that it is a major concern for the scalability of the Internet and a threat for its effective operation in the future. Proposals exist to modify the current routing architecture in order to decelerate the growth of the routing tables in the DFZ, but they are difficult to deploy. The locator/identifier (Loc/ID) split principle is significantly different from routing and addressing in today's Internet, but it is expected to improve routing scalability. We explain its basic idea, address interworking issues, point out design options, and review current implementation proposals.

In the second part of this chapter, which is based on [10, 35, 36], we develop the GLI-Split framework, which fully implements the Loc/ID split architecture. It separates the functionality of current IP addresses into a stable identifier and two independent locators, one for routing in the Internet core and one for edge networks. This makes routing in the Internet more stable and provides more flexibility for edge networks. GLI-Split can be incrementally deployed and it is

backward-compatible with the IPv6 Internet. We describe its architecture, compare it to other approaches, present its benefits, and finally present a proof-of-concept implementation of GLI-Split.

To find the location of a host, it requires a mapping system that returns appropriate locators in response to map-requests for specific identifiers. In the third part of this chapter, which is based on [11,12,37–39], we propose FIRMS, a "Future Internet Routing Mapping System". It is fast, scalable, reliable, secure, and it is able to relay initial packets. We introduce its design, show how it deals with partial failures, explain its security concept, and evaluate its scalability.

Beyond this monograph, the author contributed to several other studies and projects in the field of future networks, routing, and optimization. We examined the efficiency of routing and resilience mechanisms in packet-switched communication networks [3,32], and looked at IP link costs and resilient multilayer networks [16–18,33]. Together with the Simula Research Laboratory in Norway, we examined advanced IP-FRR concepts. We optimized routing for notvia-addresses in combination with loop-free alternates [4, 34], and improved and optimized Simula's FRR mechanism, which uses multiple routing configurations [5, 19]. We solved issues with multiple equal-cost paths that lead to uncertainties in routing optimization [20] and proposed a routing optimization that covers the full failure cycle, i.e., the failure free routing, FRR state, reconverged state, and all loop-free convergence phases [21]. In cooperation with the Warsaw University of Technology, Poland, we compared our heuristically optimized shortest-path-based routing with explicit MPLS path layouts created by integer linear programs (ILPs) [6, 7, 22, 23]. In addition to the optimization tool, we also developed and published a resilience analysis tool to evaluate different resilience aspects of a given network topology [8, 24, 25]. During our work on future Internet protocols, we proposed improvements to the LISP mobile node architecture [26–28,40]. As visiting researcher at the University of Vienna and in cooperation with the Vienna University of Technology, we worked on Virtual Network Embedding [29, 30]. Finally, we also investigated and optimized the placement of controllers for SDN networks [31,41,42].

In addition, we showed interest in related research areas and future Internet applications, in particular in Crowdsourcing [13] and Big Data [14].

1.2 Thesis Outline

The remainder of this monograph is structured as follows. In Chapter 2 we study the effects of IP routing optimization for different objective functions. First we analyze regular IP routing, then, the IP fast reroute mechanism LFA is taken into account. Finally, an analysis of optimizations for the PCN admission control mechanisms is presented. Chapter 3 presents the design of a new naming and addressing protocol for the future Internet. First, the general concept of our new global locator, local locator and identifier split (GLI-Split) protocol is introduced and then FIRMS, a solution for the required mapping service, is presented. Chapter 4 summarizes this work and draws conclusions. A table with nomenclature and abbreviations is provided in the appendix.

2 Optimization of IP-based Routing Protocols

Try as hard as we may for perfection, the net result of our labors is an amazing variety of imperfectness.
(Antoine de Saint-Exupéry)

Intradomain routing in IP networks follows least-cost paths according to administrative link costs. Distributed routing algorithms disseminate the cost values for all links throughout the network so that routers have a consistent view on the topology and the link costs. Based on this information, they calculate least-cost paths and install appropriate entries in the forwarding tables that are used to steer traffic quickly through the network.

Traffic engineering in IP networks is rather complex, even when considering routing only under failure-free conditions, since the path layout can be controlled only indirectly by modifying the administrative link costs. For a network with given link capacities and traffic matrix, the link cost settings can be optimized by minimizing a certain objective function like, e.g., the maximum utilization of all links in the network. This optimization problem has been extensively studied in the past and proven to be NP-complete [44].

One of the main influence factors that determines the result of the optimization is the selection of an appropriate objective function. Many different functions can be used, but the application of a specific objective function must be fine tuned to the goals that should be accomplished with the routing optimization. This chapter

investigates the influence of various objective functions on the networks performance, taking into account different protocols.

In the first section we analyze the optimization of regular IP routing and its distributed failure recovery mechanism. When links or nodes fail in an IP network, this information is propagated to all nodes so that they can recalculate the least-cost paths and modify their forwarding tables accordingly. Thus, after some time, traffic can be routed again in the remaining topology. This property makes IP networks very robust against failures since correct routes between two nodes are installed as long as they still have a physical connection. We present an improved heuristic optimization tool and heuristics that can optimize the link costs in a network to minimize a certain objective function. The tool is used for all optimizations and evaluations in this chapter. Different objective functions for optimization of failure free routing have been proposed. We analyze and compare them using the network optimization tool. Then we propose several extensions of the objective functions to include failure scenarios in the calculation process and analyze and compare the resulting routings.

The second section of this chapter examines a more sophisticated protocol, which improves failure handling. In IP networks, failures occur on a regular basis and often last only for a short time [45]. The distributed IP rerouting process as optimized in the first section is simple and robust [46], but it may be too slow for applications and services that require continuous network availability [47]. Recently, fast reroute (FRR) mechanisms have been proposed for IP networks [48]. With IP-FRR, a router can detour traffic around a failure location immediately after it has detected that the regular next-hop is no longer reachable. This reduces the time during which packets are lost from several seconds down to less than 50 ms. Then, regular IP rerouting is triggered. Therefore, the traffic affected by the failure is forwarded by IP-FRR mechanisms only until the rerouting process completes or the failure disappears. The only IP-FRR mechanism that is already standardized by the IETF and implemented in new routers, e.g., current versions of Cisco IOS and Juniper OS, is the loop-free alternates (LFAs) concept [49]. An LFA is an alternate next-hop to which certain traffic can be sent without creating

any loops so that this traffic reaches its destination over an alternative path. When the regular next-hop for a certain destination is no longer reachable by a router, it can deflect traffic to this destination over the LFA. LFAs do not require any signaling, they do not require changes to the basic IP routing protocol, and they do not require tunneling. These features facilitate incremental as well as partial deployment, even in a multi-vendor network, and make LFAs a very attractive solution. However, LFAs have also two disadvantages. First, nodes may not have LFAs for all destinations [50–52] so that some traffic cannot be protected against single link or node failures although the network topology has alternate working paths. Second, some LFAs may cause routing loops in case of node or multiple failures. In this section, we classify LFAs in three different categories according to their ability to avoid loops and propose several objective functions for IP routing optimization which can be used for different use-cases of LFA application. We show that the maximization of LFA coverage can lead to high link utilization and present a Pareto-optimization that offers the network administrator several routing options to choose from.

The third section of this chapter analyzes another protocol that is currently developed to provide higher service quality to premium users of a network. New technologies and services have significantly increased the traffic volume in carrier networks. Today, ISPs rely on capacity overprovisioning (CO) to support quality of service (QoS) in terms of packet loss and delay. In [53] admission control (AC) was proposed for IP networks and new networks should support some form of QoS assurance such as AC to enable services that cannot be provided with CO [54]. Conventional AC prevents overload due to increased user activity. If congestion occurs in core networks, it is mainly caused by failures and redirected traffic, and only to a minor degree by increased user activity [55]. Thus, resilient AC is required, so that admitted traffic can be rerouted in likely failure scenarios without causing congestion on backup paths [56]. The Internet Engineering Task Force (IETF) works on “Congestion and Pre-Congestion Notification” (PCN) [57] with the objective to standardize feedback-based admission control (AC) and flow termination (FT) for high-priority PCN traffic for single

DiffServ domains [58]. We propose algorithms to configure the link-specific PCN parameters so that resources are utilized efficiently and fairly by competing traffic aggregates while meeting resilience constraints. The results for two proposed architectures are compared and we derive objective functions to optimize the underlying routing system for both cases.

The chapter is organized as follows. In Section 2.1, the optimization of regular IP routing is examined and different objective functions are compared. Section 2.2 looks at the IP fast reroute mechanism LFA, surveys different fields of application where it can be used, and presents suitable objective functions. In Section 2.3, we fine-tune the admission control mechanism PCN and optimize the underlying routing. Finally, Section 2.4 summarizes some condensed insights.

2.1 Objective Functions for Optimization of Resilient IP Routing

Many heuristic algorithms have been proposed for the optimization of administrative link costs in IP networks. Most of them use objective functions which are based on the utilization of the links in the network. The most prominent functions are the maximum link utilization and a function proposed by Fortz in [59], which takes the load of all links into account. Other objective functions exist and some of them have been compared in [60] in a more general context than IP routing.

We consider traffic engineering for resilient IP networks where expected link loads should be low or well balanced even in case of certain outages. Traffic should be carried on appropriate paths under failure-free conditions and also in certain failure scenarios. Therefore, optimization of resilient routing requires objective functions reflecting the quality of the path layout under failure-free conditions and after rerouting in the failure scenarios of interest. Before this study, only few papers have tackled optimization of resilient IP routing. Most of them use the maximum link utilization as objective function or a specific failure-comprising extension of the function proposed by Fortz [46].

In this section, we propose additional objective functions for optimization of resilient IP routing, which intuitively extend Fortz's objective function to include failures. We investigate whether the objective functions are equivalent and lead to comparable optimization results. We examine link utilizations and path lengths in differently optimized IP routings and measure the impact of the considered objective functions on the runtime of our heuristic algorithm. Furthermore, we propose enhanced techniques to minimize computation time of objective functions and present an extension to our heuristic that allows the combined optimization of different objective functions.

The content of this section is mainly taken from a study that we published and presented in [15]. It is structured as follows. Section 2.1.1 gives an overview of related work. Section 2.1.2 introduces various objective functions for resilient and non-resilient routing optimization. In Section 2.1.3 we review our optimization approach and propose the new combined optimization and computation speedup techniques. Section 2.1.4 investigates various objective functions for optimization of resilient and non-resilient IP routing.

2.1.1 Related Work on Routing Optimization

We briefly review existing work regarding the optimization of IP routing with and without resilience requirements.

2.1.1.1 Optimization of Non-Resilient IP Routing

The problem of IP routing optimization without resilience requirements is NP-complete [44, 61]. Some papers solve the problem by (integer) linear programs [44, 62–70]. Since the search space is rather large, others prefer fast heuristics and use local search techniques [59], genetic algorithms [71–77], simulated annealing, or other heuristics [78, 79]. Riedl et al. consider non-additive link costs and thereby increase the solution space [80]. Xu et al. propose a new link state routing protocol PEFT that is based on link costs and can achieve optimal traffic engineering [81].

The papers use various objective functions. The most often applied objective function is the minimization of the maximum utilization of all links [67,72,74,76, 80,81]. A similar objective is the maximization of unused bandwidth [78]. Others combine the maximum link utilization and some other performance objective [62,65]. In contrast, Fortz et al. propose a more complex objective function where the utilization of each link in the network contributes to the target value using a continuous, piece-wise linear, monotonically increasing penalty function [59]. This function has been adopted by many other researchers [71, 79, 81].

2.1.1.2 Optimization of Resilient IP Routing

Optimization of resilient IP routing improves the path layout for the failure-free scenario and for a set of protected failure cases. First solutions to this problem were presented in [46, 82, 83] for single link failures. The presented algorithms use a local search technique combined with a tabu list or a hash function to mark solutions already visited. To escape from local minima, [46] sets some link weights to random values. To speed up the algorithm, [82] investigates only a random fraction of possible neighboring configurations (link cost settings) while [83, 84] benefit from a heuristic choosing appropriate neighboring configurations that are explored next. To accelerate the computation speed, [85] evaluates the objective function only for a reduced set of critical links instead of the entire set of protected failure scenarios. We have presented another heuristic for the optimization of resilient IP routing based on the idea of threshold accepting [16]. In [86] the efficiency of heuristic methods based on tabu search and steepest ascent were compared with bounds provided by mixed integer programs. Fortz et al. also propose to modify a few link costs to improve the current load situation when a link fails or when the traffic matrix changes. Thus, they tackle this problem by configuring new link costs rather than to find link cost settings that perform well for a given set of conditions [87, 88].

Optimization of resilient IP routing requires different objective functions than optimization of IP routing for the failure-free scenario. The maximum utilization of all links in all protected failure scenarios has been minimized in [16]. The

authors of [83] use a combination of the maximum link utilization in the failure-free scenario and the maximum link utilization in all protected failure scenarios as objective function which needs to be minimized. In [84] they include another performance metric motivated by service level agreements. The authors of [86] maximize the minimum unused capacity on the links in all protected failure scenarios. Fortz et al. extend their objective function previously proposed for the failure-free case by computing its value for the failure-free case and for all protected failure scenarios; their new objective function for resilient routing consists of half the value for the failure-free scenario and half the average of the value of all failure scenarios [46]. Sridharan et al. [85] use a generalized weighted average of these values. Taking the maximum over the values that are obtained by Fortz's function for all failure scenarios has been mentioned by Yuan [82]. Equal-cost paths may occur in IP networks which may be good for load balancing purposes, but bad for prediction of load distribution. Therefore, we proposed a method to optimize for unique shortest paths [20]. In [16] and in the later sections of this chapter we consider more complex optimization goals that take into account other technological constraints.

2.1.2 Objective Functions for Routing Optimization

In the following, we introduce the used nomenclature and present various objective functions for resilient and non-resilient routing optimization.

2.1.2.1 Nomenclature

We model a network topology by a graph consisting of a set[1] of nodes (vertices) $\mathcal{V}$ and a set of directed links (edges) $\mathcal{E}$. We describe a failure scenario $s \subseteq (\mathcal{V} \cup \mathcal{E})$ by the set of failed elements. The failure-free scenario is denoted by $s = \emptyset$, and $\mathcal{S}$ describes the set of all considered scenarios. In the remainder of this section, we usually consider the failure free scenario together with all single bidirectional link failure scenarios, i.e., $\mathcal{S} = \{\emptyset\} \cup \{\{l\} : l \in \mathcal{E}\}$.

[1] Calligraphic letters $\mathcal{X}$ denote sets and the operator $|\mathcal{X}|$ indicates the cardinality of a set.

We represent bandwidths and administrative link costs of all links by the vectors[2] $\mathbf{c}$ and $\mathbf{k}$. The link costs are integers between $k_{\min} = 1$ and $k_{\max}$, thus, they are taken from a vector space with $(k_{\max})^{|\mathcal{E}|}$ elements. The default routing that is used for comparison uses uniform link costs (e.g., all link costs set to 1) and it is denoted by $\mathbf{k}_{\mathrm{u}} = \mathbf{1}$. The routing in IP networks depends on the administrative link costs $\mathbf{k}$ and the specific set s of failed elements. The traffic aggregates (demands) between the pairs of different nodes constitute the traffic matrix $\mathcal{D}$. An aggregate $d_{v,w} \in \mathcal{D}$ has a source node v and destination node $w \in \mathcal{V}$, and its rate is given by $r(d_{v,w})$. If detailed information is not necessary, the simplified forms d and $r(d)$ are used. The function $u_s^{\mathbf{k}}(l, v, w)$ indicates the fraction of traffic from v to w that is carried over link l in failure scenario s when link costs $\mathbf{k}$ apply. This description models both single-path and multipath routing.

The utilization $\rho(\mathbf{k}, l, s)$ of a link l in a failure scenario s, the utilization $\rho_{\mathcal{E}}^{\max}(\mathbf{k}, s)$ of the highest loaded link $l \in \mathcal{E}$ in scenario s, the maximum utilization $\rho_{\mathcal{S}}^{\max}(\mathbf{k}, l)$ of link l in all failure scenarios $s \in \mathcal{S}$, and the maximum utilization $\rho_{\mathcal{S},\mathcal{E}}^{\max}(\mathbf{k})$ of all links $l \in \mathcal{E}$ in all failure scenarios $s \in \mathcal{S}$ are calculated for any link cost vector $\mathbf{k}$ by

$$\rho(\mathbf{k}, l, s) = \left(\sum_{v,w \in \mathcal{V}} u_s^{\mathbf{k}}(l, v, w) \cdot r(d_{v,w}) \right) / c(l), \tag{2.1}$$

$$\rho_{\mathcal{E}}^{\max}(\mathbf{k}, s) = \max_{l \in \mathcal{E}} \left(\rho(\mathbf{k}, l, s) \right), \tag{2.2}$$

$$\rho_{\mathcal{S}}^{\max}(\mathbf{k}, l) = \max_{s \in \mathcal{S}} \left(\rho(\mathbf{k}, l, s) \right), \tag{2.3}$$

$$\rho_{\mathcal{S},\mathcal{E}}^{\max}(\mathbf{k}) = \max_{l \in \mathcal{E}} \left(\rho_{\mathcal{S}}^{\max}(\mathbf{k}, l) \right). \tag{2.4}$$

Fortz et al. [59] explain that it is cheap to send traffic over links with small utilization, but gets increasingly more expensive when a link utilization approaches its capacity. Therefore, they define a function ϕ that depends on the link utilization and penalizes high values. The function ϕ (cf. Figure 2.1) is continu-

[2] A link-specific property x is denoted in a compact way by a vector $\mathbf{x}$ that is printed boldface. The indexed components of a vector are denoted by $\mathbf{x}(l)$ with $l \in \mathcal{E}$.

ous and piecewise linear as some integer linear program problem solvers require these properties for their optimization. Furthermore, it is monotonically increasing (concave) in order to favor short paths in the network (cf. Section 2.1.4.3) while mitigating high link utilizations. It is defined by $\phi(0) = 0$ and its derivative

$$\phi'_a(x) = \begin{cases} 1 & \text{for} & 0 & \leq x < & 1/3 \\ 3 & \text{for} & 1/3 & \leq x < & 2/3 \\ 10 & \text{for} & 2/3 & \leq x < & 9/10 \\ 70 & \text{for} & 9/10 & \leq x < & 1 \\ 500 & \text{for} & 1 & \leq x < & 11/10 \\ 5000 & \text{for} & 11/10 & \leq x < & \infty \end{cases} . \tag{2.5}$$

The general Fortz function for a given scenario $s \in S$ is shown in Equation (2.6). In contrast to the original definition in [59] we normalized the sum by the number of links

$$\Phi(s) \quad = \quad \frac{1}{|\mathcal{E}|} \sum_{l \in \mathcal{E}} \phi_a(\rho(\mathbf{k}, l, s)). \tag{2.6}$$

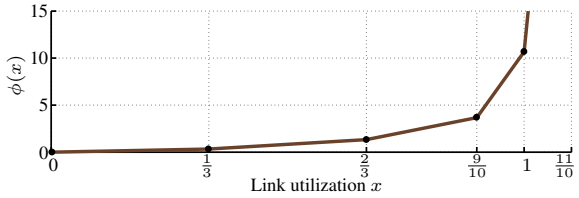


Figure 2.1: *Fortz's utilization-dependent penalty function* ϕ.

2.1.2.2 Objective Functions for Non-Resilient IP Routing

All objective functions calculate a benchmark value for the routing, based on the current link costs $\mathbf{k}$. For routing optimization without resilience requirements, the following two objective functions are mainly used.

$\rho_{\emptyset}^{\max}$: The maximum link utilization, which has been used, e.g., in [16], reflects only the utilization of the most loaded link. $\rho_{\emptyset}^{\max} = \rho_{\mathcal{E}}^{\max}(\mathbf{k}, \emptyset)$.

$\Phi_{\emptyset}$: This is the general Fortz function applied to failure free routing: $\Phi_{\emptyset} = \Phi(\emptyset)$. In contrast to $\rho_{\emptyset}^{\max}$, it reflects the load level of all links.

2.1.2.3 Objective Functions for Resilient IP Routing

For resilient routing optimization, the objective functions $\rho_{\emptyset}^{\max}$ and $\Phi_{\emptyset}$ need to be extended to reflect the link utilizations in all protected scenarios $s \in \mathcal{S}$.

$\rho_{\mathcal{S}}^{\max}$: We extend objective function $\rho_{\emptyset}^{\max}$ for resilient routing by defining

$$\rho_{\mathcal{S}}^{\max} = \rho_{\mathcal{S},\mathcal{E}}^{\max}(\mathbf{k}). \tag{2.7}$$

We used this function in [16]. A similar function has been used by Nucci et al. [83]. They composed a weighted average of the failure free performance and the maximum utilization over all failures:

$$\rho_{\mathcal{S}}^{\text{weighted}} = (1 - w) \cdot \rho_{\emptyset}^{\max} + w \cdot \rho_{\mathcal{S}}^{\max}. \tag{2.8}$$

$\Phi_{\mathcal{S}}^{\text{weighted}}$: Fortz et al. [46] calculate a weighted average and give equal importance to the failure-free scenario as to all other protected scenarios together. Hence, we get

$$\Phi_{\mathcal{S}}^{\text{weighted}} = \frac{1}{2} \cdot \Phi_{\emptyset} + \frac{1}{2} \cdot \frac{1}{|\mathcal{S}| - 1} \cdot \sum_{s \in \mathcal{S}, s \neq \emptyset} (\Phi(s)). \tag{2.9}$$

We also examined the simple equal-weighted average $\Phi_{\mathcal{S}}^{\text{avg}}$:

$$\Phi_{\mathcal{S}}^{\text{avg}} = \frac{1}{|\mathcal{S}|} \cdot \sum_{s \in \mathcal{S}} (\Phi(s)). \tag{2.10}$$

It produces similar results as $\Phi_{\mathcal{S}}^{\text{weighted}}$, but takes longer to calculate as explained in the following section. Thus, only the interesting computation time results are presented for this objective function.

$\Phi_{\mathcal{S}}^{\text{max,out}}$: Another option is to take the maximum $\Phi(s)$ of all considered failure scenarios $s \in \mathcal{S}$, which results in

$$\Phi_{\mathcal{S}}^{\text{max,out}} = \max_{s \in \mathcal{S}}(\Phi(s)). \tag{2.11}$$

A function based on this is used, e.g., in [82].

$\Phi_{\mathcal{S}}^{\text{max,in}}$: We extend the objective function $\Phi_{\emptyset}$ for resilience purposes by substituting the utilization $\rho(\mathbf{k}, l, \emptyset)$ of a link in the failure-free scenario by its maximum utilization $\rho_{\mathcal{S}}^{\max}(\mathbf{k}, l)$ in all considered failure scenarios $s \in \mathcal{S}$. This results in

$$\Phi_{\mathcal{S}}^{\text{max,in}} = \frac{1}{|\mathcal{E}|} \cdot \sum_{l \in \mathcal{E}} \phi(\rho_{\mathcal{S}}^{\max}(\mathbf{k}, l)). \tag{2.12}$$

2.1.3 Heuristic Optimizer

To compare the quality of different objective functions, we built a heuristic routing optimizer, based on previous work [16]. In the following explanations, we denote a generic objective function with Ψ, and if the dependency on a special $\mathbf{k}_{\text{spec}}$ is important, we write $\Psi(\mathbf{k}_{\text{spec}})$. We present a new optimized objective calculation process which lowers computation times drastically. We also developed a new combined optimization technique, which can improve two different objective functions simultaneously.

2.1.3.1 Optimization Algorithm

Finding the optimal routing solution for a given network and traffic matrix is an NP-complete problem. The search space of possible link cost settings is very large and enumeration of all settings is impossible even for very small networks. Thus, a heuristic is required for the optimization process. We implemented threshold accepting, a probabilistic heuristic that tries to minimize the objective functions in the network.

Threshold accepting starts with a random initialization of all link costs $\mathbf{k}$ with values between 1 and k_{max}. At each iteration step, the algorithm randomly selects a (random) number of links whose link costs are (randomly) changed. This new link cost setting $\mathbf{k}_{\text{new}}$ results in a new routing, and thus, a new objective function value Ψ_{new}. If Ψ_{new} is better than ever before, the algorithm stores this value in Ψ_{best} and also stores $\mathbf{k}_{\text{best}}$ to return it later as final result when no better value is found until then. If Ψ_{new} is not worse than a threshold T above the current best value ($\Psi_{\text{new}} \leq \Psi_{\text{best}} + T$), the link costs $\mathbf{k}_{\text{new}}$ are accepted as starting point for the next iteration. The threshold is introduced to increase the chance to escape from one of the numerous local minima and find the global minimum. If the new value is above the threshold, the next iteration starts with the previous $\mathbf{k}$. Finally, if no new Ψ_{best} is found after i_{max} iterations, Ψ_{best} and $\mathbf{k}_{\text{best}}$ are returned as result. The algorithm can then be restarted with a different seed, resulting in a different initialization and other random neighbors so that new areas of the link cost search space are explored.

2.1.3.2 Combined Optimization

Some objective functions only consider specific attributes of a path layout. For example, $\rho_{\emptyset}^{\text{max}}$ only looks at the link with the highest utilization. All other links are not regarded at all, but routing on them can still be improved using another objective function. Thus, we extended the threshold accepting heuristic to perform combined optimization of two objective functions Ψ_1 and Ψ_2. The primary objective function Ψ_1 is minimized as before. When a new cost set-

ting $\mathbf{k}_{\text{new}}$ results in $\Psi_1(\mathbf{k}_{\text{new}}) = \Psi_1(\mathbf{k}_{\text{best}})$, it is only accepted as new best when $\Psi_2(\mathbf{k}_{\text{new}}) < \Psi_2(\mathbf{k}_{\text{best}})$. When $\Psi_1(\mathbf{k}_{\text{new}}) < \Psi_1(\mathbf{k}_{\text{best}})$, then it is accepted (and $\mathbf{k}_{\text{best}}$ is set to $\mathbf{k}_{\text{new}}$) regardless of the performance of the secondary objective function. Thus, Ψ_2 can also increase as more importance is given to Ψ_1.

Note that the computational overhead of the combined optimization is very low since Ψ_2 must only be calculated if $\Psi_1(\mathbf{k}_{\text{new}}) = \Psi_1(\mathbf{k}_{\text{best}})$. Also, the link utilizations are already calculated for Ψ_1 so that no new routing calculation must be performed. Figure 2.2 presents a typical combined optimization run. It shows the values of two different optimization functions. The upper line represents the current best value of the primary objective function $\Psi_1(\mathbf{k}_{\text{best}})$ which only decreases. The bottom points are the current values of the secondary objective function $\Psi_2(\mathbf{k}_{\text{best}})$.

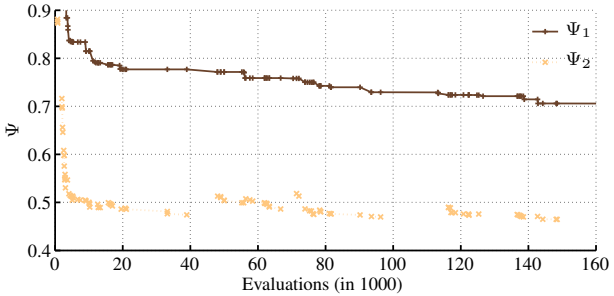


Figure 2.2: *Evolution of two objective values Ψ_1 and Ψ_2 in a combined optimization run. Ψ_1 is almost constantly improved while Ψ_2 also degrades.*

2.1.3.3 Improved Computation of Objective Functions

The most time-consuming step in the optimization algorithm is the calculation of the objective function Ψ. Dijkstra's algorithm must be called $|\mathcal{V}|$-times to calculate the routing towards each node $v \in \mathcal{V}$. Then, link utilizations must be calculated from these shortest paths. These complex calculations have to be repeated for every protected scenario $s \in \mathcal{S}$.

Sridharan et al. [85] accelerate the evaluation of the objective function by considering only a set of critical failures instead of the entire set of protected failure scenarios $\mathcal{S}$. This requires careful selection of the included failures. The critical scenarios can also change when routing is changed. Thus, the optimization could run in a wrong direction until a new set of critical links is chosen.

Our algorithm has three speedup improvements. First, it takes the previous Ψ and the current threshold T as input when calculating Ψ_{new}. In each iteration we iterate linearly over a list of all failure scenarios and incrementally calculate Ψ_{new}. The calculation can then be stopped as soon as it is clear that Ψ_{new} will be larger than $\Psi + T$. Second, we check which failure scenario had the largest impact on Ψ_{new} and move this scenario to the front of the scenario list. Then, subsequent calculations of Ψ start with the evaluation of previous worst-case scenarios, and in many cases, the calculation can be aborted early. The third improvement is incremental routing calculation, where only the failure-free routing is calculated completely. In each failure scenario, only the flows that used a failed network element need to be recalculated, while the remaining routing stays the same. All these features together lead to a very fast heuristic optimization and allow us to run a multitude of experiments.

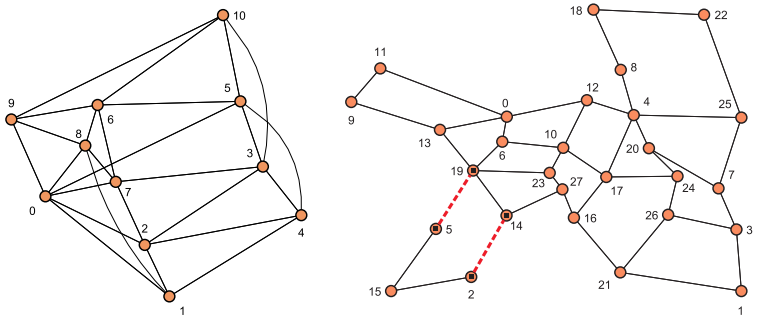
2.1.4 Runtime Analysis and Comparison of Objective Functions

We present the networks under study and look at the average runtime required by our heuristic to compute different objective functions. Then, we compare properties of optimized link cost settings achieved through different objective functions. We first consider non-resilient IP routing, since it is easier to analyze, and then resilient IP routing.

2.1.4.1 Experiment Setup

We use well-known realistic networks for our experiments: COST239 [89] and NOBEL [90]. We also evaluated the GEANT [91] and Labnet03 [56] topologies,

but we present here mainly the results for COST239 and some results for NOBEL (see Figures 2.3(a) and (b)), since the results from the other networks do not yield additional insights. As benchmark for the optimization we use the unoptimized routing with uniform link costs where each link cost is set to the same value (e.g. $\mathbf{k}_u = \mathbf{1}$). The resulting paths are shortest paths with respect to hop count (HC). The equal-cost multipath (ECMP) routing option is used when multiple equal-cost paths exist. For the optimization and analysis of resilient routing we consider all single bidirectional link failures. We use population-based traffic matrices [56] and scale them so that the maximum link utilization reaches 100% in the worst link failure scenario for routing with uniform link costs.



(a) COST239 network with 11 nodes and 52 unidirectional links.

(b) NOBEL network with 28 nodes and 82 unidirectional links.

Figure 2.3: *Networks under study.*

We run our threshold accepting heuristic for routing optimization with a small threshold $T = 0.001$, a maximum link cost $k_{\max} = 10$, and a high number of unsuccessful iterations $i_{\max} = 100,000$. The small threshold results from previous work and evaluations, where we discovered that larger thresholds can sometimes improve the result at the cost of much longer runtimes and worse average re-

sults. The same applies for the relatively small upper limit for the link costs. Larger numbers only increase the search space but usually do not improve the resulting best routing. On the contrary, the average results deteriorate slightly for ECMP routing as a larger search space makes the accidental discovery of equal length path unlikely [20, 92]. Most optimization runs tested between 200,000 and 350,000 different link cost settings. We have optimized every network for every objective function during 24 hours, i.e., when a run was finished, the heuristic was restarted with a new seed. The number of performed optimization runs is in the magnitude of about 1000 and strongly depends on the used objective functions and whether resilience was needed or not. Usually, only the final link cost setting of the best optimization run is returned. This best link cost setting for an objective function Ψ is denoted by $\mathbf{k}^{\Psi}$. For the following evaluations, we stored the results of all runs to calculate average results, to compare different functions, and to re-evaluate the results optimized with one function with all other objective functions.

2.1.4.2 Average Evaluation Time of Objective Functions

The optimization step requires the evaluation of the objective function for each considered link cost setting $\mathbf{k}_{\text{new}}$. This includes the calculation of the routing and the link utilization for each protected failure scenario $s \in \mathcal{S}$. Thus, optimizing resilient routing is usually more complex than only optimizing the failure-free scenario.

Section 2.1.3.3 describes methods for computation speedups and here we show their effectiveness depending on the objective function. When calculations are aborted early and worst scenarios are moved to the front of the list, on average only a few scenarios must be evaluated during the search for a new Ψ_{best}. In addition, incremental routing calculation speeds up the computation of the routing in failure scenarios. Table 2.1 lists average computation times for the (possibly early aborted) evaluation of objective functions during a regular optimization run. It also shows the average number of evaluated scenarios.

Table 2.1: *Average computation effort and average number of scenario evaluations for different objective functions.*

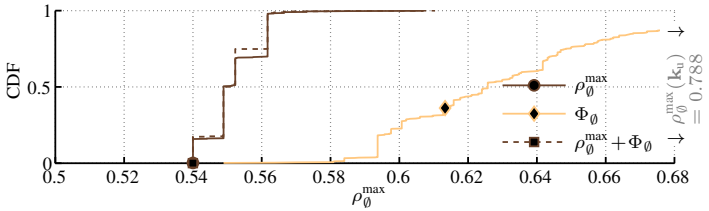
Network	$\rho_{\emptyset}^{\max}$	$\Phi_{\emptyset}$	$\rho_{\mathcal{S}}^{\max}$	$\Phi_{\mathcal{S}}^{\text{avg}}$	$\Phi_{\mathcal{S}}^{\text{weighted}}$	$\Phi_{\mathcal{S}}^{\max,\text{out}}$	$\Phi_{\mathcal{S}}^{\max,\text{in}}$
COST239	0.30ms	0.29ms	0.35ms	0.46ms	0.42ms	0.33ms	0.38ms
$\lvert\mathcal{S}\rvert = 27$	/ 1	/ 1	/ 1.88	/ 7.12	/ 5.69	/ 1.30	/ 2.86
NOBEL	2.47ms	2.47ms	3.50ms	8.00ms	5.71ms	2.97ms	3.96ms
$\lvert\mathcal{S}\rvert = 42$	/ 1	/ 1	/ 3.29	/ 21.20	/ 12.36	/ 1.57	/ 4.20

Especially the computation of objective functions that include a maximum can be aborted early. For $\rho_{\mathcal{S}}^{\max}$, on average only 1.88 out of 27 scenarios are evaluated in the COST239 network and in the NOBEL network 3.29 out of 42. All protected scenarios $s \in \mathcal{S}$ need to be computed only when $\Psi(\mathbf{k}_{\text{new}})$ is close to the current best result.

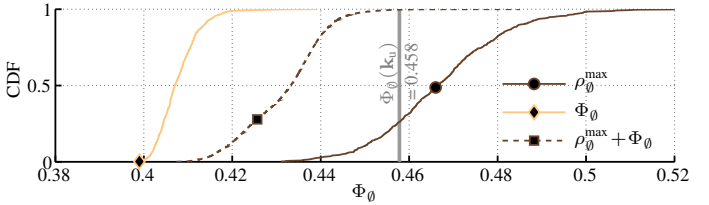
When evaluating an objective function that calculates an average, e.g. the extended Fortz function $\Phi_{\mathcal{S}}^{\text{avg}}$, more scenarios need to be computed before a worse $\mathbf{k}_{\text{new}}$ can be rejected. As each scenario contributes only a small share to the average value, many scenarios need to be computed before the currently computed average value can exceed a reference value. The evaluation of $\Phi_{\mathcal{S}}^{\text{avg}}$ for the COST239 network was aborted on average after the computation of 7.12 out of 27 scenarios. In the NOBEL network, even 21.20 out of 42 scenarios were computed on average. The evaluation of $\Phi_{\mathcal{S}}^{\text{weighted}}$ can be aborted earlier because the failure-free scenario is always computed first and contributes half to the weighted average. Therefore, compared to $\Phi_{\mathcal{S}}^{\text{avg}}$, for $\Phi_{\mathcal{S}}^{\text{weighted}}$ fewer other scenarios need to be computed until the interim weighted average exceeds a reference value. The average calculation times in milliseconds show the effect of the incremental routing calculation. Even if many scenarios have to be computed, the additional overhead is low. For example, the evaluation of the single failure-free scenario in $\rho_{\emptyset}^{\max}$ takes 2.47ms in the NOBEL network. Evaluating 21.2 scenarios instead of a single one for the calculation of $\Phi_{\mathcal{S}}^{\text{avg}}$ takes only 8ms which is just 3.24 times longer.

2.1.4.3 Optimization of Non-Resilient IP Routing

We investigate whether the optimization with the objective functions $\rho_{\emptyset}^{\max}$ and $\Phi_{\emptyset}$ leads to link cost settings that are also good in the view of the other. Furthermore, we study average path lengths and link utilizations caused by the different objective functions. Besides, we illustrate the impact of combined optimization using two different objective functions $\rho_{\emptyset}^{\max} + \Phi_{\emptyset}$.



(a) CDFs of maximum link utilizations $\rho_{\emptyset}^{\max}$.



(b) CDFs of Fortz objective function values $\Phi_{\emptyset}$.

Figure 2.4: *Cumulative distribution of objective function values of all link cost settings optimized for non-resilient IP routing, obtained within 24 hours for the COST239 network (lines: optimization objective, small markers: best result according to each objective function).*

Mutual optimization We would like to know whether link cost settings optimized with one objective function lead also to good values when evaluated by other objective functions. Therefore, we have evaluated the $\rho_{\emptyset}^{\max}$ and $\Phi_{\emptyset}$ values for all link cost settings optimized with $\rho_{\emptyset}^{\max}$, $\Phi_{\emptyset}$, and $\rho_{\emptyset}^{\max} + \Phi_{\emptyset}$. Figure 2.4(a)

shows the cumulative distribution function (CDF) of the $\rho_{\emptyset}^{\max}$ values for the optimized link cost settings in the COST239 network. The small marks denote the best link cost settings $\mathbf{k}^{\Psi}$ according to the three evaluated objective functions Ψ. Link cost settings optimized with $\rho_{\emptyset}^{\max}$ or $\rho_{\emptyset}^{\max} + \Phi_{\emptyset}$ mostly have maximum link utilizations $\rho_{\emptyset}^{\max}$ between 0.54 and 0.57. This is a significant improvement of about 30% since the $\rho_{\emptyset}^{\max}$ value for unoptimized routing with uniform link costs is 0.79. The distinct steps in the CDF imply that there are many link cost settings which lead to the same $\rho_{\emptyset}^{\max}$ value. The maximum link utilizations $\rho_{\emptyset}^{\max}$ for the link cost settings optimized with $\Phi_{\emptyset}$ are significantly larger and are spread over a wide utilization range. Nevertheless, they are clearly smaller than the maximum link utilization with unoptimized uniform link costs $\mathbf{k}_{\mathrm{u}}$. The best link cost setting for $\Phi_{\emptyset}$ has a relatively large maximum link utilization of $\rho_{\emptyset}^{\max}(\mathbf{k}^{\Phi_{\emptyset}}) = 0.613$.

Figure 2.4(b) presents the CDFs of Fortz's original objective function $\Phi_{\emptyset}$ for all optimized link cost settings. All CDFs have a continuous shape which means that only very few optimized link cost settings have the same $\Phi_{\emptyset}$ value, regardless of the objective function used for their optimization. The $\Phi_{\emptyset}$ values for link cost settings optimized with $\Phi_{\emptyset}$ are significantly smaller than those for link cost settings optimized with $\rho_{\emptyset}^{\max}$ only. About 75% of the latter are even worse than the $\Phi_{\emptyset}$ value of routing with unoptimized uniform link costs $\mathbf{k}_{\mathrm{u}}$. However, the link cost settings of the combined optimization with $\rho_{\emptyset}^{\max} + \Phi_{\emptyset}$ lead to significantly improved $\Phi_{\emptyset}$ values. The best link cost settings $\mathbf{k}^{\rho_{\emptyset}^{\max}}$ and even $\mathbf{k}^{\rho_{\emptyset}^{\max}+\Phi_{\emptyset}}$, which are primarily optimized to minimize the maximum link load, lead to rather high Fortz function values $\Phi_{\emptyset}$. The simple optimization even leads to Fortz values that are larger than the value for unoptimized routing with uniform link costs: $\Phi_{\emptyset}(\mathbf{k}^{\rho_{\emptyset}^{\max}}) > \Phi_{\emptyset}(\mathbf{k}_{\mathrm{u}})$. This shows that link cost settings with good $\rho_{\emptyset}^{\max}$ values can produce rather moderate $\Phi_{\emptyset}$ values.

Average path lengths Figure 2.5 shows CDFs of the average path lengths for IP routing with optimized link cost settings. Unoptimized routing with uniform link costs $\mathbf{k}_{\mathrm{u}}$ leads to the shortest possible average path length of 1.564. Routing optimization with objective function $\rho_{\emptyset}^{\max}$ is likely to move traffic away

from shortest paths when they have a large utilization otherwise. This leads to a longer average path length.

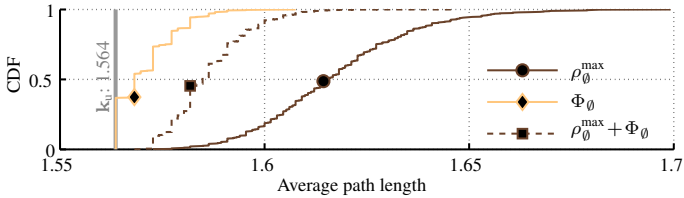


Figure 2.5: *CDFs of average path lengths for all link cost settings optimized for non-resilient IP routing (lines: optimization objective, small markers: best result according to each objective function).*

Imagine an additive objective function similar to $\Phi_\emptyset$, whose utilization-dependent penalty function is purely linear. Its lowest value is achieved if traffic is carried on shortest paths, but it does not offer incentives to move traffic away from paths with high utilization. The utilization-dependent penalty in Fortz's objective function $\Phi_\emptyset$ has different slopes. Its objective value increases if traffic is not carried over shortest paths, unless the utilizations of some links are thereby reduced under a lower threshold of the penalty function. Therefore, link cost settings optimized with $\Phi_\emptyset$ lead to shorter path lengths compared to those optimized with $\rho_\emptyset^{max}$. The combined optimization with $\rho_\emptyset^{max} + \Phi_\emptyset$ significantly reduces the average path length compared to optimization with $\rho_\emptyset^{max}$ only.

Link utilizations Figure 2.6 shows the complementary CDF (CCDF) of the link utilizations for the three link cost settings with the best objective function values in the COST239 network. The logarithmic y-axis makes differences in the high utilization range more visible. Routing with uniform link costs $\mathbf{k}_u$ leads to rather high utilization values on a few links. We use it as reference for comparison. The link cost setting $\mathbf{k}^{\Phi_\emptyset}$ optimized to minimize the Fortz function value $\Phi_\emptyset$ reduces the high utilization values and increases the load on many other links. The link cost setting $\mathbf{k}^{\rho_\emptyset^{max}}$ optimized with $\rho_\emptyset^{max}$ minimizes the maximum link uti-

lization dramatically but also increases the load on many other lightly utilized links. The link cost setting obtained from the combined optimization $\rho_{\emptyset}^{\max} + \Phi_{\emptyset}$ limits the maximum link utilization to the same value as the link cost setting optimized with $\rho_{\emptyset}^{\max}$ but it increases the load of fewer links.

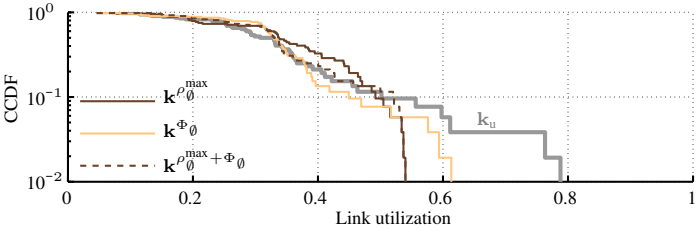


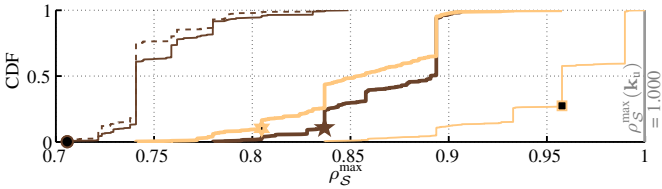
Figure 2.6: *CCDF of the link utilization for the link cost settings with the best objective function values (COST239 network).*

2.1.4.4 Optimization of Resilient IP Routing

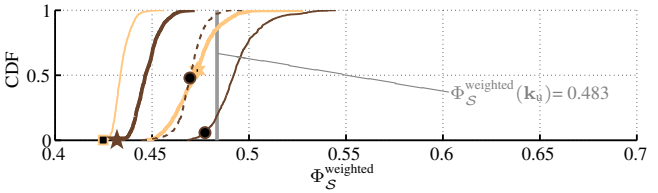
We consider the potential of mutual optimization of objective functions for resilient IP routing optimization and report on path lengths and maximum link utilizations. The NOBEL network has extreme bottlenecks in failure cases. We show that combined optimization is particularly useful for resilient IP routing optimization in such cases.

Mutual optimization Similar to Figures 2.4(a) and (b), Figures 2.7(a) - (d) show the CDFs of specific objective functions for link cost settings which were optimized with different objective functions in the COST239 network. The objective function values for the best link cost settings $\mathbf{k}^{\Psi}$ as well as those for routing with uniform link costs $\mathbf{k}_u$ are also indicated.

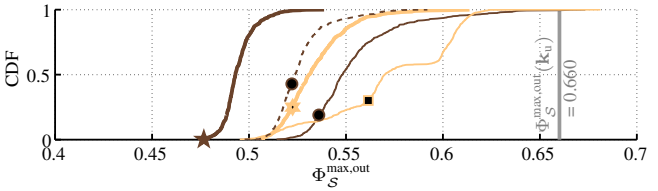
We observe that the four considered objective functions produce values in different ranges. $\rho_{\mathcal{S}}^{\max}$ looks only at the maximum utilization values and, therefore, leads to different numbers than the other functions whose values are derived from



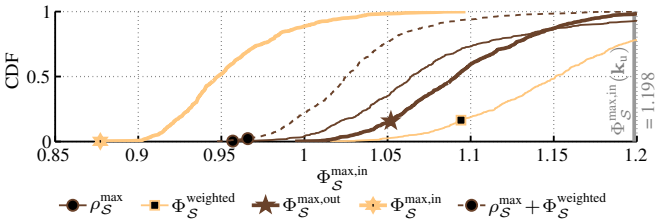
(a) CDFs of maximum link utilization values $\rho_{\mathcal{S}}^{\max}$.



(b) CDFs of weighted Fortz function values $\Phi_{\mathcal{S}}^{\text{weighted}}$.



(c) CDFs of Fortz function values for the worst failure scenarios $\Phi_{\mathcal{S}}^{\max,\text{out}}$.



(d) CDFs of Fortz function values $\Phi_{\mathcal{S}}^{\max,\text{in}}$ with links at their maximum utilization.

Figure 2.7: *CDFs of objective functions for all link cost settings optimized for resilient IP routing (lines: optimization objective).*

the utilization-dependent penalty function in Figure 2.1. The values obtained for $\Phi_{\mathcal{S}}^{\text{weighted}}$ are larger than those for the failure-agnostic $\Phi_{\emptyset}$ in Figure 2.4(b), followed by $\Phi_{\mathcal{S}}^{\text{max,out}}$ and $\Phi_{\mathcal{S}}^{\text{max,in}}$. Due to failures they use larger utilization values as arguments and also various maximization operations lead to larger values. However, it does not make sense to compare values from different objective functions with each other.

The smallest values for a specific objective function Ψ are achieved by link cost settings that were optimized for Ψ. The figure also shows that link cost settings with the best optimized objective value for one objective function lead only to moderate other objective values. For all objective functions except for $\Phi_{\mathcal{S}}^{\text{weighted}}$, routing optimization with any objective function achieves a significant improvement compared to unoptimized routing with uniform link costs. Compared to $\rho_{\mathcal{S}}^{\max}$, the combined optimization $\rho_{\mathcal{S}}^{\max} + \Phi_{\mathcal{S}}^{\text{weighted}}$ improves the values for all considered objective functions. Link cost settings optimized with $\Phi_{\mathcal{S}}^{\text{weighted}}$ lead to the worst values for all other objective functions. However, this latter finding depends on the particular network under study.

Average path lengths We report essentially the same findings for average path lengths for optimized resilient IP routing as for optimized non-resilient IP routing in Section 2.1.4.3. Unoptimized routing produces the shortest paths but link cost settings optimized with $\Phi_{\mathcal{S}}^{\text{weighted}}$ also lead to quite short paths. Link cost settings optimized with $\rho_{\mathcal{S}}^{\max}$ lead to significantly longer average path lengths, but combined optimization with $\rho_{\mathcal{S}}^{\max} + \Phi_{\mathcal{S}}^{\text{weighted}}$ reduces these values. The relation of the average path lengths of $\Phi_{\mathcal{S}}^{\text{max,out}}$ and $\Phi_{\mathcal{S}}^{\text{max,in}}$ to those of the other objective function depends on the network.

Maximum link utilization With resilient routing, we consider the maximum utilization of each link over all failure scenarios. Unoptimized routing ($\mathbf{k}_{\text{u}}$) produces the largest maximum utilization values on a few links (cf. Figure 2.8). Link cost settings optimized with $\rho_{\mathcal{S}}^{\max}$ lead to rather low maximum utilization values but some links that carried only little traffic with $\mathbf{k}_{\text{u}}$ carry significantly

more load. Link cost settings optimized with $\Phi_{\mathcal{S}}^{\text{weighted}}$ also lead to large maximum link utilization values, but many links have a lower maximum utilization compared to link cost settings optimized with $\rho_{\mathcal{S}}^{\max}$. Combined optimization with $\rho_{\mathcal{S}}^{\max} + \Phi_{\mathcal{S}}^{\text{weighted}}$ yields the same upper bounds for maximum utilization values as $\mathbf{k}^{\rho_{\mathcal{S}}^{\max}}$ and lightly loaded links carry slightly less traffic. The relation of the CDFs of the maximum link utilization of $\Phi_{\mathcal{S}}^{\max,\text{out}}$ and $\Phi_{\mathcal{S}}^{\max,\text{in}}$ to those of the other objective function depends on the network.

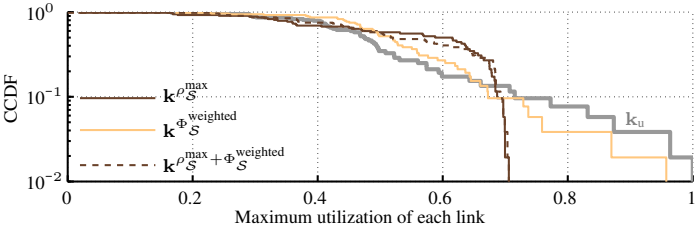
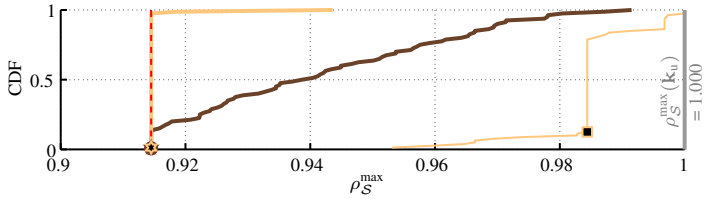


Figure 2.8: *CCDF of the maximum link utilization over all single link failure scenarios for the link cost settings with the best objective function values (COST239 network).*

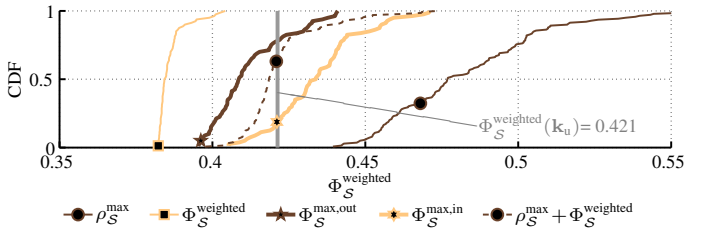
Effect of combined optimization in the presence of bottlenecks We now consider the NOBEL network. When one of the links $14 \leftrightarrow 2$ or $19 \leftrightarrow 5$ (highlighted in Figure 2.3(b)) fails, the other link has a high utilization of 91.45% since nodes 2, 5, and 15 are reachable from the rest of the network only via this link. This sets a lower bound for the maximum link utilization $\rho_{\mathcal{S}}^{\max}$.

Figures 2.9(a) and 2.9(b) illustrate the effect of resilient IP routing optimization. Figure 2.9(a) shows that all link cost settings optimized with $\rho_{\mathcal{S}}^{\max}$, $\Phi_{\mathcal{S}}^{\max,\text{in}}$, and the combined optimization $\rho_{\mathcal{S}}^{\max} + \Phi_{\mathcal{S}}^{\text{weighted}}$ find a routing that minimizes the maximum link utilization to 91.45%. However, most link cost settings optimized with $\Phi_{\mathcal{S}}^{\max,\text{out}}$ exceed this value and link cost settings optimized with $\Phi_{\mathcal{S}}^{\text{weighted}}$ are far from reaching this bound at all. Conversely, Figure 2.9(b) shows that the link cost settings optimized with $\rho_{\mathcal{S}}^{\max}$ have all significantly worse $\Phi_{\mathcal{S}}^{\text{weighted}}$ than un-

optimized routing with uniform link costs. However, link cost settings optimized with $\Phi_{\mathcal{S}}^{\text{max,in}}$ or the combined optimization lead to the same maximum link utilization, but to significantly better $\Phi_{\mathcal{S}}^{\text{weighted}}$ values.



(a) CDFs of maximum link utilization values $\rho_{\mathcal{S}}^{\max}$.



(b) CDFs of Fortz function values $\Phi_{\mathcal{S}}^{\text{weighted}}$.

Figure 2.9: *CDFs of objective functions for all link cost settings optimized for resilient IP routing (obtained within 24 hours for the NOBEL network).*

The combined optimization achieves impressive improvements for any secondary Fortz-based objective function Ψ. The first row of Table 2.2 shows the best values of 3 different optimizations using resilient Fortz-based functions, and the second row shows best results of 3 combined optimizations, using $\rho_{\mathcal{S}}^{\max}$ together with a Fortz-based function. The columns specify which Fortz-based function was used. The lower part of Table 2.2 shows objective function values for a best result $\mathbf{k}^{\rho_{\mathcal{S}}^{\max}}$ of only minimizing the maximum link utilization $\rho_{\mathcal{S}}^{\max}$ and for routing with uniform link costs $\mathbf{k}_u$. The best $\rho_{\mathcal{S}}^{\max}$ link cost setting leads to

Table 2.2: *Best link cost settings in the NOBEL network.*

Ψ	$\rho_{\mathcal{S}}^{\max}$	$\Phi_{\mathcal{S}}^{\text{weighted}}$	$\Phi_{\mathcal{S}}^{\max,\text{out}}$	$\Phi_{\mathcal{S}}^{\max,\text{in}}$
$\mathbf{k}^{\Psi}$		**0.3823**	**0.5200**	**1.2186**
$\mathbf{k}^{\rho_{\mathcal{S}}^{\max}+\Psi}$		0.3978	0.5364	1.3299
$\mathbf{k}^{\rho_{\mathcal{S}}^{\max}}$	**0.9145**	*0.4680*	0.6341	*1.5910*
$\mathbf{k}_{\text{u}}$	1.0000	0.4213	0.6612	1.5328

Fortz-based objective values that are about as bad or even worse than those with unoptimized routing with $\mathbf{k}_{\text{u}}$. On the contrary, the combined optimizations using $\rho_{\mathcal{S}}^{\max}$ and a Fortz-based objective function can almost close the gap to the best Fortz results, while still maintaining the same best $\rho_{\mathcal{S}}^{\max}$ value of 0.9145.

2.2 Routing Optimization for IP Networks with Loop-Free Alternates

After the analysis of objective functions and routing optimization for regular IP routing under failure conditions, we shift our focus onto an additional mechanism, which provides much faster failure recovery after outages. The loop-free alternates (LFAs), as introduced in the beginning of this chapter, are calculated decentrally in every router. When the primary path towards a destination fails, a pre-determined LFA can immediately be used to reroute packets without recalculation and coordination with other routers. However, usually not all destinations can be protected by LFAs so that some failures can lead to packet loss despite available connectivity. Metrics are defined in this section that identify different types of LFA failure coverage, i.e., the potential of LFAs to protect against failures. A second drawback of LFAs is that some of them may cause extra-loops in case of unexpected node or multiple failures. An extra-loop is a forwarding loop caused by LFAs where packets loop between two or more nodes. This can even

overload links and routers that are otherwise unaffected by the failure.

There are various incentives for the use of LFAs in IP networks. We call them applications and consider several of them. We argue that the utility of available LFAs depends on the application and measure the utility by application-specific LFA coverages. Some examples:

- LFA coverage can be measured by the fraction of destinations that each node can protect by LFAs, averaged over all nodes. This is an intuitive definition that nicely reflects the availability of LFAs in a network and was used for that purpose in most existing studies on LFAs. However, it does not relate to any specific application.

- One goal of IP-FRR is to reduce traffic loss between failure detection and the completion of the rerouting process. This is reflected by the fraction of the traffic that is lost due to missing LFAs, averaged over all considered failures. We use this as an indirect measure for LFA coverage.

- Network providers can sell improved availability guarantees if traffic is protected by LFAs on its entire path so that only marginal traffic is lost in case of a failure. Thus, the LFA coverage may be quantified by the fraction of traffic for which the entire path can be protected by LFAs.

- If all flows carried over a link can be protected by LFAs, this link may fail without losing any traffic after LFA activation. As a consequence, IP rerouting may be delayed when such a link fails and graceful reconvergence techniques [93–96] can be utilized to prevent micro-loops. For short-lived link failures or maintenance operations, IP rerouting, which can lead to routing instabilities and micro-loops, may be avoided even twice: once when the link goes down and once when it comes up again. For these applications, the LFA coverage may be expressed by the fraction of links for which all traffic carried under failure-free operation can be protected by LFAs.

We further diversify the definitions of LFA coverage with regard to the types of LFAs that may be used: all LFAs or only those, which do not create extra-loops. The relevance of avoiding temporary extra-loops is certainly application-specific.

The availability of LFAs and the LFA coverage obviously depend on the network topology and the routing. Thus, LFA coverage may be increased by changing the topology: additional (physical or virtual) links may be installed which provide LFAs that can be used during failures [97, 98]. LFA coverage can also be increased through routing adjustments by configuration of appropriate administrative link costs that determine the path layout in IP networks [99, 100].

In this section, we investigate the different definitions of LFA coverage in test networks with uniform link costs. We further apply these definitions as objective functions for link costs optimization to maximize LFA coverage. With this approach we achieving significant improvements in LFA coverage. However, tweaking link costs influences not only LFA coverage but also traffic distribution within the network. We show that maximizing LFA coverage can lead to significantly increased link loads both under failure-free conditions and after rerouting in failure cases so that traffic may be lost due to overload. This is not acceptable since these phases persist longer than the short rerouting interval for which LFAs reduce packet loss. Hence, maximization of LFA coverage can be counterproductive. To fix this problem, we propose Pareto-optimization to generate a set of link costs that are Pareto-optimal with regard to LFA coverage and maximum link loads. Some of these link costs lead to relatively high LFA coverage and relatively low maximum link loads so that a network administrator can choose appropriate ones to configure the network.

The content of this section is mainly taken from our study [1]. Its remainder is structured as follows. Section 2.2.1 explains LFAs and Section 2.2.2 gives an overview of related work. Section 2.2.3 discusses various applications of LFAs that require different definitions of LFA coverage, and the potential of routing optimization is illustrated. Section 2.2.4 shows that there is a tradeoff between high LFA coverage and low link loads and suggests Pareto-optimization to find good compromises.

2.2.1 Loop-Free Alternates

LFAs provide fast protection for IP networks using link state routing protocols. They are intended to be used by a node immediately after it has detected a failure until the failure disappears or until IP rerouting has converged. In this section we review the definition of LFAs [49]. As general LFAs may cause extra-loops under some conditions, we define three sets of LFAs that avoid extra-loops to a different extent.

2.2.1.1 General or Link-Protecting LFAs

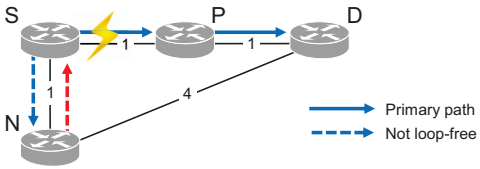


Figure 2.10: *Neighbor N cannot be used as LFA because it does not meet the loop-free condition.*

We consider a *source node* S and a *next-hop* P on a shortest path towards *destination* D, just like in Figure 2.10, but with a link cost less than 3 for the link from N to D. In this scenario, another *neighbor node* N of S can be used by S as LFA to D for the potential failure of the link $S \rightarrow P$ when the shortest path from N to D does not contain S. To avoid loops, the following loop-free condition must be met:

$$\text{dist}(N, D) < \text{dist}(N, S) + \text{dist}(S, D), \tag{2.13}$$

whereby $\text{dist}(A, B)$ denotes the least cumulative cost on a path between A and B. If link $S \rightarrow P$ fails, S detours the traffic destined to D via LFA N, and from N the deviated packets take the shortest path towards D. Figure 2.10 shows that such

an LFA does not always exist. The numbers associated with the links are the link costs taken into account for shortest path computation. When link $S \rightarrow P$ fails, packets can only be rerouted to neighbor N. However, this creates a forwarding loop since the shortest path from N to D leads over S. Therefore, N cannot be used as LFA by S to protect against the failure of link $S \rightarrow P$. As node S does not have any other neighbor, this example shows that LFAs cannot protect all traffic against single link failures. In Figure 2.11 both neighbors N_1 and N_2 of source S fulfill the loop-free condition with regard to destination D and can serve as LFAs to protect against the failure of the link $S \rightarrow P$.

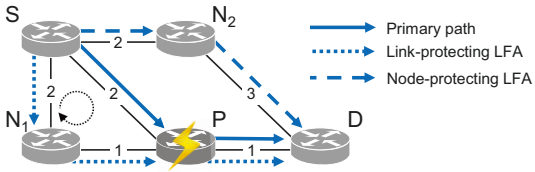


Figure 2.11: *Only the node-protecting LFA N_2 can be used to protect against the failure of node P.*

2.2.1.2 Node-Protecting LFAs

Now, we consider the failure of node P in Figure 2.11. If node S reroutes traffic to the alternate neighbor N_1, the next-hop is again P so that N_1 uses S as LFA, returns the traffic, and an extra-loop occurs. Therefore, N_1 cannot be used by S as LFA to protect against the failure of node P, but N_2 can be used for that purpose. A neighbor node N must meet the following node protection condition to protect destination D as LFA in case that node P fails:

$$\text{dist}(N, D) < \text{dist}(N, P) + \text{dist}(P, D). \tag{2.14}$$

An LFA meeting only the loop-free condition is called link-protecting while an LFA also meeting the node protection condition is called node-protecting. Since

the node protection condition implies the loop-free condition [4], every node-protecting LFA is also link-protecting, but not vice-versa.

2.2.1.3 Downstream LFAs

We consider source S and destination D in Figure 2.12. N provides a node-protecting LFA for S and vice-versa. If two nodes P_S and P_N fail simultaneously, S reroutes its traffic to N. Node N cannot forward the traffic, either, and reroutes it to S so that an extra-loop occurs. Such loops may happen during multiple failures and can be avoided if an LFA fulfills the downstream condition:

$$\text{dist}(N, D) < \text{dist}(S, D). \tag{2.15}$$

An LFA fulfilling this condition is called downstream LFA. Allowing only downstream LFAs guarantees loop avoidance for all failure cases because packets always get closer to the destination. In Figure 2.12, N is a downstream LFA for S but not vice-versa.

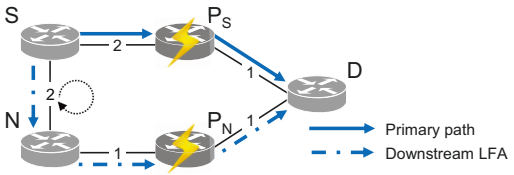


Figure 2.12: *Neighbor N is a downstream LFA of S but not vice-versa. The use of only downstream LFAs avoids loops in the presence of multiple failures.*

2.2.1.4 Use of LFAs

Loop-free alternates are pre-computed and installed in the forwarding information base (FIB) of a router. Normally, this is done for each destination so that we speak of per-prefix LFAs. If an LFA can protect all traffic for a specific next-hop,

it may be used as a per-link LFA to simplify forwarding tables. However, per-link LFAs cannot protect as much traffic as per-prefix LFAs, they cannot protect against node failures, and cause forwarding loops in case of some node failures or multiple failures. Therefore, per-prefix LFAs are the preferred mechanism [101] and we study only per-prefix LFAs.

2.2.1.5 Loop Avoidance Classes

For our analysis, we define three loop avoidance classes (LACs) of LFAs [4].

LP All link-protecting LFAs are used. They may cause extra-loops after node failures or multiple failures.

NP Only node-protecting LFAs are used to protect against failures except for the failure of the last link towards a destination, which may be protected by a link-protecting LFA. By definition, there is no node-protecting LFA for the last link. Due to the potential use of link-protecting LFAs, extra-loops may occur in case of multiple failures or when the destination node fails.

ND Only node-protecting downstream LFAs are used to protect against failures except for the failure of the last link towards a destination, which may be protected by a link-protecting downstream LFA. The selected LFAs do not cause any extra-loops.

The remainder of this section concentrates on the LP-LAC and the ND-LAC since they excel with the highest LFA coverage or avoidance of any loops, respectively.

2.2.2 Related Work on IP Fast Reroute Mechanisms

Multiple fast reroute mechanisms have been developed for IP networks [102, 103]: multiple routing configurations [104], failure insensitive routing [105], not-via addresses [106], failure-carrying packets [107], and others. They can protect the network against all single failures as long as the network topology provides alternate paths. Therefore, routing optimization in this context usually aims at

minimizing relative link loads during failure-free operation and sometimes also for likely failure scenarios. The authors of [108] minimize link utilization for failure-free conditions while taking care that link capacities suffice to accommodate the backup traffic in all single failures.

Loop-free alternates are simpler, easier to implement and deploy than the FRR methods mentioned before, and currently the only IP-FRR solution offered by vendors. However, LFAs often cannot protect all traffic against all single link failures and never against all single node failures even if the network topology provides alternative paths [50–52]. A recent IETF document [109] reports that in typical service provider access networks, all single link failures can be protected by general LFAs. It also analyzes the LFA coverage in several simulated backbone topologies. Another Internet draft [110] suggests to consider link bandwidths when selecting LFAs. The Cisco software Cariden MATE [111] illustrates and evaluates the LFA coverage. Retvari et al. [98, 100] studied the availability of LFAs from a structural point of view, formulated topological prerequisites for high LFA coverage, and provided lower and upper bounds for LFA coverage for certain network structures. All these papers have in common that they consider only general LFAs which may cause loops in case of node failures or multiple failures. In previous work [4] we formulated the three loop avoidance classes, analyzed the LAC-specific LFA coverage, and showed that it heavily depends on link costs.

Ho et al. [99] optimize link costs to maximize the average fraction of protected destinations per node and to minimize the maximum relative link load under failure-free conditions at the same time. In contrast to our work, they do not differentiate between different LFA types, they use only per-link LFAs, and they do not consider the relative link load in failure scenarios. Retvari et al. [98, 100] propose a mixed integer program and a heuristic approach to improve the LFA coverage by link cost optimization. They show that the problem is NP-complete, and recently included the protection of node failures as well as lower and upper bounds on LFA coverage in their work [112].

As it may be impossible to achieve full LFA coverage, additions and modifica-

tions to LFAs have been proposed. In [4] we considered a combination of LFAs and not-via addresses. Juniper proposes in its LFA implementation guide [97] to increase LFA coverage by adding links or tunnels, e.g., MPLS label switched paths. Also Retvari et al. [98, 100] showed that sometimes the addition of a few links significantly increases the availability of general LFAs and makes the network even fully protectable against single link failures. The authors of [113] propose E-LFAs to increase the LFA coverage, but they require protocol changes and they are more complex than normal LFAs, defeating their major advantage over other IP-FRR methods. Another modification of LFAs with the same pros and cons uses failure notifications [114]. Remote LFAs [115] have been recently proposed to extend the coverage of local LFAs. They are pre-installed tunnels and relay traffic to another node in the network from which the traffic can be forwarded to its destination. They are used in failure cases if local LFAs are not available. Like with not-via addresses, the drawback of remote LFAs is the tunneling overhead, but they do not require network-wide coordination. Csikor and Retvari showed that remote LFAs can greatly improve the LFA coverage in well-meshed networks, but they still had to add new IP links to achieve 100% LFA coverage [116].

2.2.3 Analysis and Optimization of LFA Coverage

In this section, we first present a general analysis on LFA coverage. Then we show the networks under study and briefly introduce our link cost optimization method. We introduce various applications of IP-FRR and suggest performance metrics that capture the application-specific LFA coverage. To maximize the LFA coverage, we optimize link costs using various metrics as objectives functions and compare their benefit for specific LFA applications. Our study is LFA-type-aware in the sense that we separately consider general LFAs and LFAs that do not create extra-loops in case of node and multiple failures.

2.2.3.1 General LFA Coverage Analysis

The potential of loop-free alternates to protect against network failures is heavily dependent on the routing configuration. Some structures in a network topology cannot be protected completely by LFAs, independently of the applied routing. In this section, we analyze some of these general structures.

A simple topology that can be 100% protected by LFAs is the triangle-topology. With shortest path routing based on uniform link costs, each router sends traffic directly to its neighbors. When a direct link fails, the failure-detecting router can reroute the packets to the remaining neighbor, which can forward the packets loop-free, using its direct link to the destination.

The simplest network topology that cannot be 100% protected by LFAs, is the square topology. We illustrate this on the example topology in Figure 2.13. Instead of a direct link between router 1 and router 4, we put an arbitrary network topology, which leads to an often observed structure in different topologies. If the path from router 1 to router 3 passes through router 2, this router does not have an LFA to protect against the failure of link $2 \rightarrow 3$, as the only other adjacent router 1 would loop the packets back to router 2. If, on the other hand, the path from 1 to 3 passes through router 4, then router 4 has no LFA to protect against the failure of link 3-4, because router 1 would send packets back to router 4. Thus, here it is impossible to achieve 100% fast failure protection using LFAs.

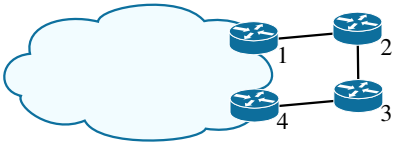


Figure 2.13: *Typical network structure that cannot be 100% protected with LFAs.*

In general, a router A with n neighbors does not have an LFA for the link $A \rightarrow B$, when all other $n - 1$ neighbors send their packets towards router B over router A. This can often not be avoided in IP networks because all paths are shortest paths according to the given link-costs metric. Any subpath of a shortest

path must itself be a shortest path. Thus, if two routings towards different destination nodes both include paths from node X to node Y, then these two paths must be identical. This is in contrast to explicit routing, e.g. in MPLS, where routing between nodes is independent from other paths. It also explains, why not only the square-topology (Figure 2.13), but all ring-like topology-parts cannot be completely LFA-protected. Each node Y with degree 2 that receives traffic from a node X and forwards it to node Z does not have an alternate when the link towards Z fails. The only other neighboring router X would loop packets back to Y.

But also in network topologies with higher node-degree, there are often network structures that cannot be 100% protected with LFAs. An example is the cube-topology, where each node has degree 3 (Figure 2.14). With uniform link costs, a router does not have alternates for packets that are destined to one of its neighbors. The failure-detecting router cannot be sure whether any of its remaining neighbors would loop back packets or forward them on a different path with equal costs. With our optimization framework, the link-costs settings can be optimized so that only one router remains with a single unprotected destination. Because of the shortest-path routing with its sub-path constraints, this cannot be improved.

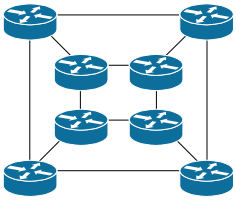


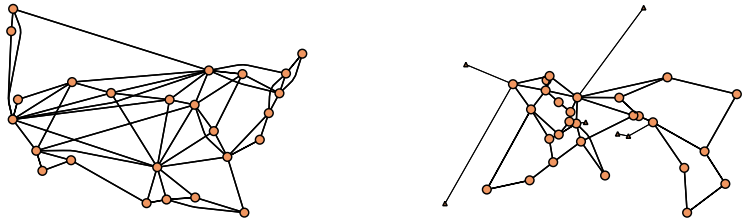
Figure 2.14: *Topology with node degree 3 that cannot be 100% LFA protected.*

When considering LFAs of the ND- or NP-class, the LFA coverage is even lower. There is at least one incoming path to each router that cannot be protected by an LFA. This can easily be explained. When node failures are also considered,

the failure-detecting router must use a downstream LFA when the last-hop link towards the destination seems to be down. This is the only way to avoid looping traffic if not the link, but the destination router is broken. For every router X, there is at least one neighboring router Y with minimum dist(Y, X), i.e., the router which has the lowest link costs towards X compared to all other neighbors of X. Router Y does not have a downstream LFA to protect against a failure of link $Y \rightarrow X$, because no other router is "closer" to the destination X as Y is already the "closest". Thus, the maximum fraction of LFA protected destinations that can be achieved when router failures are taken into consideration is as follows. With n routers in the network and $n \cdot (n-1)$ destinations to protect, the maximum LFA coverage (as described in Section 2.2.3.4) is:

$$\max\left(\pi_{\text{ND}}^{\text{dest}}\right) = \frac{n \cdot (n-1) - n}{n \cdot (n-1)} = \frac{n-2}{n-1} \tag{2.16}$$

2.2.3.2 Networks under Study



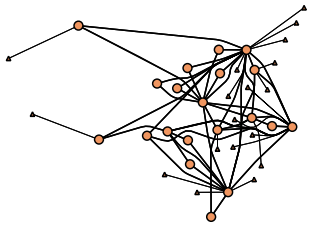
(a) AT&T Network:
25 core nodes, 112 unidirectional links.

(b) BICS Network:
27 core nodes, 84 unidirectional links.

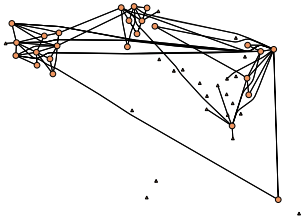
Figure 2.15: *Network topologies under study (Part I).*

For the evaluation of the following algorithms and objective functions we use several widely used research topologies from the "Topology Zoo" [117, 118] and the topology from the Nobel project [90]. They are illustrated in Figures 2.15 - 2.17. A topology is two-connected if any link or node can be removed without splitting the remaining network into several disconnected islands. As re-

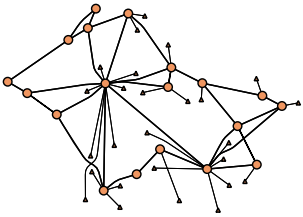
silience mechanisms require such two-connected topologies to reroute traffic, we removed nodes from the original topologies to make them two-connected in order to simplify our analysis. The removed nodes are drawn as small triangles in the figures.



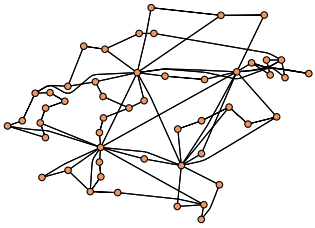
(a) China Telecom Network:
20 core nodes, 88 unidirectional links.



(b) NTT Network:
25 core nodes, 112 unidirectional links.



(c) CESNET Network:
19 core nodes, 60 unidirectional links.

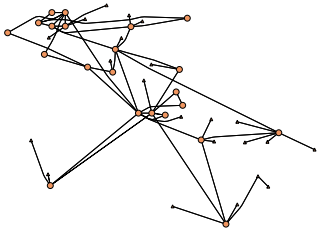


(d) DFN Network:
51 core nodes, 160 unidirectional links.

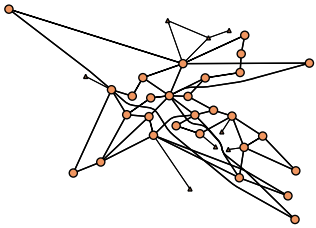
Figure 2.16: *Network topologies under study (Part II).*

Table 2.3 provides the number of nodes $|\mathcal{V}|$ and links $|\mathcal{E}|$ in our investigated networks as well as the number of (access) nodes $|\mathcal{V}_A|$ removed from the original topologies to make them two-connected. All networks in our study have homogeneous link capacities except for the Rediris network for which real link capacites are provided in [118]. Table 2.3 also indicates the maximum and average node degree which is the number of neighbors of a node. We constructed traffic matrices according to [119] which is a simplification of the method in [120].

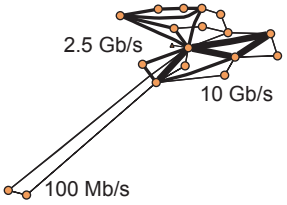
In the following experiments, we assume that a failure affects links in both directions and we use single-shortest-path routing instead of equal-cost-multipath (ECMP), as ECMP routing can lead to several problems, especially with IP fast reroute [20, 92]. Therefore, we scale our artificially generated traffic matrices such that the relative link load $\rho^{\max}_{\mathcal{S},\mathcal{E}}(\mathbf{k}_u)$ reaches 100% when uniform link costs $\mathbf{k}_u$ are used under single-shortest path routing.



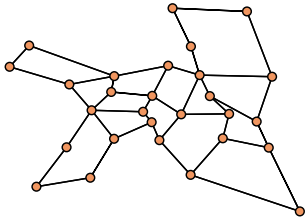
(a) GARR Network:
22 core nodes, 72 unidirectional links.



(b) GEANT Network:
29 core nodes, 94 unidirectional links.



(c) RedIris Network:
18 core nodes, 60 unidirectional links.



(d) Nobel Network:
28 core nodes, 82 unidirectional links.

Figure 2.17: *Network topologies under study (Part III).*

2.2.3.3 Link Cost Optimization

Throughout this study, we use the "Threshold Accepting" heuristic presented in Section 2.1.3 to optimize link costs for a given objective function. While we have

Table 2.3: *Networks under study.*

Network name and date		Size $\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{V}_A\rvert$	Degree d avg.	max.	Geo location
Commercial Network Topologies from Topology Zoo [118].							
AT&T	2007-2008	25	112	0	4.48	10	US
BICS	2011/01	27	84	6	3.11	7	EU
China Telecom	2010/08	20	88	18	4.40	14	CH
NTT	2011/03	25	112	22	4.48	11	Global
Research and Education Network Topologies from Topology Zoo [118].							
CESNET	2010/06	19	60	26	3.16	8	CZ
DFN	2011/01	51	160	0	3.14	12	DE
GARR	2010/12	22	72	22	3.27	8	IT
GEANT	2010/08	29	94	8	3.24	9	EU
RedIris	2011/03	18	60	1	3.33	10	ES
Topology from EU-Project NOBEL [90].							
Nobel	2005/10	28	82	0	2.93	5	EU

used the relative link load $\rho^{\max}$ as objective function in previous work, we study new objective functions in this section to quantify the LFA coverage. To support these new objective functions, we extend the heuristic so that it first analyzes the availability of different LFA types in every node of a network. Then, objective functions are calculated on this basis. We denote the new objective functions π_X^Y whereby Y indicates the specific variant and $X \in \{\text{LP}, \text{ND}\}$ indicates whether all LFAs (general LFAs, LP-LAC) or only those avoiding extra-loops (ND-LAC) are considered for protection. The objective functions are used for optimization of link costs and the resulting optimized link costs are denoted by $\mathbf{k}_X^Y$.

In Section 2.1.3.2 we extended the general optimization algorithm of [16] so

that it can optimize for a primary and secondary objective function. If not mentioned differently, we use the LFA coverage defined in the next sections as primary objective function and the maximum relative link load $\rho^{\max}$ as secondary objective function. In Section 2.2.4 we go into details of the optimization algorithm to extend it towards Pareto-optimization.

2.2.3.4 Use of LFAs to Protect Destinations

In all previous works, the fraction of protected destinations in a node, averaged over all nodes of a network (π^{dest}), has been used to quantify the LFA coverage. Moreover, only general LFAs have been taken into account ($\pi^{\text{dest}}_{\text{LP}}$) for which extra-loops can occur under some conditions. Therefore, link costs $\mathbf{k}^{\text{dest}}_{\text{LP}}$ optimized according to objective function $\pi^{\text{dest}}_{\text{LP}}$ are denoted as conventionally optimized link costs. Both uniform link costs $\mathbf{k}_{\text{u}}$ and conventionally optimized link costs $\mathbf{k}^{\text{dest}}_{\text{LP}}$ constitute the baseline for our performance comparison.

Percentage of protected destinations with LP-LAC Table 2.4 reveals that general LFAs protect between 61.2% and 98.5% of the destinations in the networks under study when uniform link costs $\mathbf{k}_{\text{u}}$ are configured. The conventionally optimized link costs $\mathbf{k}^{\text{dest}}_{\text{LP}}$ increase this range to values between 89.6% and 100%. These results confirm the findings from [98–100]: link cost optimization can tremendously increase the LFA coverage compared to uniform link costs in many networks and the achievable results depend on the network structure. In some networks (NTT, China Telecom, AT&T) even all destinations can be protected by general LFAs if conventionally optimized link costs $\mathbf{k}^{\text{dest}}_{\text{LP}}$ are used.

The Nobel network deserves special attention as it yields the least LFA coverage. Its topology does not contain any triangles, i.e., there are no two neighoring routers which have an alternate route that contains exactly one other router. As a consequence, all LFAs available with uniform link costs $\mathbf{k}_{\text{u}}$ are node-protecting [98]. Therefore, the LFA coverage is 61.2%, no matter whether LFAs of the LP-LAC or only those of the NP-LAC are used for protection. The latter is not shown in the tables.

Table 2.4: $\pi_{\text{LP}}^{\text{dest}}$: *percentage of destinations protected by general LFAs.*

Network	$\mathbf{k}_{\text{u}}$	$\mathbf{k}_{\text{LP}}^{\text{dest}}$	$\mathbf{k}_{\text{ND}}^{\text{dest}}$
AT&T	98.50	100.00	94.50
BICS	72.65	90.88	77.78
China Telecom	95.79	100.00	99.47
NTT	95.33	100.00	98.33
CESNET	87.43	98.25	83.33
DFN	72.08	93.10	76.86
GARR	74.89	98.27	87.01
GEANT	76.11	95.44	81.03
RedIris	88.24	98.69	85.62
Nobel	61.24	89.55	77.25

Percentage of protected destinations with ND-LAC General LFAs may cause extra-loops in case of some node or multiple failures. Hence, they can worsen a failure situation instead of improving it. This is avoided if only LFAs of the ND-LAC are used for protection. We now investigate the fraction of destinations protected with LFAs of the ND-LAC ($\pi_{\text{ND}}^{\text{dest}}$). The results in Table 2.5 show that LFAs of the ND-LAC protect only between 13.7% and 51.1% of all destinations in networks with uniform link costs $\mathbf{k}_{\text{u}}$ and between 36.8% and 68.4% in networks with conventionally optimized link costs $\mathbf{k}_{\text{LP}}^{\text{dest}}$. This is due to the fact that the metric $\pi_{\text{ND}}^{\text{dest}}$ reduces the set of eligible LFAs compared to $\pi_{\text{LP}}^{\text{dest}}$ so that the LFA coverage is consistently lower or equal to the corresponding values in Table 2.4. Table 2.5 also shows that LFAs of the ND-LAC with optimized link costs $\mathbf{k}_{\text{ND}}^{\text{dest}}$ can protect between 51.9% and 94.5% of the destinations, which is a significant improvement compared to uniform link costs $\mathbf{k}_{\text{u}}$ or conventionally optimized link costs $\mathbf{k}_{\text{LP}}^{\text{dest}}$. Thus, a large fraction of destinations can be protected by LFAs while avoiding extra-loops, but using the appropriate objective function for optimization is a prerequisite. Again, the achievable LFA coverage highly depends on the network structure, and can never reach 100% (see Equation 2.16).

Table 2.5: π_{ND}^{dest}: *percentage of destinations protected only by LFAs that avoid extra-loops.*

Network	$\mathbf{k}_u$	$\mathbf{k}_{LP}^{dest}$	$\mathbf{k}_{ND}^{dest}$
AT&T	34.67	65.50	91.83
BICS	23.79	44.59	61.54
China Telecom	51.05	68.42	94.21
NTT	41.83	67.50	94.50
CESNET	13.74	36.84	66.67
DFN	27.06	42.90	57.49
GARR	36.80	49.57	71.43
GEANT	23.65	49.01	67.86
RedIris	29.08	46.41	75.82
Nobel	29.23	43.25	51.85

For application in practice, it might be worthwhile to maximize the fraction of destinations protectable by LFAs that do not create extra-loops under any conditions and extend this LFA coverage by general LFAs where LFAs of the ND-LAC are not available. Table 2.4 shows that in the considered networks between 77.3% and 99.5% of the destinations can be protected. The comparison of the values $\mathbf{k}_{ND}^{dest}$ and $\mathbf{k}_{LP}^{dest}$ makes a tradeoff evident: minimizing extra-loops reduces also the percentage of protected destinations; the extent of this reduction depends on the network structure.

2.2.3.5 Use of LFAs to Reduce Traffic Loss

The major reason for using LFAs is the reduction of traffic loss from the detection of a failure until the completion of the IP rerouting process. To quantify the LFA coverage for this purpose, the fraction of protected destinations is not appropriate. We now consider the fraction of protected traffic to quantify the LFA coverage. However, this metric yields numbers close to 100% which are rather cumbersome to compare. Therefore, we take the fraction of unprotected traffic as

metric instead, which can be interpreted as traffic loss in failure cases, and so we denote it as π^{loss}. We compute it as follows. For each link failure we calculate the fraction of traffic which is affected by the failure and not protected by an LFA, and average these values over all link failures. We take only single (bidirectional) link failures into account as we assume that their probability is two orders of magnitude larger than the one of node failures or multiple failures [121].

In contrast to the fraction of protected destinations π^{dest}, the unprotected traffic π^{loss} accounts for heterogeneous traffic matrices and for the amount of traffic forwarded by each node. Therefore, the calculation of the traffic loss π^{loss} requires the knowledge of the traffic matrix, which should be sufficiently stable to make the proposed metric meaningful. If the traffic matrix is not known for a network, we create a traffic matrix as described in Section 2.2.3.2.

Table 2.6: $\pi^{\text{loss}}_{\text{LP}}$: *percentage of lost traffic when using general LFAs.*

Network	$\mathbf{k}_{\text{u}}$	$\mathbf{k}^{\text{dest}}_{\text{LP}}$	$\mathbf{k}^{\text{loss}}_{\text{LP}}$	$\mathbf{k}^{\text{loss}}_{\text{ND}}$
AT&T	0.02	0.00	0.00	0.01
BICS	1.69	0.55	0.33	1.01
China Telecom	0.12	0.00	0.00	0.00
NTT	0.32	0.00	0.00	0.01
CESNET	1.76	0.47	0.08	0.52
DFN	1.18	0.83	0.38	0.70
GARR	2.13	0.34	0.09	0.44
GEANT	1.66	0.42	0.13	0.66
RedIris	0.46	0.18	0.10	0.15
Nobel	3.75	1.31	0.62	1.30

Percentage of lost traffic with LP-LAC Table 2.6 reports the traffic loss in failure cases when general LFAs are installed. The percentages vary between 0.02% and 3.75% for uniform link costs $\mathbf{k}_{\text{u}}$. Conventionally optimized link costs

$\mathbf{k}_{\text{LP}}^{\text{dest}}$ reduce these values to a range between 0% and 1.31%. The improvement depends a lot on the network structure. The largest improvement is achieved in the GARR network where the unprotected traffic is reduced from 2.13% to 0.34%. When optimizing the link costs to minimize the fraction of unprotected traffic ($\mathbf{k}_{\text{LP}}^{\text{loss}}$), the percentages of unprotected traffic lie in the range between 0% and 0.62% and are clearly lower than those for conventionally optimized link costs $\mathbf{k}_{\text{LP}}^{\text{dest}}$.

Thus, the new objective function $\pi_{\text{LP}}^{\text{loss}}$ leads to superior optimized link costs because LFAs are preferably available in nodes that forward lots of traffic and for destinations to which lots of traffic is forwarded. This is different for the other link costs $\mathbf{k}_{\text{u}}$ and $\mathbf{k}_{\text{LP}}^{\text{dest}}$ which are either not optimized or optimized without the information of the traffic matrix. These results underline that the selection of an appropriate objective function has a great influence on the optimization and the quality of the resulting routing.

Table 2.7: $\pi_{\text{ND}}^{\text{loss}}$: *percentage of lost traffic when using only LFAs that avoid extra-loops.*

Network	$\mathbf{k}_{\text{u}}$	$\mathbf{k}_{\text{LP}}^{\text{dest}}$	$\mathbf{k}_{\text{LP}}^{\text{loss}}$	$\mathbf{k}_{\text{ND}}^{\text{loss}}$
AT&T	2.31	1.47	0.49	0.14
BICS	4.62	4.56	4.06	2.11
China Telecom	2.48	1.25	0.88	0.07
NTT	2.20	1.19	1.15	0.11
CESNET	6.43	5.57	4.60	1.69
DFN	2.96	2.68	2.55	1.74
GARR	4.25	3.90	3.78	1.23
GEANT	4.46	4.22	4.72	1.61
RedIris	4.85	4.04	3.23	0.75
Nobel	5.11	4.58	4.86	2.71

Percentage of lost traffic with ND-LAC We now allow only LFAs of the ND-LAC to avoid potential extra-loops. According to Table 2.7 the percentage of unprotected traffic is in a range between 2.20% and 6.43% for uniform link costs $\mathbf{k}_{\mathrm{u}}$, in a range between 1.19% and 5.57% for conventionally optimized link costs $\mathbf{k}_{\mathrm{LP}}^{\mathrm{dest}}$, and in a range between 0.49% and 4.86% for $\mathbf{k}_{\mathrm{LP}}^{\mathrm{loss}}$. These values are all rather high. Appropriate optimization takes into account that only LFAs of the ND-LAC can be used; correspondingly optimized link costs $\mathbf{k}_{\mathrm{ND}}^{\mathrm{loss}}$ can reduce the percentage of unprotected traffic to a range between 0.07% and 2.71%, which is a significant improvement. Thus, conventionally optimized link costs $\mathbf{k}_{\mathrm{LP}}^{\mathrm{dest}}$ are not universal enough to sufficiently well approximate the quality of appropriately optimized link costs $\mathbf{k}_{\mathrm{ND}}^{\mathrm{loss}}$.

When using link costs $\mathbf{k}_{\mathrm{ND}}^{\mathrm{loss}}$, LFAs of the ND-LAC may be primarily used and complemented by general LFAs to minimize both the risk of extra-loops and traffic loss. Table 2.6 shows that this variant leaves about the same amount of traffic unprotected as conventionally optimized link costs $\mathbf{k}_{\mathrm{LP}}^{\mathrm{dest}}$; however, the risk of extra-loops is clearly reduced because mostly LFAs of the ND-LAC are taken.

Traffic loss distribution for general LFAs The observed percentages of unprotected traffic are average values and seem small. Their real implication becomes clear in Figure 2.18. It depicts the number of single link failures that cause more traffic loss than a certain percentage x due to missing LFAs; the evaluation is performed for the Nobel network with general LFAs.

With uniform link costs $\mathbf{k}_{\mathrm{u}}$, 3.75% of the traffic is lost on average due to missing LFAs during single link failures. In 31 out of 41 link failure scenarios the traffic loss is lower than 5%, but 10 link failures cause more than 5% traffic loss. The failure of the link from Rome to Athens generates even 17.6% traffic loss which is quite a lot.

Conventionally optimized link costs $\mathbf{k}_{\mathrm{LP}}^{\mathrm{dest}}$ cause a smaller average traffic loss of 1.31%. In 19 out of 41 link failure scenarios even all traffic can be protected by LFAs. Only two link failures cause more than 5% traffic loss and the largest traffic loss is 8% for the failure of the link from Brussels to Frankfurt.

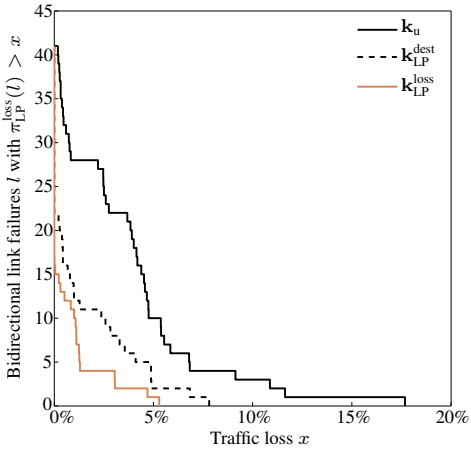


Figure 2.18: *Distribution of traffic loss for all single bidirectional link failures in the Nobel network when general LFAs are used.*

Link costs optimized to minimize the unprotected traffic $\mathbf{k}_{LP}^{loss}$ lead to even better results. In 25 out of 41 link failure scenarios all traffic can be protected by LFAs. Only the failure of the link from Athens to Belgrade exceeds the value of 5% and generates 5.5% traffic loss.

This evaluation exhibits that link costs optimized to reduce traffic loss are advantageous compared to uniform link costs $\mathbf{k}_u$ or conventionally optimized link costs $\mathbf{k}_{LP}^{dest}$. They increase the number of link failure scenarios in which all traffic can be protected and clearly decrease the number of link failure scenarios in which a large fraction of more than 5% of the overall traffic is lost. Thus, link cost optimization can have large effects for particular failure scenarios.

2.2.3.6 Use of LFAs to Increase the Availability of Entire Paths

If all the links of a path from an ingress to an egress node are protected with LFAs, the overall availability of this path is tremendously improved: whatever link fails,

the time to repair will be very short. On such paths, an ISP can provide a high-availability service to its customers. In the following, we evaluate the fraction of traffic whose paths can be fully protected by LFAs. We denote this metric as fully protected traffic π^{full} and link costs optimized according to this metric are denoted as $\mathbf{k}^{\text{full}}$.

Table 2.8: $\pi_{\text{LP}}^{\text{full}}$: *percentage of traffic fully protected by general LFAs.*

Network	$\mathbf{k}_{\text{u}}$	$\mathbf{k}_{\text{LP}}^{\text{dest}}$	$\mathbf{k}_{\text{LP}}^{\text{full}}$	$\mathbf{k}_{\text{ND}}^{\text{full}}$
AT&T	98.76	100.00	100.00	99.29
BICS	48.32	81.85	89.05	70.19
China Telecom	94.40	100.00	100.00	100.00
NTT	81.60	100.00	100.00	100.00
CESNET	46.21	85.29	97.49	83.53
DFN	30.83	44.88	73.82	56.03
GARR	30.24	87.27	96.78	80.27
GEANT	34.82	80.74	95.20	66.41
RedIris	86.41	94.53	97.07	92.33
Nobel	0.00	47.64	79.22	52.14

Percentage of fully protected traffic with LP-LAC Table 2.8 indicates the percentage of fully protected traffic. With uniform link costs $\mathbf{k}_{\text{u}}$ between 30.2% and 98.8% of the traffic can be fully protected by general LFAs. The Nobel network is an exception since not a single flow can be fully protected. Due to the absence of triangles and the use of uniform link costs $\mathbf{k}_{\text{u}}$, only node-protecting LFAs exist which cannot protect the last link of a path.

Conventionally optimized link costs $\mathbf{k}_{\text{LP}}^{\text{dest}}$ increase the values $\pi_{\text{LP}}^{\text{full}}$ to a range between 44.9% and 100%, and lead to 47.6% in the Nobel network. With link costs optimized for fully protected traffic $\mathbf{k}_{\text{LP}}^{\text{full}}$ between 73.8% and 100% of the traffic can be protected and even 79.2% in the Nobel network. Hence, the appro-

priately optimized link costs $\mathbf{k}_{\text{LP}}^{\text{full}}$ significantly increase the percentage of fully protected traffic $\pi_{\text{LP}}^{\text{full}}$ compared to $\mathbf{k}_{\text{LP}}^{\text{dest}}$.

Table 2.9: $\pi_{\text{ND}}^{\text{full}}$: *percentage of traffic fully protected only by LFAs that avoid extra-loops.*

Network	$\mathbf{k}_{\text{u}}$	$\mathbf{k}_{\text{LP}}^{\text{dest}}$	$\mathbf{k}_{\text{LP}}^{\text{full}}$	$\mathbf{k}_{\text{ND}}^{\text{full}}$
AT&T	0.00	31.64	74.18	93.35
BICS	0.00	8.17	17.69	56.17
China Telecom	0.00	43.90	70.45	97.24
NTT	0.00	37.28	54.78	95.02
CESNET	0.00	7.17	22.31	69.67
DFN	0.00	4.07	7.71	28.82
GARR	0.00	3.62	17.07	75.18
GEANT	0.00	3.36	6.57	57.36
RedIris	0.00	12.63	29.39	83.86
Nobel	0.00	0.74	7.87	37.39

Percentage of fully protected traffic with ND-LAC Table 2.9 extends this study towards the exclusive use of LFAs that avoid extra-loops in any failure scenario. With uniform link costs $\mathbf{k}_{\text{u}}$, not a single flow can be protected under these conditions in any network. This is caused by the same effect as in Section 2.2.3.1: The last hop towards a destination has distance 1 to this destination so that no other neighbor is closer to this destination. Therefore, it is impossible to find downstream LFAs to protect the failure of the last hop. So, not a single path can be fully protected by LFAs of the ND-LAC with uniform link costs $\mathbf{k}_{\text{u}}$.

Conventionally optimized link costs $\mathbf{k}_{\text{LP}}^{\text{dest}}$ fully protect between 3.4% and 43.9% of the traffic and only 0.7% in the Nobel network. Link costs optimized for fully protected traffic $\mathbf{k}_{\text{LP}}^{\text{full}}$ with use of general LFAs fully protect between 6.6% and 74.2% of the traffic, but appropriately optimized link costs $\mathbf{k}_{\text{ND}}^{\text{full}}$ fully

protect between 28.8% and 97.2% of the traffic. Again, using the appropriate metric for routing optimization is crucial as approximations by similar metrics yield significantly worse results.

Table 2.8 demonstrates that complementing the coverage of LFAs of the ND-LAC with LFAs of the LP-LAC for $\mathbf{k}_{\text{ND}}^{\text{full}}$ significantly increases the fractions of fully protected traffic to a range between 52.1% and 100%. The percentage of fully protected traffic is then similar to the one of conventionally optimized link costs $\mathbf{k}_{LP}^{\text{dest}}$ but most potential extra-loops are avoided. However, $\mathbf{k}_{\text{ND}}^{\text{full}}$ leads to clearly less fully protected traffic than $\mathbf{k}_{\text{LP}}^{\text{full}}$ when general LFAs can be used.

2.2.3.7 Use of LFAs to Preferably Protect Traffic with High-Availability Requirements

Only some networks allow to avoid traffic loss completely or to fully protect all traffic with LFAs. Therefore, it seems reasonable to preferably protect traffic with high-availability requirements in those networks. This approach can be considered as a form of differentiated resilience [122–125]. For that purpose, we propose an extension of objective functions for link cost optimization and demonstrate its effectiveness in a challenging experiment.

Extension of objective functions for routing optimization with preferred protection of high-priority traffic We assume that traffic of some ingress-egress pairs $d \in \mathcal{D}_h$ has high-availability requirements and that all other traffic has low-availability requirements. We call these traffic classes high- and low-priority traffic. Our goal is to preferably protect high-priority traffic. To prioritize high-priority traffic for the purpose of optimization, we modify the original traffic matrix by adding a priority offset $r_{\text{offset}}^{\text{prio}}$ to the rates of high-priority demands:

$$r_{\text{modified}}(d) = \begin{cases} r(d) & d \in \mathcal{D} \setminus \mathcal{D}_h \\ r(d) + r_{\text{offset}}^{\text{prio}}(d) & d \in \mathcal{D}_h \end{cases} . \tag{2.17}$$

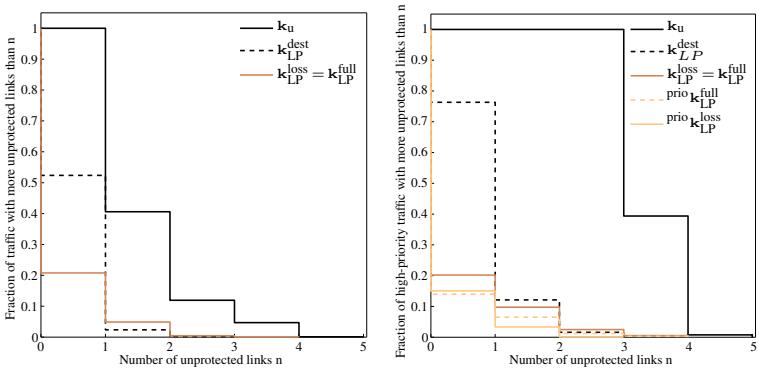
We use the overall traffic rate in the network $D_\Sigma = \sum_{d \in \mathcal{D}} r(d)$ to define the priority offset as

$$r^{\text{prio}}_{\text{offset}}(d) = D_\Sigma \cdot \frac{r(d)}{\min_{\{\delta \in \mathcal{D}_h\}} (r(\delta))}, d \in \mathcal{D}_h. \tag{2.18}$$

This definition makes the demand-specific priority offsets $r^{\text{prio}}_{\text{offset}}(d)$ proportional to the original traffic rates $r(d)$ and ensures that the priority offset for the high-priority demand with the smallest rate equals the overall traffic rate D_Σ. Thereby, $r_{\text{modified}}(d)$ of the smallest high-priority traffic aggregate is larger than the sum of modified rates of all other low-priority traffic aggregates. As a consequence, the smallest high-priority traffic aggregate will be more respected in optimizations than any other low-priority traffic aggregate.

Variations for the modification of the traffic matrix, e.g., scalar multiplications or zeroing the rates of low-priority aggregates, are possible. Extensions based on modifications of the traffic matrix can be successfully applied only to traffic-aware objective functions such as π^{loss} proposed in Section 2.2.3.5 or π^{full} proposed in Section 2.2.3.6. It cannot be applied to the conventional objective function π^{dest} as this is not aware of any traffic demands.

Evaluation In Figure 2.19(a) we report the fraction of traffic in the Nobel network for which more than n links cannot be protected by general LFAs. These values depend on the link costs used in the network. With uniform link costs $\mathbf{k}_{\text{u}}$ any traffic is affected by the failure of at least one link. This is in accordance with the results presented in Table 2.8. About 12% of the traffic cannot be protected against the failure of 3 or 4 links on its path. Such traffic is quite vulnerable. Conventionally optimized link costs $\mathbf{k}^{\text{dest}}_{\text{LP}}$ clearly reduce the fraction of traffic affected by various numbers of link failures. Optimized link costs $\mathbf{k}^{\text{full}}_{\text{LP}}$ minimize the fraction of traffic affected by one or more link failures. In this particular experiment, these link costs also minimize the traffic loss so that $\mathbf{k}^{\text{full}}_{\text{LP}}$ equals $\mathbf{k}^{\text{loss}}_{\text{LP}}$. Note that these link costs lead to a larger fraction of traffic that is affected by at least two link failures compared to conventionally optimized link costs $\mathbf{k}^{\text{dest}}_{\text{LP}}$.



(a) with regard to the overall traffic.

(b) with regard to high-priority traffic.

Figure 2.19: *Distribution of the number of unprotected links.*

With uniform link costs $\mathbf{k}_u$, 12% of the traffic cannot be protected on 3 or 4 links. As a challenge, we define it as high-priority traffic. We preferably protect this traffic with general LFAs using the metrics π_{LP}^{loss} and π_{LP}^{full} combined with the extension presented above. The optimization returns the optimized link costs $^{prio}\mathbf{k}_{LP}^{loss}$ and $^{prio}\mathbf{k}_{LP}^{full}$. Figure 2.19(b) displays the percentage of high-priority traffic for which more than n links cannot be protected by LFAs. With uniform link costs, 3 or more links cannot be protected for 100% of this traffic which is in line with the design of the experiment. Conventionally optimized link costs $\mathbf{k}_{LP}^{dest}$ reduce the number of links on which the high-priority traffic cannot be protected, but 77% of the high-priority traffic still misses LFA protection on one or more links. Link costs optimized to minimize traffic loss $\mathbf{k}_{LP}^{loss}$ (as well as $\mathbf{k}_{LP}^{full}$) reduce this percentage to 20%. Our proposed extensions decrease the value further down to about 15% and they also reduce the fraction of high-priority traffic that cannot be protected on more than 1 or 2 links. This shows that it is possible to improve the protection of a preferred subset of high-priority traffic.

2.2.3.8 Use of LFAs to Fully Protect Link Failures

As outlined before, advanced applications of IP-FRR delay the normal rerouting process. This can be done with losing hardly any traffic only if all traffic affected by a link failure is protected by LFAs. Therefore, the advanced applications can be performed only for links for which all carried traffic is protected by LFAs. We define a link as fully protected if all traffic affected by its bidirectional failure can be fully protected by LFAs. Furthermore, we define π^{link} as the fraction of fully protected links which should be maximized. The link costs optimized by this metric are denoted as $\mathbf{k}^{\text{link}}$.

Percentage of fully protected links with LP-LAC Table 2.10 gives the fraction of links fully protected with general LFAs. The percentages for uniform link costs $\mathbf{k}_{\text{u}}$ vary between 35.0% and 96.4%. The Nobel network is again an exception as not a single link can be fully protected with LFAs under uniform link costs $\mathbf{k}_{\text{u}}$. With uniform link costs $\mathbf{k}_{\text{u}}$, any link is a last link of the paths towards this link's destination, and cannot be protected since all available LFAs are node-protecting in the Nobel network due to the absence of triangles. Conventionally optimized link costs $\mathbf{k}_{\text{LP}}^{\text{dest}}$ clearly increase the fraction of fully protected links to a range between 46.3% and 100%. Using $\pi_{\text{LP}}^{\text{link}}$ as objective function for optimization, the fractions of fully protected links can be even further increased in most cases; we observe the major effect of appropriate optimization in the Nobel network with 70.7% fully protected links compared to 46.3% for conventionally optimized link costs $\mathbf{k}_{\text{LP}}^{\text{dest}}$. Our results manifest that LFAs can be applied in some networks to fully protect all links. This allows for delayed IP rerouting and enables advanced applications. However, in most networks, only a subset of all links are fully protected so that IP rerouting can be delayed only for those links whose traffic is fully protected by LFAs. As a consequence, advanced applications may be performed only on a link-specific basis. This introduces complexity and lowers the benefit which rather questions the use of LFAs to enable advanced applications in such networks.

Table 2.10: $\pi_{\mathrm{LP}}^{\mathrm{link}}$: *percentage of links fully protected by general LFAs.*

Network	$\mathbf{k}_{\mathrm{u}}$	$\mathbf{k}_{\mathrm{LP}}^{\mathrm{dest}}$	$\mathbf{k}_{\mathrm{LP}}^{\mathrm{link}}$	$\mathbf{k}_{\mathrm{ND}}^{\mathrm{link}}$
AT&T	96.43	100.00	100.00	92.86
BICS	42.86	73.81	78.57	54.76
China Telecom	90.91	100.00	100.00	77.27
NTT	82.14	100.00	100.00	87.50
CESNET	66.67	83.33	86.67	56.67
DFN	35.00	53.75	58.75	40.00
GARR	41.67	77.78	86.11	61.11
GEANT	48.94	78.72	85.11	61.70
RedIris	66.67	86.67	86.67	66.67
Nobel	0.00	46.34	70.73	41.46

Percentage of fully protected links with ND-LAC When delaying normal rerouting in IP networks, it may be crucial to avoid extra-loops caused by fast rerouting mechanisms as they persist until rerouting has completed. Therefore, avoidance of extra-loops seems important in this context. Table 2.11 compiles the percentages of links that can be fully protected by LFAs of the ND-LAC. With uniform link costs $\mathbf{k}_{\mathrm{u}}$ not a single link can be fully protected. Due to the uniform link costs $\mathbf{k}_{\mathrm{u}}$, any link is a last link on the path to its destination, and appropriate downstream LFAs cannot exist to protect such links. With conventionally optimized link costs $\mathbf{k}_{\mathrm{LP}}^{\mathrm{dest}}$ at least a small percentage of links – at most 59.0%, mostly clearly less – can be fully protected without causing extra-loops. Link costs maximizing the fraction of links fully protected by general LFAs $\mathbf{k}_{\mathrm{LP}}^{\mathrm{link}}$ fully protect an even lower fraction of links – at most 29.6%. This looks surprising, but they were not optimized for the use of ND-LAC LFAs. Link costs optimized to maximize the percentage of links fully protected by ND-LAC LFAs $\mathbf{k}_{\mathrm{ND}}^{\mathrm{link}}$ yield clearly better results than $\mathbf{k}_{\mathrm{LP}}^{\mathrm{dest}}$ and $\mathbf{k}_{\mathrm{LP}}^{\mathrm{link}}$ in the range between 41.5% and 73.2%. The DFN

network is an exception with only 20% fully protected links. Although appropriate routing optimization can significantly increase the fraction of fully protected links, still a good subset of links cannot be fully protected. This observation holds for any investigated network.

Table 2.11: $\pi_{\text{ND}}^{\text{link}}$: *percentage of links fully protected only by LFAs that avoid extra-loops.*

Network	$\mathbf{k}_{\text{u}}$	$\mathbf{k}_{\text{LP}}^{\text{dest}}$	$\mathbf{k}_{\text{LP}}^{\text{link}}$	$\mathbf{k}_{\text{ND}}^{\text{link}}$
AT&T	0.00	58.93	21.43	73.21
BICS	0.00	19.05	9.52	47.62
China Telecom	0.00	52.27	29.55	68.18
NTT	0.00	58.93	21.43	71.43
CESNET	0.00	30.00	0.00	53.33
DFN	0.00	15.00	0.00	20.00
GARR	0.00	16.67	2.78	52.78
GEANT	0.00	23.40	2.13	51.06
RedIris	0.00	20.00	10.00	60.00
Nobel	0.00	2.44	4.88	41.46

2.2.4 Keeping Link Loads under Control

In the previous section we have optimized link costs to maximize the LFA coverage using different metrics, e.g. π^{dest}, π^{loss}, π^{full}, or π^{link}. In addition to the reported results we have observed that optimized link costs sometimes lead to high relative link loads although relative link loads were minimized by the optimizer's secondary objective function. This problem has not been pointed out in literature before.

In the following, we first define a link load metric that is suitable in the context of resilient networks using LFAs. We propose an extension to our link cost opti-

mization algorithm to find Pareto-optimal link costs. Performance results suggest that optimality in LFA coverage and in link load seem to be contradicting goals. Nevertheless, we demonstrate that it is possible to choose Pareto-optimal link costs leading to good LFA coverage and to moderate relative link loads.

2.2.4.1 Definition of Relative Link Load

The load of a link l can be determined by the sum of the rates $r(d)$ of all demands d that are forwarded over link l. In particular, we consider in this section the load $\rho(l)$ of a link relative to its capacity $\mathbf{c}(l)$.

We observe three different link load stages in IP networks with regard to failure scenarios:

1. Link load under failure-free operation,
2. Link load with traffic rerouted by LFAs before rerouting,
3. Link load after IP routing has reconverged to failure state.

Thereby, we neglect stages during the rerouting process that may temporarily increase some link load values. The definition of the relative link load should cover all relevant stages that persist for a sufficiently long time.

We assume now that LFAs are used to reduce the lost traffic until rerouting has completed. Here, link load stage (1) and (3) are persistent so that their maximum values should be respected for evaluations. In contrast, stage (2) is negligible as it lasts only in the order of a second. Furthermore, we consider node failures and multiple link failures clearly less likely than single (bidirectional) link failures. Thus, we can use $\rho_{\mathcal{S}}^{\max}$ (or simplified $\rho^{\max}$) as defined in Equation 2.1.2.3 as the performance metric of interest. As scenarios $\mathcal{S}$, we use failure-free conditions and single bidirectional link failures. Note that this metric is independent of the kind of LFAs that are used for protection because the definition of $\rho^{\max}$ does not include the fast reroute stage. As we do not assume the persistent use of LFAs, we can afford temporary extra-loops through LFAs for rare failure events. Therefore, we choose the use of general LFAs for our experiments.

Traffic loss happens if the load on a link is larger than 100% and reduces the traffic rate seen by a next-hop. However, we do not work with original traffic matrices but with traffic matrices that are scaled such that the maximum link load $\rho^{\max}$ is 100% for uniform link costs $\mathbf{k}_{\mathrm{u}}$. As the scaling of our traffic matrices is artificial anyway, we do not take into account that traffic is lost on links with more than 100% load, which may happen for other than uniform link costs $\mathbf{k}_{\mathrm{u}}$. This approach is justified as we are only interested in the ability of different link costs to equally distribute the load from a relative traffic matrix through the network.

2.2.4.2 Pareto-Optimization of Link Costs

An element of a set is Pareto-optimal with regard to several metrics if no other element of this set is better with regard to all considered metrics. We are interested in finding a set of optimized link costs that are Pareto-optimal with regard to the fraction of lost traffic $\pi_{\mathrm{LP}}^{\mathrm{loss}}$ due to missing LFAs and relative link load $\rho^{\max}$. To achieve this, we briefly review the principle of our optimization heuristic [16] and extend it for Pareto-optimization.

Link cost optimization using threshold accepting Threshold accepting randomly steps through the solution space of all link costs and searches for the link cost vector that optimizes the objective function f. The algorithm works with a current link cost vector $\mathbf{k}$ and records the best link cost vector $\mathbf{k}_{\mathrm{best}}$ ever found. It explores the solution space by randomly choosing a new link cost vector $\mathbf{k}_{\mathrm{new}}$ from a defined "neighborhood" of $\mathbf{k}$. To be able to escape from a local minimum, $\mathbf{k}_{\mathrm{new}}$ is not only accepted as next current link cost vector if it is better than the current $\mathbf{k}$, but also if it is not worse than a threshold θ, i.e., if $f(\mathbf{k}_{\mathrm{new}}) < f(\mathbf{k}) + \theta$. The exploration of the search space continues until no more improvements can be found for a specified number of iteration steps.

Extension of threshold accepting for Pareto-optimization We modify the sketched threshold accepting algorithm to find Pareto-optimal results. We now have multiple objective functions f_i and a set of Pareto-optimal link costs $\mathcal{K}_{\mathrm{Par}}$

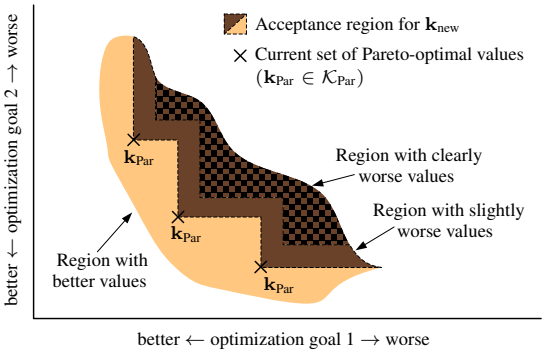
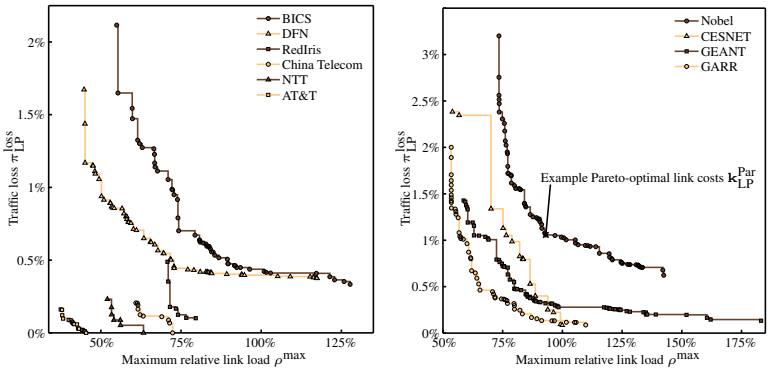


Figure 2.20: *Threshold accepting algorithm with two objective functions: acceptance region for a new link cost vector* $\mathbf{k}_{new}$.

instead of a single best result $\mathbf{k}_{best}$. We only need to define the acceptance regions for the extension of threshold accepting. A new link cost vector $\mathbf{k}_{new}$ is accepted if there is no other Pareto-optimal link cost vector $\mathbf{k}_{Par} \in \mathcal{K}_{Par}$ which is more than θ_i better than $\mathbf{k}_{new}$ in all objective functions f_i. This principle is depicted in Figure 2.20 for two objective functions. There is a region with better link costs, a region with acceptable link costs that are not Pareto-optimal, and a region with unacceptable link costs. After a new Pareto-optimal link cost vector has been found, link cost vectors that are no longer Pareto-optimal need to be removed from the set of Pareto-optimal link costs $\mathcal{K}_{Par}$.

2.2.4.3 Evaluation

We perform the above described Pareto-optimization for all test networks to minimize traffic loss π_{LP}^{loss} and the maximum relative link load ρ^{max}. Figures 2.21(a) and 2.21(b) reveal the outcome. The results of the different networks are partitioned into the two figures in a way that optimizes readability. For this reason also the scaling of the x-axis differs in both figures.



(a) Pareto-optimal link costs part I.

(b) Pareto-optimal link costs part II.

Figure 2.21: *Percentage of traffic loss π_{LP}^{loss} and maximum link load ρ^{max} for Pareto-optimal link costs.*

Each point in the figures corresponds to a Pareto-optimal link cost vector. Its position in the graph reveals the percentage of traffic without LFA protection π_{LP}^{loss} as well as the relative link load ρ^{max}. The Pareto-optimal link costs of a single network are linked by lines and identified by the same markers. Values for uniform link costs $\mathbf{k}_u$ are not presented in the figures for the sake of readability. They all lead to 100% maximum link load and their traffic loss is given in Table 2.6.

We first consider the AT&T, the NTT, and the China Telekom networks in Figure 2.21(a). All Pareto-optimal link costs of these networks extend only over a relatively small region. However, the link costs creating the least traffic loss π_{LP}^{loss} lead to about 15% more relative link load ρ^{max} than the link costs minimizing that metric. This is already a significant difference so that care must be taken in choosing an appropriate link cost vector for configuration.

For all other networks in both Figure 2.21(a) and Figure 2.21(b), the traffic loss due to missing LFAs and the relative link load of the Pareto-optimal link cost vectors are spread over a large region. Optimized link costs with low traffic

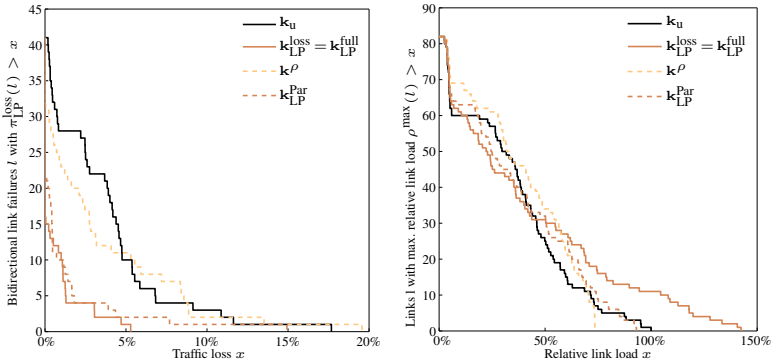
loss due to missing LFAs often lead to relative link loads above 100%, which is worse than with uniform link costs $\mathbf{k}_u$. In general, low values for relative link load apparently lead to large values of traffic loss and vice-versa. Thus, the two considered performance metrics seem to be contradicting optimization goals.

Nevertheless, the evaluations in Figures 2.21(a) and 2.21(b) also show for all investigated networks that some of the Pareto-optimal link costs perform relatively well with regard to traffic loss and maximum relative link loads. A network administrator can choose one of these link costs for configuration by trading off traffic loss π_{LP}^{loss} for maximum relative link load ρ^{max}. As a consequence, the network will face only little traffic loss due to missing LFAs and face limited link loads even after rerouting in case of single link failures.

2.2.4.4 Quality of Selected Pareto-Optimal Link Costs

For further analysis, we select the Pareto-optimal link costs $\mathbf{k}_{LP}^{Par}$ for the Nobel network that are marked in Figure 2.21(b). We compare them in detail with uniform link costs $\mathbf{k}_u$, link costs optimized to minimize the maximum link load $\mathbf{k}^{\rho}$, and link costs optimized to minimize the percentage of unprotected traffic $\mathbf{k}_{LP}^{loss}$.

Figure 2.22(a) displays the percentage of unprotected traffic in the Nobel network. It is similar to Figure 2.18 but includes different optimized link costs. With uniform link costs $\mathbf{k}_u$, at least some traffic cannot be protected in any single bidirectional link failure scenario and in 10 failure scenarios more than 5% traffic will be lost. With link costs optimized to minimize the maximum link load $\mathbf{k}^{\rho}$, some traffic remains unprotected in 32 out of 41 bidirectional single link failure scenarios and more than 5% of the traffic cannot be protected in 11 link failure scenarios. In contrast, with the selected Pareto-optimal link costs $\mathbf{k}_{LP}^{Par}$, some traffic remains unprotected in 22 out of 41 bidirectional single link failure scenarios and more than 5% of the traffic cannot be protected in only 2 link failure scenarios. Link costs optimized to minimize the percentage of unprotected traffic $\mathbf{k}_{LP}^{loss}$ lead to traffic loss in only 16 of 41 bidirectional link failure scenarios and the failure of only one link leads to more than 5% traffic loss. Thus, the optimized



(a) Distribution of traffic loss for all single bidirectional link failures.

(b) Distribution of maximum relative load for all unidirectional links.

Figure 2.22: *Comparison of Pareto-optimal link costs marked in Figure 2.21(b) with other link costs in the Nobel network for general LFAs.*

link costs $\mathbf{k}_{\mathrm{LP}}^{\mathrm{loss}}$ outperform all other presented link costs with regard to traffic loss, but the chosen Pareto-optimal link costs are not much worse.

Figure 2.22(b) provides the number of links $l \in \mathcal{E}$ for which the relative link load $\rho^{\max}(l)$ exceeds a certain link load value x. The link costs optimized to minimize the percentage of unprotected traffic $\mathbf{k}_{\mathrm{LP}}^{\mathrm{loss}}$ lead to maximum link loads larger than 75% on 18 out of 82 unidirectional links. This seems unacceptable compared with the performance of the other link costs. Even uniform link costs $\mathbf{k}_{\mathrm{u}}$ have only 6 links with maximum link loads larger than 75% and a significantly lower maximum load for most links. This is improved by the selected Pareto-optimal link costs $\mathbf{k}_{\mathrm{LP}}^{\mathrm{Par}}$ and of course by the link costs $\mathbf{k}^{\rho}$ minimizing the maximum relative link load. In particular, the maximum link load for $\mathbf{k}_{\mathrm{LP}}^{\mathrm{Par}}$ is lower than the one for uniform link costs $\mathbf{k}_{\mathrm{u}}$. This deeper analysis qualifies the selected Pareto-optimal link costs as a good tradeoff between link loads and unprotected traffic.

2.3 Configuration and Routing Optimization of PCN-based Admission Control

Network optimization for resilient IP networks as described in the sections before can be used to keep link loads at acceptable rates during considered failure scenarios. Additional capacity is required in the network to handle the traffic that has to be rerouted after a failure. This strategy is called capacity overprovisioning (CO). Unfortunately, it is usually not sufficient to guarantee certain Quality of Service (QoS) parameters like delay and packet loss to paying customers [54]. Traffic patterns can change quickly and failures can occur that were not accounted for during the network planning and optimization phase. As a solution to this problem, admission control (AC) was proposed for IP networks [53]. With AC, new users are only admitted to the network when there is enough capacity to cope with the users demand. There are no significant bandwidth savings when AC is used instead of, or in addition to CO for QoS provisioning [126]. However, the dynamic behavior of users and services and networks makes future demands unpredictable so that ISPs see the need for AC to offer premium services over integrated IP networks in the future. The "Congestion and Pre-Congestion Notification" (PCN) protocol that has recently been developed provides a feedback-based admission control for high-priority PCN traffic for single DiffServ domains [58]. It also provides a flow termination (FT) mechanism, which allows to terminate already admitted flows if the available network capacity runs low.

Each link l of a so-called PCN domain is associated with an admissible and a supportable rate threshold ($AR(l)$, $SR(l)$) and the egress nodes of the domain are notified via appropriately marked packets if these thresholds are exceeded by high-priority PCN traffic. This feedback is used to implement AC and FT. Various packet marking schemes as well as AC and FT methods are proposed [127]. Some proposals provide two metering and marking schemes [128, 129], to control the admissible and the supportable rate independently of each other (*dual marking PCN, DM-PCN*). Others provide only a single metering and marking scheme [130, 131] that controls only the admissible rate (*single marking PCN,*

SM-PCN). They implicitly assume that the supportable rate $SR(l)$ of a link l is a fixed multiple b of its admissible rate $AR(l)$ in the entire PCN domain

$$SR(l) = b \cdot AR(l). \tag{2.19}$$

As a consequence, egress nodes can infer from the ratio of marked and unmarked traffic whether only the admissible or also the implicit supportable rate is exceeded on some link. We call the parameter b the "backup factor" as it controls the relation of primary and backup capacity on the links. The advantage of SM-PCN is that it needs fewer codepoints in the IP header for packet marking and less metering and marking support by routers. Its disadvantage is that Constraint (2.19) limits traffic engineering capabilities and makes the configuration of the rate thresholds harder when resource efficiency is an objective. In addition, it does not work well with multipath routing and when single edge-to-edge aggregates carry only little traffic [127].

This section investigates the rate threshold setting problem for PCN-based AC and FT. Furthermore, it proposes objective functions for routing optimization in resilient PCN networks. Performance results compare the resource efficiency of DM-PCN and SM-PCN with and without routing optimization for a large set of sample networks. The algorithms presented in this study also serve to configure and optimize PCN networks in practice.

The content of this section is mainly taken from our study that was published in the journal "Computer Networks" [2]. Its structure is as follows. Section 2.3.1 reviews related work showing the historic roots of PCN and similar AC approaches. Section 2.3.2 introduces PCN and explains how AC and FT work in the single and dual marking PCN architecture (SM-PCN, DM-PCN). Section 2.3.3 proposes algorithms to set the admissible and supportable rate thresholds appropriately for resilient AC. Section 2.3.4 provides objective functions to optimize IP routing in order to maximize the admissible protected traffic. Section 2.3.5 compares the resource efficiency of SM-PCN and DM-PCN for a large set of networks with different characteristics.

2.3.1 Related Work on Congestion Notifications and Admission Control

We review related work regarding Random Early Detection (RED), Explicit Congestion Notification (ECN), and stateless core concepts for AC as they can be viewed as historic roots of PCN.

2.3.1.1 Random Early Detection

Random Early Detection (RED) was originally presented in [132], and in [133] it was recommended for deployment in the Internet. It was intended to detect incipient link congestion and to throttle only some TCP flows early to avoid severe congestion and to improve the TCP throughput. RED measures the average buffer occupation avg in routers and packets are dropped or marked with a probability that increases linearly with the average queue length avg.

2.3.1.2 Explicit Congestion Notification

Explicit Congestion Notification (ECN) is built on the idea of RED to signal incipient congestion to TCP senders to reduce their sending window [134]. Packets of not-ECN-capable flows can be differentiated by a "not-ECN-capable transport" (not-ECT, '00') codepoint from packets of a ECN-capable flow which have an "ECN-capable transport" (ECT, '10', '01') codepoint. In case of incipient congestion, RED gateways possibly drop not-ECT packets while they just switch the codepoint of ECT packets to "congestion experienced" (CE, '11') instead of discarding them. This improves the TCP throughput since packet retransmission is no longer needed. Both the ECN encoding in the packet header and the behavior of ECN-capable senders and receivers after the reception of a marked packet is defined in [134]. ECN comes with two different codepoints for ECT: ECT(0) ('10') and ECT(1) ('01'). They help to detect cheating network equipment or receivers [135] that do not conform to the ECN semantics. The four codepoints are encoded in the (currently unused) bits of the differentiated services codepoint

(DSCP) in the IP header which is a redefinition of the type of service octet [136]. The ECN bits can be redefined by other protocols and [137] gives guidelines for that. They are likely to be reused for encoding of PCN marks.

2.3.1.3 Admission Control

We briefly review some specific AC methods that can be seen as forerunners of the PCN principle. They measure the rate of admitted traffic on each link of a network and give feedback to the network boundary if that rate exceeds a pre-configured admissible rate threshold. Thereby, no per-flow reservations need to be kept for a link and the network core remains stateless. This is a key property of PCN-based AC.

Admission control based on reservation tickets To keep a reservation for a flow across a network alive, ingress routers send reservation tickets in regular intervals to the egress routers. Intermediate routers estimate the rate of the tickets and can thereby estimate the expected load. If a new reservation sends probe tickets, intermediate routers forward them to the egress router if they have still enough capacity to support the new flow and the egress router bounces them back to the ingress router indicating a successful reservation; otherwise, the intermediate routers discard the probe tickets and the reservation request is denied. Periodic reservation tickets do not need to be sent explicitly, their information can also be conveyed in form of some markings in normal data packets. Several stateless core mechanisms work according to this idea [138–140].

Admission control based on packet marking Gibbens and Kelly [141, 142] theoretically investigated AC based on the feedback of marked packets whereby packets are marked by routers based on a virtual queue with configurable bandwidth. This core idea is adopted by PCN. Marking based on a virtual instead of a physical queue also allows to limit the utilization of the link bandwidth by premium traffic to arbitrary values between 0 and 100%. Karsten and

Schmitt [143, 144] integrated these ideas into the IntServ framework and implemented a prototype. They point out that the marking can also be based on the CPU usage of the routers instead of the link utilization if this turns out to be the limiting resource for packet forwarding. An early version of a PCN-like AC has been reported in [145].

Resilient admission control Resilient admission control admits only as much traffic as still can be carried after rerouting in a protected failure scenario [126, 146]. This is necessary since overload in wide area networks mostly occurs due to link failures and not due to increased user activity [55]. It can be implemented with PCN by setting the admissible rate thresholds low enough such that the rate of PCN traffic on a link is lower than its supportable rate threshold after rerouting.

2.3.2 PCN-Based Flow Control

This section illustrates the basic idea of PCN-based admission control (AC) and flow termination (FT). An example illustrates how PCN-based AC and FT fit into the overall Internet structure. We review how AC can be implemented based on appropriate metering and marking schemes. FT methods may reuse the marking scheme for AC or require their own. This leads to the definition of a *single and dual marking PCN architecture (SM-PCN, DM-PCN)*. We show how PCN-based AC and FT can be used to implement conventional and resilient AC. Finally, we explain the threshold setting and routing optimization problem for resilient PCN-based AC and FT which is the focus of this section.

2.3.2.1 Pre-Congestion Notification

Pre-Congestion Notification (PCN) is intended for use in DiffServ networks and defines a new traffic class that receives preferred treatment by PCN nodes. It provides information to support AC and FT for this traffic type. PCN introduces an admissible and a supportable rate threshold ($AR(l)$, $SR(l)$) for each link l of

the network which imply three different link states as illustrated in Figure 2.23. If the PCN traffic rate $r(l)$ is below $AR(l)$, there is no pre-congestion and further flows may be admitted. If the PCN traffic rate $r(l)$ is above $AR(l)$, the link is *AR*-pre-congested and the traffic rate above $AR(l)$ is *AR*-overload. In this state, no further flows should be admitted. If the PCN traffic rate $r(l)$ is above $SR(l)$, the link is *AR*- and *SR*-pre-congested and the traffic rate above $SR(l)$ is *SR*-overload. In this state, some already admitted flows should be terminated.

PCN traffic enters a PCN domain with a "no-pre-congestion" (NP) codepoint. PCN nodes monitor the PCN traffic rate on their links and re-mark the codepoints of the packets depending on the pre-congestion states of these links. The PCN egress nodes evaluate the packet markings and their essence is reported to the AC and FT entities of the network so that they can admit or block new flows or even terminate already admitted flows. Therefore, this concept is called pre-congestion notification.

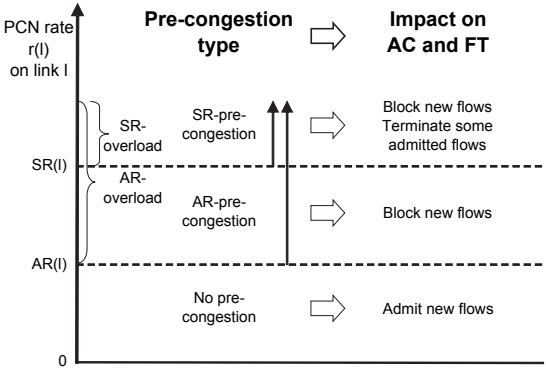


Figure 2.23: *The admissible and the supportable rate ($AR(l), SR(l)$) define three types of pre-congestion concerning the PCN traffic rate $r(l)$ on a link.*

2.3.2.2 Application of PCN in the Internet

There are different mechanisms for QoS support in the future Internet. Some domains use extensive capacity overprovisioning for all traffic. Others enable RSVP [147] in all nodes granting prioritized forwarding to flows with individual reservations according to the IntServ principle [148]. DiffServ relies on traffic prioritization for high priority traffic that is identified by an appropriate DiffServ codepoint and hence per-flow reservations are not required at all. To protect a network against overload, AC is required and flows must be individually treated at least at network boundaries. The IntServ-over-DiffServ concept [149] provides a controlled load (CL) service over DiffServ networks using per-flow AC at the ingress nodes of a domain. The CL service offers the same QoS a flow would receive from lightly loaded network elements [150] and is useful for inelastic flows, e.g., realtime media. The PCN mechanism can be used to implement AC and FT in those networks. A prerequisite is that admission requests for high-priority traffic are triggered by end-to-end signaling protocols such as SIP, RSVP, or similar mechanisms for each flow. Depending on the network-specific QoS support, this signaling is respected or ignored. This is depicted in Figure 2.24. The PCN ingress node of a PCN region may serve as AC entity and admits or blocks admission requests.

2.3.2.3 PCN-Based Admission Control

Admission control (AC) methods require that routers mark PCN traffic on links inside a PCN domain when they are *AR*-pre-congested. *Exhaustive marking* marks all PCN packets in that case with "admission-stop" (AS) while *excess marking* marks only those PCN packets that exceed the *AR* of the respective link. The currently preferred AC and FT methods work on aggregated feedback from ingress-egress aggregates (IEAs) [58] and an admission state indicating *admit* or *block* is kept per IEA. If the IEA is in the *admit* state, new flows fitting into this IEA are admitted, otherwise they are blocked. PCN egress nodes classify PCN packets according to their PCN ingress nodes and evaluate their markings

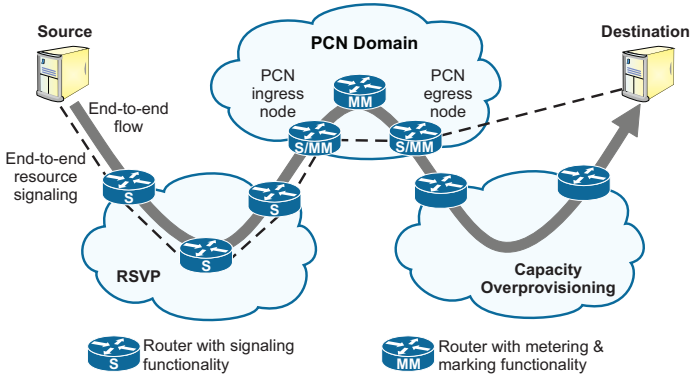


Figure 2.24: *PCN-based AC guarantees a controlled load (CL) service over a DiffServ region and per flow admission requests to the PCN domain are triggered by external signaling protocols.*

per IEA. At the end of a measurement interval, the egress nodes compute the congestion level estimate (CLE), i.e. the fraction of AS-marked packets. If the CLE exceeds an upper threshold $T_{\mathrm{CLE}}^{\mathrm{AStop}}$, the admission state is turned to *block* and if it falls below a lower threshold $T_{\mathrm{CLE}}^{\mathrm{ACont}}$, the admission state is turned to *admit*. To that end, admission-stop or admission-continue messages are sent to the AC entity of the network. Some proposals use this CLE-based AC (CLEBAC) in combination with exhaustive marking [128, 129] and some others with excess marking [130, 131]. There are other methods for PCN-based AC [151], but they are not needed for this study.

2.3.2.4 PCN-Based Flow Termination

Flow termination (FT) is a new flow control function protecting the network against congestion caused by already admitted traffic. At first sight, FT does not seem to be necessary when the admission of new PCN flows is controlled. However, admitted traffic can lead to severe overload so that it is beneficial for the

network to terminate some flows when the PCN traffic rate exceeds the *SR* of a link. *SR*-overload occurs due to various reasons. (1) In failure cases admitted traffic can be rerouted and cause congestion on the backup path. (2) Already admitted flows may change their typical behavior and switch from low bit rates to high bit rates. (3) Flows are possibly admitted before the effect of previously admitted flows is reflected by the markings and so overload can occur. This is likely in case of flash crowds when lots of flows request admission within short time. For all these reasons it makes sense to deploy FT in a network that already uses AC for the admission of new flows. Flow termination mechanisms may reuse the marking required for AC or they may require their own marking scheme. This leads to *single and dual marking PCN architectures (SM-PCN, DM-PCN)*. Various FT algorithms exist and a survey is given in [127]. In the following we present only two simple examples showing how FT works with SM- and DM-PCN.

Flow termination with dual marking DM-PCN uses two metering and marking algorithms. The AC method requires exhaustive marking based on the admissible rate as reference rate as described above. The FT method requires excess marking based on the supportable rate as reference rate. As a consequence, *SR*-overload is marked with "excess-traffic" (ET). To implement FT, the egress node determines the rate of ET-marked traffic (*ETR*) for each IEA and triggers the termination of appropriate flows from the IEA to quickly reduce the PCN traffic rate by *ETR* in order to remove *SR*-overload. The mechanisms in [128, 129] work similarly.

Flow termination with single marking SM-PCN uses only a single metering and marking algorithm. The AC method requires excess marking based on the admissible rate as reference rate. The FT method does not need another marking algorithm, it just requires that the supportable rates are fixed multiples of admissible rates (cf. Equation (2.19)). To implement FT, each egress node determines the rate of AS-marked and non-AS-marked traffic (*ASR*, *nASR*) per IEA. If the overall PCN traffic rate ($ASR + nASR$) is larger than b times the fraction of non-

AS-marked traffic ($b \cdot nASR < ASR + nASR$), some link was *SR*-pre-congested. Thus, the rate to be terminated from the IEA is

$$TR = \max(0, ASR + nASR - b \cdot nASR) \quad (2.20)$$

$$= \max(0, ASR - (b-1) \cdot nASR). \quad (2.21)$$

The mechanisms in [130, 152] work similarly.

Advantages and drawbacks of single and dual marking As mentioned above, SM-PCN requires less support in routers than DM-PCN. Furthermore, SM-PCN re-marks NP-marked packets only to "admission-stop" (AS) while DM-PCN re-marks NP-marked packets to "admission-stop" (AS) and "excess-traffic" (ET). Thus, DM-PCN requires more codepoints in the packet header than SM-PCN and is, therefore, harder to implement in today's Internet as free codepoints in the IP header are a scarce resource and hardly available. However, SM-PCN does not work well with multipath routing [127] and AC methods do not work well with small IEAs. They react with significant delay when the packet rate of the IEA is small because excess marking AS-marks only a small fraction of the traffic. Small IEAs are not negligible because they are expected to be the majority of IEAs in future core networks [153]. Nevertheless, SM-PCN currently seems to be the preferred option in the standardization process.

2.3.2.5 Conventional and Resilient Admission Control with and without Flow Termination

We discuss the use of conventional and resilient AC with and without FT.

Conventional AC The objective of conventional AC is to block new flows to avoid overload created by users. Almost the full link bandwidth can be used to carry high-priority traffic as long as delay bounds are respected. As a consequence, the admissible rate threshold $AR(l)$ of a link l can be set close to its bandwidth $\mathbf{c}(l)$ when the traffic is smooth enough.

Resilient AC In case of failures, traffic is possibly rerouted and can lead to congestion on backup paths. In fact, this is the major reason for congestion in today's Internet. As shown in [55], only 20% of the congestion observed in core networks are caused by increased user activity, but 80% of the congestion is caused by traffic which is redirected due to failures. Conventional AC cannot guarantee QoS for such cases, but this can be achieved by resilient AC [56,126]. We call the failures, for which no congestion should occur protected failures. Only a fraction of the link bandwidth can be used to carry primary traffic since the remaining fraction is required for backup purposes in case of protected failures. This needs to be respected by AC, and *AR*-thresholds must be set low enough.

Conventional AC with FT Conventional AC cannot avoid overload situations in case of failures. Therefore, it may be combined with FT. The supportable rates $SR(l)$ are also set close to the link bandwidth $\mathbf{c}(l)$, but larger than $AR(l)$. Some safety margin is required between $AR(l)$ and $SR(l)$ to avoid unwanted termination of admitted traffic and between $SR(l)$ and $\mathbf{c}(l)$ to avoid slow flow termination. In case of a failure, a large number of admitted flows are possibly terminated. This may be acceptable for some applications and unacceptable for others.

Networks using conventional AC with FT can be provided with sufficient backup capacity. The difference to resilient AC is that almost the entire link bandwidth can be used to admit new traffic. This has two implications. On the one hand, it reduces blocking when more traffic than expected requests admission. On the other hand, if more traffic than expected is admitted, the capacity on backup paths might not suffice in failure cases and hence flows must be terminated. Thus, resilient transport services cannot be provided for admitted traffic. However, they are desirable for demanding applications such as tele-medicine or tele-control of industrial applications.

Resilient AC with FT Resilient AC admits only as much traffic as can be carried without QoS degradation over the network after rerouting in case of pro-

tected failures. However, unlikely failures can happen for which backup capacity does not suffice. Therefore, FT is also a desirable function in combination with resilient AC. Again, the *SR* thresholds may be set close to the link bandwidths with a safety margin towards $\mathbf{c}(l)$ in order to guarantee a sufficiently fast termination process. *AR* thresholds are set to lower values. In contrast to conventional AC with FT, a flow is not likely to be terminated once it is admitted so that resilient transport services can be offered.

2.3.2.6 Threshold Configuration and Routing Optimization

When PCN-based AC is configured for conventional non-resilient AC, the *AR*-thresholds can be set to almost the link bandwidth and no sophisticated algorithms are required. Resilient AC in general is more difficult. In [56, 154] algorithms are provided to calculate tunnel-specific capacities for a resilient tunnel-based AC. However, this solution cannot be applied for resilient PCN-based AC. Resilient PCN-based AC requires the computation of link-specific *AR*- and *SR*-thresholds. They must be set in such a way that admitted traffic can be accommodated after rerouting in case of protected failure scenarios without being terminated. In case of DM-PCN, only *AR*-thresholds need to be calculated since *SR*-thresholds can be set close to the link bandwidth independently of corresponding *AR*-thresholds. This is different for SM-PCN as the *SR*- and *AR*-thresholds are connected via Equation (2.19) which makes the threshold assignment problem more complex.

The amount of traffic which can be carried over a network during normal operation and after rerouting in protected failure cases depends on the routing and rerouting function. Moreover, more flows can be carried when they have shorter paths. To be independent of this issue, we consider for throughput maximization problems the fraction or multiple of a traffic matrix that can be supported by a network. In the previous sections and in [16], we provided heuristic methods to optimize IP routing to maximize the protected transport capacity for a fraction or multiple of a given traffic matrix. It is applicable in DM-PCN, but not in SM-PCN because SM-PCN requires that the ratio of primary and backup capacity is

exactly $1/(b-1)$. Thus, IP routing optimization is more complex for SM-PCN than for DM-PCN and new objective functions are required.

We develop algorithms for the threshold setting and routing optimization problem to provide traffic engineering for resilient PCN-based AC and FT, both for DM-PCN and the more complex SM-PCN. Moreover, a performance study quantifies their difference in the ability to use network resources efficiently.

2.3.3 Threshold Configuration for PCN-Based Flow Control

In this section we propose simple and improved algorithms for the configuration of the *AR*- and *SR*-thresholds for SM- and DM-PCN. The simple algorithms set thresholds in such a way that the same fraction of all expected ingress-egress aggregates can be admitted as high priority traffic. This possibly leaves some of the link capacities unused. Therefore, the improved algorithms strive for a higher resource utilization while implementing max-min fairness [155] among ingress-egress aggregates with regard to their admissible rates. This is conceptually similar to the problems treated in [56, 154, 156] but significantly differs by technical constraints. We illustrate the effect of the simple and improved algorithms by numerical results.

2.3.3.1 Test Environment and Nomenclature

For our study, we use the Labnet03 network given in Figure 2.25 with equal capacity links. We assume a traffic matrix proportional to the city sizes. We use the wide-spread gravity model because of its simplicity although recent research has shown that other models are more realistic [120, 157]. However, our findings do not depend on the accuracy of the traffic matrix. Explicit formulae for the gravity model are given in [56, Equation (3.41)].

We use the same network model as in the previous sections, but here, $\mathbf{c}(l)$ only denotes the capacity of a link that can be used for the transmission of high priority traffic. The flows between any two routers $v, w \in \mathcal{V}$ constitute demand

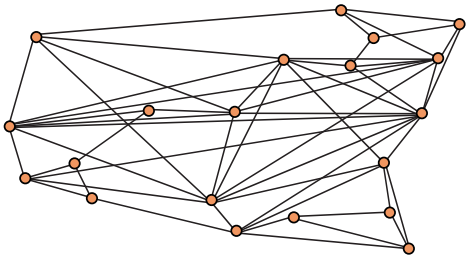


Figure 2.25: *Labnet 03 Topology.*

or ingress-egress aggregate (IEA) d whose rate $r(d)$ (or $r(d_{v,w})$) is given by the traffic matrix $\mathcal{D}$. We want PCN-based AC to prevent congestion in the presence of a set $\mathcal{S}$ of protected failure scenarios. In the performance studies in this section, $\mathcal{S}$ comprises the failure-free case as well as all single link and router failures. As standard link costs, again, we use uniform link costs $\mathbf{k}_{\mathrm{u}}$, while the link costs are modified in Section 2.3.4 to optimize the routing. As in the previous study, we use only single-path routing, taking the next-hop with the lowest ID in case of equal-cost paths. This is a reasonable decision because some methods for PCN-based AC and FT do not work well with multipath routing.

2.3.3.2 Simple Assignment of Admissible and Supportable Rate Thresholds

We present a simple, intuitive algorithm to set the admissible and supportable rate thresholds $AR(l)$ and $SR(l)$ for PCN-based AC and FT. The required inputs are the network bandwidths $\mathbf{c}(l), l \in \mathcal{E}$, the traffic matrix $r(d), d \in \mathcal{D}$, and the routing $u_s^{\mathbf{k}}(l, v, w)$. The objective is to set the AR-thresholds in such a way that all IEAs d can send the same maximum multiple $\sigma(\mathbf{k})$ of their expected rates $r(d)$ without causing congestion in protected failure scenarios $s \in \mathcal{S}$ after rerouting. In the following, we call this maximum multiple $\sigma(\mathbf{k})$ the "scaling factor". It is the metric chosen for the performance comparison.

Dual marking PCN architecture The largest link utilization in the network including protected failures is $\rho^{\max}_{\mathcal{S},\mathcal{E}}(\mathbf{k})$. Thus, scaling the traffic matrix by

$$\sigma_{DM}(\mathbf{k}) = \frac{1.0}{\rho^{\max}_{\mathcal{S},\mathcal{E}}(\mathbf{k})}, \tag{2.22}$$

prevents link utilizations $\rho(\mathbf{k}, l, s)$ from exceeding 100% in any protected failure scenario $s \in \mathcal{S}$. Therefore, we compute the *AR*- and *SR*-thresholds for DM-PCN

$$AR(l) = \sigma(\mathbf{k}) \cdot \sum_{d_{v,w} \in \mathcal{D}} r(d_{v,w}) \cdot u^{\mathbf{k}}_{\emptyset}(l, v, w)) \tag{2.23}$$

$$SR(l) = \sigma(\mathbf{k}) \cdot \max_{s \in \mathcal{S}} \left(\sum_{d_{v,w} \in \mathcal{D}} r(d_{v,w}) \cdot u^{\mathbf{k}}_{s}(l, v, w) \right) \tag{2.24}$$

by scaling the expected link loads under failure-free operation and their maximum over all protected failure scenarios with $\sigma(\mathbf{k}) = \sigma_{DM}(\mathbf{k})$. With the proposed thresholds, the traffic fraction $\sigma(\mathbf{k})$ of all IEAs can be admitted and the largest relative link load is at most 1.0 in any protected failure scenario $s \in \mathcal{S}$.

The traffic matrix is only a long-time expectation for planning purposes, but short-time variations can occur. With *AR*-thresholds configured according to Equation (2.23) AC can admit more traffic for a particular IEA d than $\sigma(\mathbf{k}) \cdot r(d)$ and less of another. If this happens, some traffic is possibly not protected and hence may be terminated in case of a very special failure scenario. This observation holds for PCN-based AC and FT in general and is not specific to our algorithms.

Single marking PCN architecture We set the *AR*- and *SR*-thresholds for the single marking architecture in a similar way. Without AC, the maximum link utilization in all protected failure scenarios is $\rho^{\max}_{\mathcal{S},\mathcal{E}}(\mathbf{k})$; a maximum link utilization $\rho^{\max}_{\mathcal{E}}(\mathbf{k}, \emptyset)$ is observed in the failure-free scenario and Constraint (2.19) requires that up to the b-multiple of this traffic needs to be accommodated in failure

scenarios; otherwise, *SR*-pre-congestion cannot be detected. When scaling the traffic matrix with

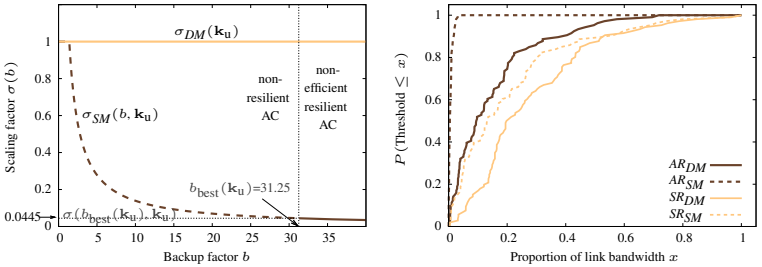
$$\sigma_{SM}(b, \mathbf{k}) = \frac{1.0}{\max\left(\rho^{\max}_{\mathcal{S},\mathcal{E}}(\mathbf{k}); b \cdot \rho^{\max}_{\mathcal{E}}(\mathbf{k}, \emptyset)\right)}, \tag{2.25}$$

neither the relative link utilizations $\rho(\mathbf{k}, l, s)$ nor the expression $b \cdot \rho(\mathbf{k}, l, \emptyset)$ exceed 1.0 and at least one of them is exactly 1.0 for at least one link $l \in \mathcal{E}$ and failure scenario $s \in \mathcal{S}$. Finally, the *AR*- and *SR*-thresholds can be set according to Equation (2.23) using $\sigma(\mathbf{k}) = \sigma_{SM}(b, \mathbf{k})$ and to Equation (2.19).

Comparison The scaling factor $\sigma(\mathbf{k})$ expresses the multiple of the traffic matrix that can be admitted as protected priority traffic. Therefore, it is a suitable measure to compare the efficiency of SM- and DM-PCN. Initially we use routing with uniform link costs $\mathbf{k}_{\mathrm{u}}$ and choose the overall traffic load in the network so that we get a scaling factor of $\sigma_{DM}(\mathbf{k}_{\mathrm{u}}) = 1.0$ for DM-PCN. For SM-PCN the scaling factor $\sigma_{SM}(b, \mathbf{k}_{\mathrm{u}})$ depends on the backup factor b and Figure 2.26(a) illustrates that it decreases with increasing b. The optimum backup factor is

$$b_{\text{best}}(\mathbf{k}) = \max_{l \in \mathcal{E}} \left(\frac{\rho^{\max}_{\mathcal{S}}(\mathbf{k}, l)}{\rho(\mathbf{k}, l, \emptyset)} \right). \tag{2.26}$$

For backup factors b smaller than $b_{\text{best}}(\mathbf{k})$, SM-PCN is not resilient: the *AR*-thresholds are set low enough that the link capacity will suffice to carry rerouted admitted traffic, but the *SR*-thresholds are possibly set to values that are too low so that some flows will be unnecessarily terminated in protected failure scenarios. If b is larger than $b_{\text{best}}(\mathbf{k})$, SM-PCN is resilient, but the large backup factor reserves too much backup capacity resulting in smaller *AR*-values. The best backup factor for the experimental setting in the Labnet03 is $b_{\text{best}}(\mathbf{k}_{\mathrm{u}}) = 31.25$ and leads to a scaling factor of $\sigma_{SM}(b_{\text{best}}(\mathbf{k}_{\mathrm{u}}), \mathbf{k}_{\mathrm{u}}) = 0.0445$. Thus, SM-PCN can carry only 4.4% of the traffic that can be supported by DM-PCN.



(a) Scaling factor $\sigma(b, \mathbf{k}_u)$ depending on the backup factor b. SM-PCN is resilient only for $b > b_{best}(\mathbf{k}_u)$.

(b) CDF of *AR*- and *SR*-threshold sizes relative to the respective link bandwidths.

Figure 2.26: *Simple threshold assignment for SM- and DM-PCN. The routing is based on uniform link costs* $\mathbf{k}_u$.

In the following, we choose the backup factor b for SM-PCN according to Equation (2.26). Figure 2.26(b) illustrates the impact of SM- and DM-PCN on the *AR*- and *SR*-threshold sizes achieved by the above algorithm. The figure shows the cumulative distribution function (CDF) of the threshold sizes relative to the respective link bandwidths. The x-axis indicates the proportion of link bandwidth x and the y-axis indicates the percentage of links for which the relative *AR*- or *SR*-threshold sizes $\frac{AR(l)}{\mathbf{c}(l)}$ and $\frac{SR(l)}{\mathbf{c}(l)}$ are smaller than or equal to x. Both SM- and DM-PCN have at least one *SR*-threshold using 100% of the respective link bandwidth. This shows that scaling factors $\sigma_{DM}(\mathbf{k}_u)$ and $\sigma_{SM}(b, \mathbf{k}_u)$ cannot be further increased. The *AR*-thresholds are substantially smaller than the *SR*-thresholds, especially for SM-PCN, which is due to the large backup factor b that cannot be decreased without losing the resilience property of the AC. The average relative size of the *AR*-thresholds is 14.94% for DM-PCN while it is only 0.67% for SM-PCN. Thus, SM-PCN can admit only very little high-priority traffic with resilience requirements.

2.3.3.3 Improved Threshold Assignment of Admissible and Supportable Rates

In the section above, all IEAs were associated with the same scaling factor $\sigma(\mathbf{k})$ which was used to set the *AR*-and *SR*-thresholds based on Equation (2.23), (2.24), and (2.19). We introduce now IEA-specific scaling factors $\sigma(d, \mathbf{k})$, i.e., the scaling factors of some IEAs can be increased if enough resources are available. This leads to larger *AR*- and *SR*-thresholds and allows better usage of link bandwidths.

The basic idea is as follows. The *AR*-threshold $AR(l)$ limits the admissible rate for all IEAs being carried over a specific link l. They can be scaled up to a certain value $\sigma(l, \mathbf{k})$ if their rate is not yet limited by other thresholds. Thus, $\sigma(l, \mathbf{k})$ indicates the competition for resources on link l: a low value expresses scarce resources while a large value expresses abundant resources. Consider an IEA d and its path $p(d)$. The IEA-specific scaling factor $\sigma(d, \mathbf{k}) = \min_{l \in p(d)} (\sigma(l, \mathbf{k}))$ is the minimum scaling factor of the links in the path of d. Limiting the rate of d according to this scaling factor assures that the capacity of the bottleneck link on its path is shared fairly among the flows competing for this link. Conversely, to limit a scaling factor $\sigma(d, \mathbf{k})$ for a certain IEA d, at least one *AR*-threshold of the links along its path $p(d)$ needs to be set to a sufficiently low value.

Dual marking PCN architecture Algorithm 1 iteratively determines the IEA-specific scaling factors $\sigma(d, \mathbf{k})$ for all IEAs $d \in \mathcal{D}$ and sets the link-specific *AR*-thresholds. Before we explain the algorithm, we need some nomenclature and auxiliary functions.

We call an IEA d "fixed" if its scaling factor $\sigma(d, \mathbf{k})$ is already determined; otherwise we call it "free". The set of all fixed and all free IEAs is denoted by $\mathcal{D}_{\text{fixed}}$ and $\mathcal{D}_{\text{free}}$. The set of IEAs with traffic routed over a specific link l in a specific failure scenario s is denoted by $\mathcal{D}(l, s) = \{d \in \mathcal{D} : u_s^{\mathbf{k}}(l, v, w) > 0\}$. If a link l carries a certain set of fixed and free IEAs in a specific failure scenario s, the capacity left over by the fixed IEAs can be shared among the free IEAs. Thus, we can calculate an upper bound on the link- and failure-scenario-specific

Algorithm 1 Computation of improved AR-thresholds.

Require: $\mathcal{D}, \mathcal{D}(l,s)$
1: $\mathcal{D}_{\text{free}} = \mathcal{D}, \mathcal{D}_{\text{fixed}} = \emptyset, \mathcal{E}^{AR}_{\text{fixed}} = \emptyset$
2: **while** $\mathcal{D}_{\text{free}} \neq \emptyset$ **do**
3: Calculate $\sigma^{\text{free}}_{\text{min}}(\mathbf{k})$ according to Equation (2.28)
4: $\mathcal{B} = \emptyset$ {Collect all bottlenecked IEAs in $\mathcal{B}$}
5: **for all** $s \in \mathcal{S}, l \in \mathcal{E}$ **do**
6: **if** $\sigma(l,s,\mathbf{k}) = \sigma^{\text{free}}_{\text{min}}(\mathbf{k})$ **then**
7: **for all** $d_{v,w} \in \mathcal{D}_{\text{free}}$ **do**
8: **if** $u^{\mathbf{k}}_s(l,v,w) > 0$ **then**
9: $\mathcal{B} = \mathcal{B} \cup d$
10: **end if**
11: **end for**
12: **end if**
13: **end for**
14: **while** $\mathcal{B} \neq \emptyset$ **do** {Enforce scaling factor $\sigma^{\text{free}}_{\text{min}}(\mathbf{k})$ for bottlenecked IEAs by setting AR-thresholds small enough.}
15: choose appropriate $d^*_{v,w} \in \mathcal{B}$
16: choose appropriate $l^* \in \mathcal{E} \setminus \mathcal{E}^{AR}_{\text{fixed}} : u^{\mathbf{k}}_s(l,v,w) > 0$
17: $AR(l^*) = \sum_{d_{v,w} \in \mathcal{D}_{\text{fixed}}} \sigma(d_{v,w},\mathbf{k}) \cdot r(d_{v,w}) \cdot u^{\mathbf{k}}_{\emptyset}(l^*,v,w) +$
18: $\sigma^{\text{free}}_{\text{min}}(\mathbf{k}) \cdot \sum_{d \in \mathcal{D}_{\text{free}}} r(d_{v,w}) \cdot u^{\mathbf{k}}_{\emptyset}(l^*,v,w)$
19: $\mathcal{E}^{AR}_{\text{fixed}} = \mathcal{E}^{AR}_{\text{fixed}} \cup l^*$
20: **for all** $d \in (\mathcal{D}(l^*,\emptyset) \cap \mathcal{D}_{\text{free}})$ **do**
21: $\sigma(d,\mathbf{k}) = \sigma^{\text{free}}_{\text{min}}(\mathbf{k})$
22: $\mathcal{B} = \mathcal{B} \setminus g$
23: $\mathcal{D}_{\text{free}} = \mathcal{D}_{\text{free}} \setminus d$
24: $\mathcal{D}_{\text{fixed}} = \mathcal{D}_{\text{fixed}} \cup d$
25: **end for**
26: **end while**
27: **end while**
28: **return** Scaling factors $\sigma(d,\mathbf{k})$ for $d \in \mathcal{D}$, threshold sizes $AR(l)$ for $l \in \mathcal{E}^{AR}_{\text{fixed}}$

scaling factor $\sigma(l,s,\mathbf{k})$ if link l carries at least one free IEA, by

$$\sigma(l,s,\mathbf{k}) = \frac{\mathbf{c}(l) - \sum_{d_{v,w}\in\mathcal{D}_{\text{fixed}}} \sigma(g,\mathbf{k}) \cdot r(d_{v,w}) \cdot u_s^{\mathbf{k}}(l,v,w)}{\sum_{d_{v,w}\in\mathcal{D}_{\text{free}}} r(d_{v,w}) \cdot u_s^{\mathbf{k}}(l,v,w)}. \tag{2.27}$$

Furthermore, we determine the smallest free scaling factor $\sigma_{\text{min}}^{\text{free}}(\mathbf{k})$ by

$$\sigma_{\text{min}}^{\text{free}}(\mathbf{k}) = \min_{\{l\in\mathcal{E},s\in\mathcal{S}:|\mathcal{D}(l,s)\cap\mathcal{D}_{\text{free}}|>0\}} \left(\sigma(l,s,\mathbf{k})\right), \tag{2.28}$$

among the combinations of (l,s) with at least one free IEA. Those combinations with $\sigma(l,s,\mathbf{k}) = \sigma_{\text{min}}^{\text{free}}(\mathbf{k})$ are bottleneck combinations and we call the respective free IEAs "bottlenecked IEAs".

Algorithm 1 starts with initializing the set of free IEAs by $\mathcal{D}_{\text{free}} = \mathcal{D}$, the set of fixed IEAs by $\mathcal{D}_{\text{fixed}} = \emptyset$, and the set of links with already assigned AR-thresholds by $\mathcal{E}_{\text{fixed}}^{AR} = \emptyset$. The algorithm repeats the following steps until all IEAs are fixed.

The minimum scaling factor $\sigma_{\text{min}}^{\text{free}}(\mathbf{k})$ of the free IEAs is calculated and the bottlenecked IEAs are collected in the set $\mathcal{B}$. Their scaling factors $\sigma(d,\mathbf{k})$ need to be limited to $\sigma_{\text{min}}^{\text{free}}(\mathbf{k})$ by setting at least one *AR*-threshold on their paths small enough. Thus, the algorithm repeats the following steps until the set of bottlenecked IEAs $\mathcal{B}$ is empty.

An appropriate IEA d^* is chosen from the set $\mathcal{B}$. It can be, e.g., an IEA with a shortest (longest) path. Other criteria are possible. To limit the scaling factor $\sigma(d,\mathbf{k})$ of this IEA, a suitable link l^* is chosen from the path $p(d)$ for which the *AR*-threshold is not yet determined. Such a link l^* carries, e.g., the smallest (largest) number of free IEAs, the smallest (largest) rate of free IEAs, or it carries on average the shortest (longest) free IEAs. [3] The correct size of this *AR*-threshold is determined and the link l^* is added to the set $\mathcal{E}_{\text{fixed}}^{AR}$. All other free IEAs that are carried over l^* in the failure-free scenario are also limited by this new *AR*-threshold. Therefore, their scaling factor is also set to $\sigma(d,\mathbf{k}) = \sigma_{\text{min}}^{\text{free}}(\mathbf{k})$, they are

[3] In our experiments, the results of this algorithm were rather insensitive towards different policies.

removed from the set of bottlenecked IEAs $\mathcal{B}$, and moved from $\mathcal{D}_{\text{free}}$ to $\mathcal{D}_{\text{fixed}}$. [4]

The algorithm terminates since at least one free IEA becomes fixed in each outer while loop. At program termination, the scaling factors $\sigma(d, \mathbf{k})$ are determined for all IEAs $d \in \mathcal{D}$ as well as the *AR*-thresholds for all links $l \in \mathcal{E}^{AR}_{\text{fixed}}$. In pathological scenarios where IEAs with one-link paths are missing, *AR*-thresholds for some links might not be fixed because the scaling factors of all IEAs carried over these links are already limited by the *AR*-thresholds of other links. Then, these *AR*-thresholds can be set to

$$AR(l) = \sum_{d_{v,w} \in \mathcal{D}} \sigma(d_{v,w}, \mathbf{k}) \cdot r(d_{v,w}) \cdot u^{\mathbf{k}}_{\emptyset}(l, v, w). \tag{2.29}$$

The *SR*-thresholds can be set to values of

$$SR(l) = \max_{s \in \mathcal{S}} \left(\sum_{d_{v,w} \in \mathcal{D}} \sigma(d_{v,w}, \mathbf{k}) \cdot r(d_{v,w}) \cdot u^{\mathbf{k}}_{s}(l, v, w) \right) \tag{2.30}$$

or larger as long as they are smaller than $\mathbf{c}(l)$.

Single marking PCN architecture The threshold assignment for SM-PCN works similarly. However, unlike DM-PCN, only $\frac{1}{b}$ of the maximum bandwidth $\mathbf{c}(l)$ is available to admit traffic in the failure-free case because of Constraint (2.19). Therefore, we adjust Equation (2.27) to calculate $\sigma(l, \emptyset, \mathbf{k})$ for the failure-free scenario by the following equation:

$$\sigma(l, \emptyset, \mathbf{k}) = \frac{\frac{\mathbf{c}(l)}{b} - \sum_{d_{v,w} \in \mathcal{D}_{\text{fixed}}} \sigma(d_{v,w}, \mathbf{k}) \cdot r(d_{v,w}) \cdot u^{\mathbf{k}}_{\emptyset}(l, v, w)}{\sum_{d \in \mathcal{D}_{\text{free}}} r(d_{v,w}) \cdot u^{\mathbf{k}}_{\emptyset}(l, v, w)}. \tag{2.31}$$

The *AR*- and *SR*-thresholds for SM-PCN are calculated in two steps. First, we determine the appropriate backup factor $b_{\text{best}}(\mathbf{k})$ based on the expected, unscaled

[4] This part of the algorithm is limited to single path routing for which PCN is currently designed.

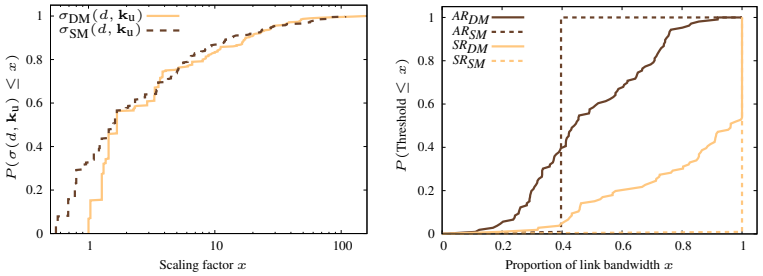
traffic matrix using Equation (2.26). We calculate the *AR*-thresholds according to Algorithm 1 based on $b_{\text{best}}(\mathbf{k})$ and the scaling factors in Equation (2.31) instead of Equation (2.27) where applicable. In a second step, we determine again the appropriate backup factor $b^*_{\text{best}}(\mathbf{k})$ using Equation (2.26) but based on the scaled traffic matrix $(r(d) \cdot \sigma(d, \mathbf{k}))_{d \in \mathcal{D}}$. The new value $b^*_{\text{best}}(\mathbf{k})$ is possibly smaller than the value $b_{\text{best}}(\mathbf{k})$ from the first step. In that case, at most $\frac{b^*_{\text{best}}(\mathbf{k})}{b_{\text{best}}(\mathbf{k})}$ of any link capacity will be used in any considered failure scenario $s \in \mathcal{S}$. Therefore, we finally multiply the obtained scaling factors $\sigma(d, \mathbf{k})$, $AR(l)$, and $SR(l)$ by $\frac{b_{\text{best}}(\mathbf{k})}{b^*_{\text{best}}(\mathbf{k})}$ to maximize the rate thresholds without risking overload in any $s \in \mathcal{S}$.

Comparison We calculate the IEA-specific scaling factors $\sigma(d, \mathbf{k})$ and the *AR*- and *SR*-threshold sizes according to the improved threshold assignment algorithm. Simple threshold assignment leads to a common scaling factor for all IEAs of 1.0 and 0.0445 for DM- and SM-PCN, respectively. Improved threshold assignment increases the scaling factors to average values of 6.90 and 6.51. However, the minimum scaling factor

$$\sigma^{\mathcal{D}}_{\text{min}}(\mathbf{k}) = \min_{d \in \mathcal{D}} \left(\sigma(d, \mathbf{k})\right) \tag{2.32}$$

limits the supportable scaling of the entire traffic matrix and the corresponding values are $\sigma^{\mathcal{D}}_{\text{min}}(\mathbf{k}_{\text{u}}) = 1.0$ and 0.5519. Thus, the value for DM-PCN does not change, but SM-PCN benefits a lot from improved threshold assignment. The CDF of individual IEA-specific scaling factors is illustrated in Figure 2.27(a) both for SM- and DM-PCN. They are distributed over a broad range with maximum values at 156.55 and 107.41. Most of the IEA-specific scaling factors $\sigma(d, \mathbf{k})$ for SM-PCN are significantly smaller than those of DM-PCN. Therefore, DM-PCN is still clearly more efficient than SM-PCN.

We study the impact of the improved threshold assignment on the relative *AR*- and *SR*-threshold sizes. Figure 2.27(b) illustrates their CDFs and a comparison with Figure 2.26(b) shows that the threshold sizes are significantly larger with improved threshold assignment than with simple threshold assignment. The av-



(a) CDF of ingress-egress-aggregate specific scaling factors $\sigma(d, \mathbf{k}_u)$.

(b) CDF of *AR*- and *SR*-threshold sizes relative to the respective link bandwidths.

Figure 2.27: *Improved threshold assignment for SM- and DM-PCN. The routing is based on uniform link costs* $\mathbf{k}_u$.

erage relative *AR*-threshold size increases from 14.94% to 48.75% for DM-PCN and from 0.67% to 39.66% for SM-PCN. We observe the tremendous increase of the *AR*-threshold sizes for SM-PCN because the improved threshold assignment decreases the backup factor from $b_{\text{best}}(\mathbf{k}_u) = 31.25$ down to $b^*_{\text{best}}(\mathbf{k}_u) = 2.52$. A closer look at the CDF of the *AR*-thresholds for SM-PCN in Figure 2.27(b) shows that all *AR*-thresholds are set to exactly 39.66% of the respective link bandwidth and the corresponding *SR*-thresholds are set to 100%. This is different for DM-PCN: some *AR*- and *SR*-thresholds use only 20% of the link bandwidth, and some others use more than 80%. Thus, optimum threshold sizes for DM-PCN are more heterogeneous than for SM-PCN.

2.3.4 Routing Optimization for PCN-Based Flow Control

In this section we derive objective functions for routing optimization to maximize the protected throughput of high-priority traffic for both SM- and DM-PCN. We illustrate the effect of the algorithms by numerical results.

2.3.4.1 Routing Optimization to Increase *AR*- and *SR*-Threshold Sizes

The maximum link utilization in the failure-free scenario $\rho_{\mathcal{E}}^{\max}(\mathbf{k}, \emptyset)$ can be minimized by routing optimization. In IP networks, the routing depends on the link costs $\mathbf{k}$ whose setting can be optimized so that $\rho_{\mathcal{E}}^{\max}(\mathbf{k}, \emptyset)$ is minimized [59]. In a similar way, the maximum link utilization for a set of protected failure scenarios $\mathcal{S}$ can be reduced [46, 82, 83]. We adopt and adapt this principle to increase the *AR*- and *SR*-thresholds by increasing the scaling factors $\sigma(\mathbf{k})$. To that end, we use the optimization software as described in Section 2.1.3 to find suitable link costs $\mathbf{k}_{\text{best}}$ that minimize a given objective function.

Dual marking PCN architecture To maximize the protected admissible traffic in terms of a proportion of a given traffic matrix for DM-PCN, $\sigma_{DM}(\mathbf{k})$ in Equation (2.22) needs to be maximized. This is achieved by finding a link cost vector $\mathbf{k}_{\text{best}}$ that minimizes the following objective function:

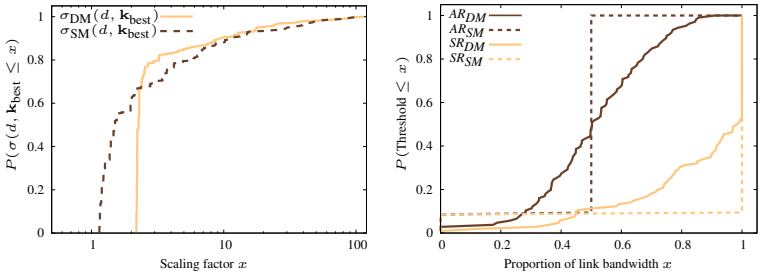
$$\rho_{\mathcal{S},\mathcal{E}}^{\max}(\mathbf{k}) \rightarrow \min . \tag{2.33}$$

Single marking PCN architecture To maximize the protected admissible traffic in terms of a proportion of a given traffic matrix for SM-PCN, $\sigma_{SM}(b_{\text{best}}^{*}(\mathbf{k}), \mathbf{k})$ in Equation (2.25) needs to be maximized. This is achieved by finding a link cost vector $\mathbf{k}_{\text{best}}$ that minimizes the following objective function:

$$\max\left(\rho_{\mathcal{S},\mathcal{E}}^{\max}(\mathbf{k}), b_{\text{best}}^{*}(\mathbf{k}) \cdot \rho_{\mathcal{E}}^{\max}(\mathbf{k}, \emptyset)\right) \rightarrow \min . \tag{2.34}$$

Hereby, the scaling factor $b_{\text{best}}^{*}(\mathbf{k})$ is calculated as in Section 2.3.3.3.

Comparison We use our routing optimization framework to minimize the objective functions presented above for SM- and DM-PCN and calculate the scaling factors as well as the *AR*- and *SR*-thresholds by improved threshold assignment. Compared to improved threshold assignment without routing optimization, the



(a) CDF of the ingress-egress-aggregate specific specific scaling factors $\sigma(d, \mathbf{k}_{\text{best}})$.

(b) CDF of the *AR*- and *SR*-threshold sizes relative to the respective link bandwidths.

Figure 2.28: *Improved threshold assignment for SM- and DM-PCN. The routing is based on optimized link costs* $(\mathbf{k} = \mathbf{k}_{best})$.

minimum scaling factors improve from $\sigma^{\mathcal{D}}_{\text{min}}(\mathbf{k}_{\text{u}}) = 1.0$ to $\sigma^{\mathcal{D}}_{\text{min}}(\mathbf{k}_{\text{best}}) = 2.1858$ for DM-PCN and from $\sigma^{\mathcal{D}}_{\text{min}}(\mathbf{k}_{\text{u}}) = 0.5519$ to $\sigma^{\mathcal{D}}_{\text{min}}(\mathbf{k}_{\text{best}}) = 1.1467$ for SM-PCN. Thus, DM-PCN is still about two times more efficient than SM-PCN when routing optimization is applied. The improvement for SM-PCN is partly due to a further reduction of the optimum backup factor from $b^{*}_{\text{best}}(\mathbf{k}_{\text{u}}) = 2.52$ to $b^{*}_{\text{best}}(\mathbf{k}_{\text{best}}) = 2.0$. The corresponding CDFs of the IEA-specific scaling factors are illustrated in Figure 2.28(a). The IEA-specific scaling factors for optimized link costs are more centered around their minimum values than those using uniform link costs (cf. Figure 2.27(a)). This holds for both SM- and DM-PCN.

After combined routing optimization and improved threshold assignment, the average relative *AR*- and *SR*-threshold sizes are 51.02% and 85.73% for DM-PCN and 45.75% and 91.51% for SM-PCN. However, looking at their CDF in Figure 2.28(b) we observe that the optimized routing for SM-PCN avoids carrying traffic on a few links. This prevents large backup factors that reduce the scaling factors for SM-PCN. The relative *AR*- and *SR*-threshold sizes of the used links are 50% and 100%, respectively. All used links have the same threshold sizes because of improved threshold assignment.

2.3.5 Efficiency of SM- and DM-PCN: A Parametric Study

In this section, we study the ability of SM- and DM-PCN to carry as much protected high-priority traffic as possible. We investigate the impact of simple and improved threshold assignment as well as routing optimization in networks of different size and with different node degree to generalize the results of Sections 2.3.3 and 2.3.4. We first describe the experiment setup and the exact performance measure and then discuss the results.

2.3.5.1 Experiment Setup and Performance Measure

A prerequisite for resilient AC is a resilient network topology which should be at least 2-connected, i.e., any node in the network can fail without partitioning its topology into disconnected subgraphs. Such structures are found in the core of wide area networks, but usually not in access networks. In typical full-fleshed Internet topologies, the number of links connected to a node, i.e. the node degree, usually follows a power law distribution as some few core nodes connect many satellite nodes. This, however, does not lead to a resilient network structure. We use the topology generator of [56] that allows to control the network parameters quite strictly. We randomly generate 15 networks for each combination of size 10, 15, 20, 25, 30, 35, 40, 45, and 50 nodes with an average node degree of 4, 5, and 6 and a maximum deviation from that average of 1. Thus, our experiments comprise altogether 405 different topologies. We use equal link bandwidths and homogenous traffic matrices. As the link bandwidths are not tailored to the traffic matrix, bottlenecks occur on some links.

Our intention is to compare the efficiency of SM- and DM-PCN configured with different threshold assignment algorithms with and without routing optimization. We want to maximize the multiple of the traffic matrix that can be admitted as protected high-priority traffic. This factor is the minimum scaling factor

$\sigma^{\mathcal{D}}_{\text{min}}(\mathbf{k})$ (cf. Equation (2.32)). We calculate the maximum resource utilization

$$\rho(\mathbf{k}) = \sigma^{\mathcal{D}}_{\text{min}}(\mathbf{k}) \cdot \sum_{l \in \mathcal{E}} \left(\sum_{d_{v,w} \in \mathcal{D}} r(d_{v,w}) \cdot u^{\mathbf{k}}_{\emptyset}(l, v, w) \right) / \sum_{l \in \mathcal{E}} \mathbf{c}(l) \qquad (2.35)$$

of the network based on this scaled traffic matrix and use it as simple performance metric to compare the efficiency of different AC types and configurations in different network topologies.

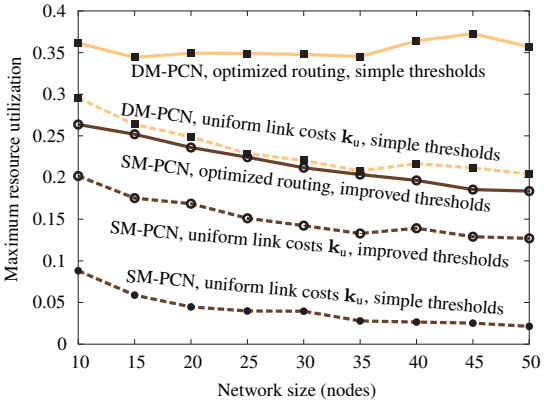


Figure 2.29: *Maximum resource utilization of SM- and DM-PCN with different configurations depending on the network size (nodes).*

2.3.5.2 Efficiency of SM- and DM-PCN with Different Configurations

For each network topology we calculate the maximum resource utilization for five different combinations of AC type, threshold assignment, and routing. Figure 2.29 shows the averaged results depending on the network size. SM-PCN with routing based on uniform link costs and simple threshold assignment is least efficient (4.1% utilization over all experiments), but it can be significantly en-

hanced by improved threshold assignment (15.2%). The combination of routing optimization and improved threshold assignment yields a further increase of the AC efficiency (21.7%). DM-PCN with simple threshold assignment and routing based on uniform link costs makes already good use of the network bandwidth (23.3%) and routing optimization further increases its efficiency (35.5%). Improved threshold assignment cannot increase the minimum scaling factor $\sigma^{\mathcal{D}}_{\min}(\mathbf{k})$ for DM-PCN, therefore, the corresponding results are missing. For most curves we observe the trend that the maximum resource utilization decreases with increasing network size. This is due to the fact that the probability for strong bottlenecks increases with the network size since the network bandwidths are not tailored to the need of the traffic matrix. Only routing optimization for DM-PCN is able to compensate this structural shortcoming. We also analyzed the impact of the node degree on the resource utilization, but we have not observed significant dependencies.

After all, improved threshold assignment is crucial to configure SM-PCN for efficient operation. Routing optimization can increase the efficiency of both SM- and DM-PCN. However, DM-PCN can carry significantly more traffic than SM-PCN with and without routing optimization especially in large networks.

2.4 Lessons Learned

In this chapter we develop improved methods to optimize different kinds of routing in IP networks. In particular, we focus on the importance of an adequate objective function used in the optimization, which is one of the main influence factors on the quality of the resulting routing.

In the first part we describe improvements to our existing heuristic network optimization tools and determine the results of network optimization for failure-free IP routing and under certain failure-conditions. We discover that the considered objective functions improve the network in different ways. Routing optimized with one objective function usually leads to a deterioration of other objective functions. This may even lead to results which are worse than unoptimized rout-

ing for certain objective functions. For the COST239 network, an unoptimized configuration results in an Fortz-function value of $\Phi_\emptyset = 0.458$. This value represents the currently experienced total load in the network and should be minimized. After optimization with respect to only the maximum link utilization, the Fortz-value increases to $\Phi_\emptyset = 0.465$. This is due to an increased average path length and an uneven distribution of traffic among the links. A combined optimization method as discussed in this chapter can alleviate this and allows a reduction of the Fortz-value to $\Phi_\emptyset = 0.425$ without impairing the primary optimization value.

The second part of this chapter extends our analysis to loop-free alternates (LFAs) as an IP fast reroute mechanism, which provides fast recovery after specific network failures. We assess the applicability of LFAs for different use cases and show its potential and its drawbacks. We use routing optimization to enhance the utility of LFAs for certain applications. An important result is that the selection of an appropriate objective function which suits to the application improves the applicability of LFAs. It was shown that for certain use cases, the amount of protected traffic can be increased from 15.2% to up to 69.4% by using the newly proposed objective function $\pi_{\text{ND}}^{\text{full}}$. Another key finding is that the optimization of almost any LFA-coverage metric leads to intolerably high link utilizations in the network. When routing is optimized to minimize link utilizations, the coverage-rate of LFAs deteriorates. For a specific scenario, we may achieve on the one hand an optimized routing where only 0.6% of traffic is lost during failures at the cost of a maximum relative link utilization of 140%, which represents a severe network overload. On the other hand, a different optimization can bring down the maximum link utilization to a much lower load of only 73%, but also to an average traffic loss of 3.2% during failures. By using our proposed Pareto-optimization, a result can be achieved that has maximum link utilizations lower than 95% while losing on average only about 1% of traffic during failures due to missing LFAs. The resulting set of Pareto-optimal link costs can be used by a human operator to choose the trade-off between the two optimization goals that works best for his network.

In the third part, we study the pre-congestion notification protocol PCN, the optimization potential that comes with it, and different options how it could be implemented. One class of the investigated methods requires dual markings (DM) and can support two different thresholds, while the simpler implementation with a single marking (SM) defines that *SR*-thresholds must be a fixed multiple of the *AR*-thresholds for all links in the PCN domain. In the examined topology, we show that simple assignment of thresholds leads to a setting where SM-PCN can carry only 4.4% of the traffic that can be supported by DM-PCN. We develop more complex algorithms and the results show that single marking still yields significantly less resource utilization than dual marking. With new objective functions for routing optimization, we are able to improve the admissible rate of protected high-priority traffic for PCN by 20% - 40% for both mechanisms. All in all, we showed that with optimized thresholds and routing, the dual marking architecture can carry 50% - 100% more protected traffic than the single marking architecture. To conclude, the additional overhead introduced by the DM mechanisms allows a performance improvement which clearly outperforms the SM mechanism. Thus, specific implementations for the usage in core networks should rely on the DM mechanism, if possible.

These configuration and optimization results and the provided analysis enable researches to improve the design of current and future routing, resilience, and admission control mechanisms and, in addition, help network administrators to configure and optimize IP routing for their networks in practice.

3 Design of a New Addressing and Routing Protocol

Though this be madness, yet there is method in't.
(William Shakespeare, Hamlet)

In the previous chapter we optimized existing protocols and mechanisms that can be used to improve the routing inside an autonomous system. Each network can individually profit from these improvements. The Internet is a connection of an ever increasing number of heterogeneous networks, which, as a whole, build the nervous system of today's modern society. The enormous success resulted in an unexpected growth and ever changing demands, which entail many challenges that cannot be solved locally, but must be managed on a global scale.

The original Internet was designed to interconnect a small number of hosts, which were attached to a growing but conceptually rather static network topology. This is no longer the case since today all sorts of devices are interconnected over the Internet protocol (IP). Many Internet service providers (ISPs) compete for customers causing modifications of the logical topology with each customer change, and mobile devices require fast support by local networks when moving from one network to another. To meet the changing requirements, several mechanisms have been adjusted or added. Examples are the domain name system (DNS) that has been introduced to decouple names from addressing, the border gateway protocol (BGP) to make routing more scalable, the transmission control protocol (TCP) to avoid a congestion collapse on the data plane, or mobile IP to accommo-

date moving users. The Internet was always subject to changes that were pushed by the insight that its future operation was at risk or at least its further expansion was hampered [158].

This chapter analyzes impending challenges for the future Internet and summarizes ideas how they could be addressed. We combine and extend different ideas and propose a new naming and routing architecture for the future Internet to overcome the discovered challenges. In the first part of the chapter we look at different facts that restrain future growth of the Internet. One of the most challenging issues is the scalability problem. The Internet is constantly growing, not only in terms of users and networks, but also in terms of interconnections between these entities. This leads to an ever increasing number of entries in the routing tables of today's routers [159]. Resilience requirements and traffic engineering generate additional entries. The expansion of the current IPv4 Internet is at its limits as the pool of free IPv4 addresses is already exhausted [160]. The introduction of IPv6 with its huge address space aggravates the situation even more as its deployment gains momentum. There might come a time where routers will no longer be able to handle this unlimited growth [161]. A future Internet architecture must account for this growth and provide mechanisms that make routing more scalable, at least for the IPv6 Internet. We show this development and point out other requirements for a future Internet routing architecture. Then, we give an overview over different options how a new architecture could be designed.

A promising approach is the separation of current IP addresses into two independent pieces of reachability and identification information. This is called locator/identifier split (Loc/ID split) [162] and helps to reduce the routing table growth. The stable identifier (ID) gives a global name to a node. A changeable locator (Loc) describes how the node can currently be reached through the global Internet. A mapping system (MS) is required to map locators to identifiers. This principle makes routing in the stable Internet core more scalable because core routing is not affected by changed attachment points and multihoming of edge networks. The deployment of Loc/ID split in the Internet requires modifications to the current routing and addressing architecture. Its development takes a long

time and implies hardware and software upgrades. Therefore, the modified Internet architecture should also satisfy additional requirements like support for renumbering, multihoming, multipath transmission, and security [163, 164].

Many of the proposed architectures have their drawbacks or do not address all issues that we consider important. Thus, we propose our own architecture for future Internet routing and addressing, which implements a global locator, local locator, and identifier split (GLI-Split). It splits the functionality of IP addresses into identifiers and two different locators, a global and a local one, and implements a true Loc/ID split with IDs that are independent of the current location. To complete the architecture, we also propose the future Internet routing mapping system FIRMS, which can be used as a locator-to-identifier mapping service, independent from GLI-Split.

This chapter is organized as follows. Section 3.1 explores the design-space for new naming and routing architectures. In Section 3.2 we propose the new GLI-Split architecture. Section 3.3 proposes the mapping system FIRMS, which can be used to support GLI-Split and many other architectures. Finally, Section 3.4 summarizes some condensed insights.

3.1 Future Internet Routing: Motivation and Design Issues

In spite of BGP, interdomain routing does not scale anymore. Too much information needs to be exchanged between border routers, their routing tables become larger and larger, and it is not clear whether router technology can keep pace with the growth of the routing tables and increased traffic volumes in the future at reasonable costs. Therefore, operators and router vendors have already recognized the need for a new change of the interdomain routing system and the Routing Research Group (RRG) of the Internet Research Task Force (IRTF) [165] provides a forum to discuss problems and proposals.

This section gives an introduction into the problems and summarizes some strategies to overcome them. The content of this section is mainly taken from [9] and it is structured as follows. Section 3.1.1 briefly reviews routing in today's Internet and Section 3.1.2 explains why it does not scale. Section 3.1.3 describes ideas to decelerate the growth of the routing table sizes in today's Internet architecture. Section 3.1.4 presents the locator/identifier split principle, interworking issues, design options, and an overview over proposed architectures.

3.1.1 Routing in Today's Internet

The Internet is an interconnection of multiple autonomous systems (ASes) using IP as the common base to exchange messages. IP networks use destination-based forwarding, routers look up the next-hop for a packet in their forwarding information bases (FIBs), which are derived from their routing tables. The FIB entries consist of address prefixes and next-hops. The longest prefix match for a destination address determines the interface over which the packet is transmitted. A default route can be provided that is taken when no matching prefix is found.

As we explained, analyzed and optimized in Chapter 2, each AS uses its own method to generate entries in the routing tables. They assign administrative costs to all links within the AS and forward the traffic along least-cost paths. This is usually realized by distributed routing protocols like OSPF or IS-IS. For larger ASes, a subdivision of the network into several routing areas helps to manage the routing complexity and to keep intra-domain routing scalable.

To reach nodes in other ASes, inter-domain routing uses the border gateway protocol (BGP). Each BGP router tells its neighbors which destination prefixes can be reached over its own network and also provides a list of ASes that need to be traversed on the path towards the destination AS. Therefore, BGP is called a path vector protocol. Routers in edge networks usually have a manageable number of prefixes in their routing tables and packets to unknown destinations are forwarded to a default router. However, BGP routers in the core of the Internet do not have default routes. They constitute the so-called default-free zone (DFZ) of

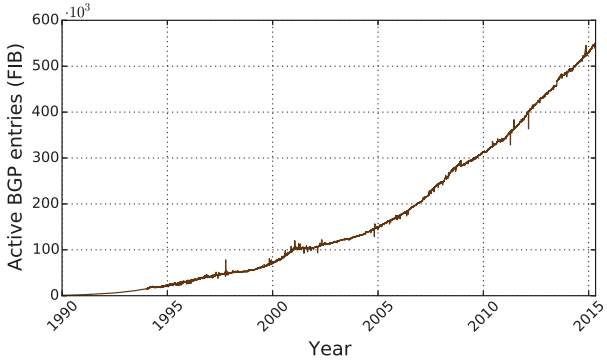
the Internet. The DFZ routers need an entry for each prefix that should be reachable in the Internet and, as a consequence, their routing table size increases with the number of reachable prefixes.

The early Internet consisted of a relatively small number of ASes and customer edge networks, which were only sparsely connected. Each AS was assigned a rather large chunk of the available IPv4 class A, B, or C addresses. The corresponding class prefixes were announced individually into the interdomain routing system. It soon became evident that this practice leads to address exhaustion because most of the assigned class A and B address space is never used. The address space of a class C network is often too small for a company or institution so that they need several of them. This heavily burdens the routing tables since each of the many (about 2 millions) and long class C prefixes requires a separate entry in the routing tables of the DFZ. The problem was alleviated by the introduction of classless interdomain routing (CIDR) in 1993 which removed the strong classification into class A, B, and C addresses. CIDR allows IP address assignment on a more fine grained level, i.e., the address space of a single class A or B address can be assigned in small portions to various customers. In addition, prefix aggregation is possible, i.e., ISPs announce only one short prefix to BGP instead of multiple longer prefixes when these prefixes cover a contiguous address block.

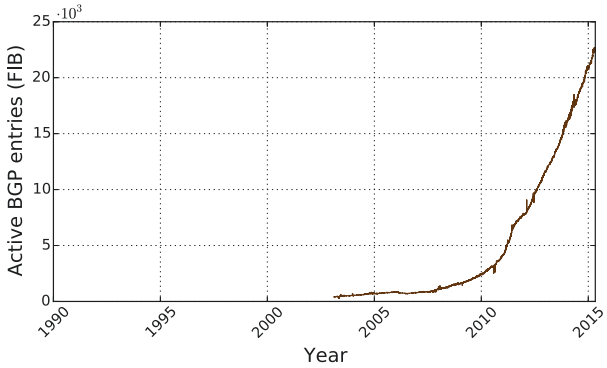
3.1.2 The Scalability Problem

Currently, we observe that the number of entries in the routing tables of the DFZ is increasing at an alarming rate. Figure 3.1(a) shows the growth of active entries in the IPv4 BGP forwarding table. To cope with larger routing tables, routers need to be more powerful. Advances in routing technology might be able to compensate the increased routing tables, but this can only be achieved at disproportionately high costs.

When the Internet finally runs out of IPv4 addresses, the introduction of IPv6 eventually brings an almost unlimited number of IP addresses. This solves the



(a) Active IPv4 FIB entries in the AS 65000.



(b) Active IPv6 FIB entries in the AS 2.0.

Figure 3.1: *Growth of the routing tables in the DFZ [166].*

problem of address depletion, but routing tables are going to grow even more because the vast amount of available IPv6 addresses leads to many additional prefixes to be announced in the DFZ. Figure 3.1(b) shows that the number of active entries in the IPv6 BGP forwarding tables is still relatively small, but it is increasing rapidly. Furthermore, IPv6 addresses are 4 times bigger than IPv4 addresses and, thus, require more of the scarce fast memory resources on Internet routers. A thorough analysis of this problem at the IAB workshop on routing and addressing [161] yielded the following conclusions. The main causes for the current growth rate of the routing tables in the DFZ are the use of provider-independent addresses, multihoming, traffic engineering for edge networks, and countermeasures against prefix hijacking. In the following, we explain these issues in more detail.

Provider-independent addressing The IP address space can belong to providers or to customers. In the first case, the addresses are called provider-aggregatable (PA). The provider rents subspace, i.e. prefixes, to customers for the duration of their contract, but remains the owner of the IP addresses. When the contract is over, the provider rents the prefixes to other customers. This has no impact on interdomain routing because packets to theses prefixes are still routed into the same AS. PA addresses limit BGP change rates and the fragmentation of the address space, i.e., they preserve the aggregation of IP addresses so that short prefixes continue to be announced through BGP. However, when a company using PA address space changes its provider, all computers and devices in that company must be renumbered to the address subspace of its new provider. This is a time-consuming and expensive task. Hence, companies prefer to obtain their own address space, so-called provider-independent (PI) addresses. This allows them to easily change providers without renumbering. For the global routing system, a provider change for PI addresses means a BGP update. In addition, the moved prefix possibly cannot be aggregated with other addresses in the AS of the new provider and needs to be announced separately to BGP which increases the number of entries in the interdomain routing tables.

Multihoming for increased reliability Customers like to be connected to more than one ISP to increase the reliability of their Internet connection. In case that the connection to one ISP fails, their traffic can be switched to the other ISPs. This requires that different paths towards these prefixes need to be announced to BGP to make the customer network reachable over multiple providers. This leads to several entries for a single prefix in the BGP routing tables.

Multihoming for traffic engineering Customers with PI addresses may wish to use different providers for service differentiation. They subdivide their address space into smaller chunks each of them serving a different purpose and being attached to a different provider. In addition, multihoming may be used for load balancing purposes. As a result, the address space is split and several longer prefixes are announced via different providers to BGP.

Countermeasure against prefix hijacking IP's destination-based forwarding uses the longest prefix match principle. When several prefixes in the FIB match the destination address of a packet, the packet follows the route specified for the longest prefix. Malicious ASes may inject prefixes they do not own. If they are more specific than other prefixes, they attract the traffic. To avoid this risk, ASes like to announce the longest possible prefixes which are 24 bits long at least for most important services such as DNS. This also leads to an increase of the routing table sizes in the DFZ.

3.1.3 Tuning BGP and Simple Overlays

The size of the Internet, its inevitable changes, and failures lead to a large rate of BGP update messages stressing router CPUs. As this rate is increasing, many proposals have been made to modify BGP in order to reduce it [167–170].

The current BGP system cannot be adjusted to be truly scalable in terms of routing table sizes. Routing schemes with preferably logarithmic scalability in network size are desired to allow for almost unlimited future growth of the

global Internet and a lot of research has been done on that topic. Unfortunately, it has been shown that logarithmic scaling on Internet-like topologies is impossible in the presence of topology dynamics and/or topology-independent addressing [171].

Any routing mechanism replacing BGP, possibly on a flag day, is almost impossible to deploy in the widely distributed Internet. Therefore overlay architectures have been proposed which leave the current BGP system in place, but take most of the load away from it. They shrink the routing tables to make interdomain routing more scalable. We explain two fundamentally different ideas for that purpose.

Aggregation proxies With aggregation proxies, ISPs announce some of their supported prefixes not via BGP, but only to special aggregation proxies. An aggregation proxy receives many long prefixes and announces aggregated and shorter prefixes to BGP. Packets in the DFZ are carried to this aggregation proxy, which tunnels them to the ISPs that announced the longer prefix to the aggregation proxy. The aggregation proxy in Figure 3.2 receives the long prefixes X.Y.0/24, X.Y.1/24, X.Y.2/24, and X.Y.3/24 and announces the prefix X.Y.0/22 to BGP. Therefore, it receives the traffic addressed to these destinations and forwards it to them over a direct tunnel to the border router of the corresponding networks. In our simple example, the routing table size in the DFZ is reduced by 3 entries. However, packets in the DFZ destined towards the long prefixes are always carried via an aggregation proxy. This kind of triangle routing possibly leads to longer paths compared to the shortest AS-path or the normal BGP path. Thus, path prolongation is the cost of aggregation proxies. Furthermore, networks should be customers of aggregation proxies or peering partners. Hence, this concept also requires substantial economic support for effective deployment. Several aggregation proxies may exist for the same prefix. The Core Router-Integrated Overlay (CRIO) implements this concept and [172] gives insights into tradeoffs like routing table size reduction vs. path length prolongation and many more.

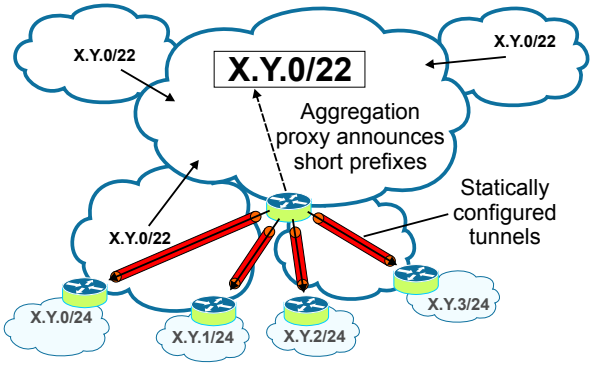


Figure 3.2: *The aggregation proxy announces a short prefix instead of many long prefixes. Packets addressed to the long prefixes are routable in the DFZ, but are forwarded to the aggregation proxy which tunnels them to their destination network.*

Lookup system for nonroutable prefixes Another concept is to retain long prefixes from BGP and to record them in a DNS-like lookup system together with a router having a routable address and being part of the destination AS. As a result, the long prefixes are not routable in the DFZ, but the lookup system knows a router from which corresponding packets can be forwarded without interdomain routing information. This concept is depicted in Figure 3.3. If a router in the DFZ does not find an entry for the destination address of a packet in its routing table, it queries the lookup system for a tunnel endpoint into the destination AS and forwards the packet over that tunnel. After decapsulation in the destination AS, the packet can be forwarded via intradomain routing. The tunneling route reduction protocol (TRRP) [173] implements this idea. It requires the introduction of a mapping service, and the DFZ routers must be changed to perform the lookup and tunneling.

The presented methods leave today's routing system and in particular the meaning of the IP addresses basically as they are, but they still require changes to the Internet that are hard to deploy.

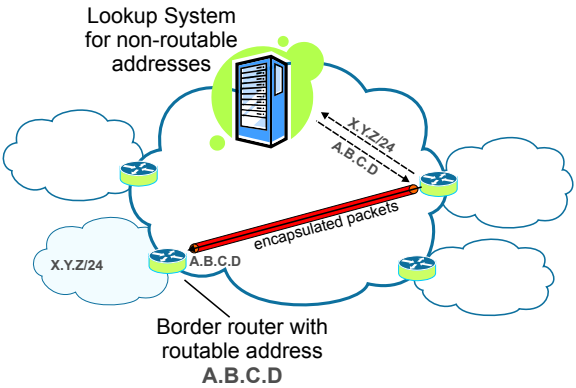


Figure 3.3: *Some long prefixes (e.g. X.Y.Z/24) are not announced to BGP. Therefore, they are not routable in the DFZ. The lookup system provides a router with a routable address for them in the destination AS. Then, packets with non-routable addresses can be tunneled to and decapsulated by such a router, and forwarded from there to their destination via intradomain routing.*

3.1.4 The Locator/Identifier Split

The locator/identifier split (Loc/ID split) is a concept that is similar to the previous approach, but it does not require an upgrade of the DFZ routers. It has been implemented by various proposals using different nomenclature and technical realizations. We provide the inherent motivation for using a Loc/ID split as solution to the scalability problem. Then we explain the actual concept in a general way and discuss interworking issues and design options. Finally, we look at implementation details of current proposals.

An analysis of the current Internet revealed that a complete redesign of the routing architecture may be needed [161]. The current approach cannot scale, because the IP addresses have two different functions. An IP address determines the position of a node within the Internet topology, i.e., it serves as *locator* for

routing purposes. In addition, it also serves as node *identifier*. The scalability of interdomain routing is based on hierarchical structures since an ISP can aggregate many long prefixes and announce them as a single short prefix. To take advantage of that principle, locators must be assigned according to the topology and should change only if the topology changes. In contrast, end users see IP addresses as identifiers and prefer to keep them if they move or change providers. This observation leads to the conclusion that a scalable Internet architecture should separate core and edge networks [174] in terms of routing and address space by splitting locator and identifier into different entities. This way, locators can easily be changed without renumbering the devices in the network of a customer when he moves to a different provider. Identifiers should be used only to give names to devices [175]. By removing the edge network prefixes from the routing tables in the DFZ, dynamic changes in the routing space of edge networks are hidden and remain local. The routing tables in the DFZ then grow with the small number of core networks and not with the number of edge networks. When a customer changes its provider and connects its network to a different gateway, it can keep all the identifiers in its network as they are and there is no need for renumbering. Only the mapping service needs to be updated about the locator change. This way, both core and edge networks profit from a separation of locators and identifiers.

The separation principle requires a new addressing and routing architecture which can be realized in many different ways.

3.1.4.1 Architectural Design Options

A Loc/ID split architecture is a two-level routing architecture. Figure 3.4 shows that there is only a single upper layer domain (global routing domain), but there are many local routing domains (intradomains) in the lower layer. Routing in the upper layer is based on locators while routing within the lower layer can be based on the identifiers. As long as communicating entities are in the same local routing domain, only their identifiers are needed to exchange messages. Communication between entities in different local routing domains is more complex. Local routing domains have gateways towards the global routing domain. These

gateways are part of the global routing domain and have own locators. A packet with a destination identifier outside the local routing domain is forwarded to such a gateway. The source gateway determines the locator of the destination gateway by some mapping service and adds this locator to the packet. Then, the packet is carried through the global routing domain according to the locator to the destination gateway. The destination gateway strips off the locator and the packet is forwarded according to its destination identifier.

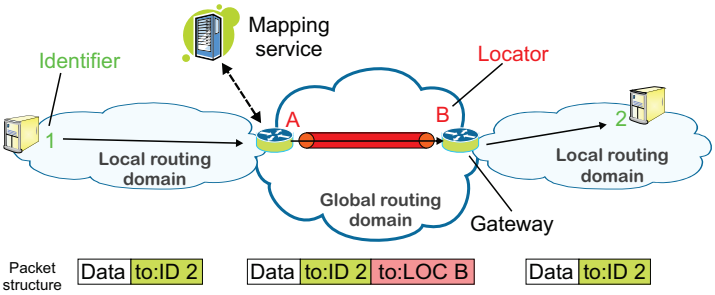


Figure 3.4: *Loc/ID split is a two-level architecture. Packets are forwarded within a local routing domain according to their identifier. Within the global routing domain they are forwarded according to a locator provided by the gateway when leaving the local routing domain.*

Location of separation A distinctive feature of a separation architecture is the location *where* the separation of core and edge addressing takes place. In a host-based architecture, each participating host performs the separation. Packets on the transport layer are addressed with identifiers and the protocol that handles the separation adds locators before the packets leave the host. On the network, only locators are used to forward packets. In a network-based architecture, as shown in Figure 3.4, identifiers are used both on transport and network layer inside a host, while the translation from identifiers to locators is done inside the network at a middlebox, e.g., a border router. In mixed architectures, both op-

tions are used and there is a separation at the host and at a network middlebox. The location of the separation and, thus, the entity that requires mappings from identifier to locator has direct influence on the design of the mapping system.

Design options for Loc/ID split transport The locator/identifier split does not stipulate a special technique for the source gateway to add a locator to a packet. There are three major alternatives that can be used: encapsulation, address rewriting, and source routing.

In case of encapsulation, the source gateway tunnels the packet addressed to the destination identifier in another packet addressed to the destination gateway locator, and the destination gateway decapsulates the packet. This is known as the map-and-encap paradigm [161]. Although the overlay solutions in Section 3.1.3 also use tunneling, they are significantly different as they do not implement the Loc/ID split approach. In case of address rewriting, there is a 1:1 mapping from local addresses (identifiers) to global addresses (locators). Gateways replace the local source and destination addresses of outgoing packets by their global addresses, and for incoming packets the reverse operation is performed. Note that this kind of address rewriting is different from today's network address translation (NAT) for private networks. It is stateless and, therefore, relatively simple. In particular, nodes within a local routing domain are reachable from outside by their global addresses. A provider change implies a locator change which results in a modified global address, but renumbering of the nodes in the affected local routing domain is not needed. The last option is source routing where locator and identifiers are encoded by the source host in the destination address. In this case, no modifications of packets along the path to the destination are necessary.

Packet en- and decapsulation costs CPU cycles on routers and encapsulation can cause problems with maximum transfer units (MTU) and/or packet fragmentation. Address rewriting can also be expensive. Depending on the implementation, an additional header is added to record the identifiers. In IPv4, additional headers cause packets leaving the fast-path of a router and being processed by the router's CPU which can severely impact its performance.

Design options for the mapping service There are many design options to implement the mapping service, which are classified in more detail in Section 3.3.5. A mapping service can be a single server or a server overlay where each server keeps the locator-identifier mapping for the entire identifier address space [176]. However, this information may also be partitioned among many servers taking advantage of some hierarchical structure in the identifier address space to facilitate effective information retrieving [177, 178]. As an alternative, a distributed hash table may be used for that purpose [179].

Communication overhead between the source gateways and the global mapping service must be kept at a minimum when locators are queried for every identifier. A local cache can help to answer queries immediately without consulting the mapping service [180]. When an intermediate router encounters a cache miss and has to wait for the answer to a mapping query, it can either store and delay packets, or simply drop them. To make this a rare event, the cache size must be large enough. In the extreme case, the local cache is a copy of the mapping information for the entire identifier address space [181]. A cache miss usually occurs for the first packets of a communication. Their number is relatively small, but it adds delay when some of them carry important signaling information such as a TCP SYN. Therefore, some mapping services have packet forwarding capabilities [177]. Initial packets causing cache misses are encapsulated and sent to the mapping service. The mapping service retrieves the locator, and sends it to the source gateway, while at the same time it acts as a proxy for the source gateway by adding the locator to the initial packets and forwarding them to the destination gateway.

The mapping service is a vital element of the Loc/ID split architecture, which needs to be up to date to achieve global reachability for identifiers. This becomes more difficult with caches because the information stored in caches may be obsolete. Both push and pull architectures are currently discussed. Either the mapping service triggers the update of local caches or the gateways are pull updates from the service. Hybrid mechanisms push the mapping information to a set of cache servers from where it can be pulled quickly to any point in the Internet.

3.1.4.2 Advanced Features and Properties

It is important to decide, which advanced mechanisms or services should be natively supported by the architecture that are not directly required for the plain architecture to work. This could be for example support for interworking, mobility, multipath routing, or traffic engineering as well as built-in security. All these options have direct influence on the design of the architecture and may influence each other so that a multitude of different scenarios has to be considered. Hence, care has to be taken when choosing the right combination of design options.

Mobility Mobility requires that nodes can move from one network to another while being reachable and without changing IP addresses on the transport layer. The latter is important for the maintenance of TCP connections. In theory, this could be achieved with the Loc/ID split, but this implies two major challenges. First, updates of the mapping system must be fast enough, and/or the gateways must implement handover functionality. Second, the IP addresses of the visiting nodes must be routable in the visited domain. The identifier space usually reflects some structure of the local routing domain to improve the routing scalability. Therefore, it is rather hard for the routing in the visited local routing domain to quickly integrate the address of the visiting node. This is due to the fact that identifiers are also used as local locators.

Traffic engineering Apart from the improved flexibility, the Loc/ID split also opens new possibilities for interdomain traffic engineering [162]. Local routing domains can be multihomed and have several gateways and locators. This entails degrees of freedom for load balancing supported by the mapping service.

Interworking with the Internet We assume that the current Internet evolves to the future upper layer domain, but it is certainly also possible to think of the Internet being one of many local routing domains in the future Internet. For interworking with the Internet, identifiers and locators should be IPv4 or IPv6 addresses. Here, we do not distinguish between IPv4 and IPv6 and think of one

common IP address format. The address spaces of locators (including the Internet) and identifiers must be disjoint to avoid ambiguities between devices in the current and the future Internet.

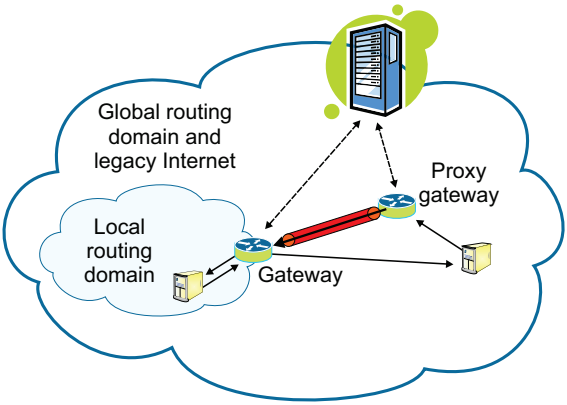


Figure 3.5: *Nodes in a local routing domain communicate directly with legacy nodes as the gateway just forwards the packets. In the reverse direction proxy gateways attract traffic with destination in local routing domains and add locators to guarantee that they reach the correct local routing domain.*

A simple solution uses proxy gateways in the upper level domain, i.e., in the plain Internet. They announce all identifier prefixes into BGP. Therefore, the identifier address space should be aggregatable to avoid additional significant growth of the routing tables. When a plain IP node sends a packet to a node in a local routing domain, it can use the identifier of this node as destination address. Then, the packet is carried to a proxy gateway, which requests the locator for the destination identifier from the mapping service and adds the destination locator to the packet. Hence, the proxy gateway performs the same operation as a common source gateway. Then the packet is eventually delivered to the correct destination gateway where the destination locator is removed. From there the packet is forwarded to its destination within the local routing domain using only the

destination identifier. Communication from a node in the new part of the future Internet to the plain old Internet is even simpler. A packet from a local routing domain is addressed to a node in the plain Internet. As the destination address is not part of the local routing domain, the packet is forwarded to a source gateway. The gateway realizes that the address is part of the plain Internet and, therefore, the packet can be forwarded in the upper layer domain without any modification. Proxy gateways in this context and aggregation proxies used for BGP tuning look similar, but there is a subtle difference. While proxy gateways can basically attract all traffic addressed to globally non-routable identifiers, aggregation proxies attract only traffic whose prefix lies within their aggregation ranges.

The other solution for interworking with legacy networks is network address translation. A node of a local routing domain in the future Internet can send a packet addressed to some node in the plain Internet. The gateway detects this and performs NAT. More specifically, it translates its source identifier to a locator address belonging to the gateway and forwards the packet to the destination in the plain Internet. Potential answers are returned to the gateway which translates its own locator in the address field of the packet to the initial source identifier and forwards the packet to the local routing domain. Communication in the reverse direction can be more complex, depending on the actual implementation.

3.1.5 Related Work on Loc/ID Split

There is a multitude of architectures for the future Internet and in particular for a new routing and addressing architecture, trying to solve the scalability problem of today's Internet [182]. In the following, we review prominent architectures in this research area, show which design options have been chosen, and address some of their shortcomings.

There are several proposals that are intended to be incrementally deployed in the Internet. The Locator/Identifier Separation Protocol (LISP) [183] is the most prominent of them. It is currently being standardized by the Internet Engineering Task Force (IETF) [184] and pilot networks already exist. LISP divides the IP

address range into two subsets. Endpoint identifiers (EIDs) identify end-hosts on a global scale and are used to forward packets inside LISP domains. LISP domains are edge networks that are connected via LISP gateways to the core of the Internet. In the Internet core, only globally routable addresses are used to forward packets. They are called routing locators (RLOCs). The communication between LISP nodes inside the same LISP domain works like in today's Internet. However, the communication between LISP nodes in different LISP domains requires tunneling. The LISP node in the source domain addresses a packet to its destination EID. The packet is forwarded to the LISP gateway which then acts as an ingress tunnel router (ITR). It queries the mapping system for the RLOC of the gateway that belongs to the LISP domain hosting the destination EID given in the packet.

The mapping system returns the desired EID-to-RLOC information to the ITR, which then encapsulates the packet towards the obtained RLOC and sends it. The ITR stores the mapping in its local cache to avoid another query for the same EID. The gateway of the destination LISP domain receives the tunneled packet and acts as an egress tunnel router (ETR). It just strips off the encapsulation header and forwards the packet according to the destination EID which is routable in the destination LISP domain. Interworking with the plain IPv6-Internet may be done using complex stateful NAT or proxy gateways [185]. LISP-nodes can send packets directly to plain IP-nodes in the Internet outside LISP-domains. When a plain IP-node sends packets to a node within a LISP-domain, the packets are forwarded by default to a proxy router in the Internet which looks up appropriate global locators and tunnels the packets to the destination LISP-domain using this locator information. Proxy routers have two major disadvantages. First, traffic cannot take the shortest AS-path but takes a detour via the proxy (triangle routing). Second, they attract and forward large data volumes and it is not clear who pays for it. Similar interworking solutions exist also for other map-and-encaps proposals.

IVIP [186] is very similar to LISP with extra features for the handling of mobile users. APT [176] adds features to the mapping service in order to provide

protection mechanisms in case of network failures.

Six/One Router [187] is a different proposal and uses address translation instead of tunneling. The gateways are called Six/One routers. The node identity internetworking architecture [188] is as well based on Loc/ID split. It integrates ideas from the host identity protocol (HIP) [189] to achieve increased security and provides improved mobility support.

The identifier locator network protocol ILNP [190–192] splits the IPv6 address into a locator and identifier part. With ILNP, applications are expected to identify nodes only by fully qualified domain names (FQDNs) and the domain name system (DNS) resolves them to possibly several addresses containing the unique identifier of a node and a locator. The lookup is done by the hosts and no gateway interaction is required. Hosts must be upgraded to take advantage of ILNP since gateways cannot take over partial functions. ILNP has evolved from the early ideas of GSE (global, site, and end-system address elements) [193,194]. GSE essentially codes a global locator, a local locator, and an identifier into an IPv6 address. Addresses are dynamically combined from these parts. It uses only the identifier for TCP checksum calculation and requires host upgrades for deployment.

The hierarchical architecture for Internet routing (HAIR) [195] is a clean-slate approach and does not need address rewriting by border routers. It implements source routing in the sense that the hosts compose destination addresses containing global locator, local locator, and identifier information. This requires host upgrades since hosts need to perform mapping lookups for that purpose.

The host identity protocol (HIP) [196] also implements the Loc/ID split. However, its intention is rather enhanced anonymity, security, and mobility instead of improved routing scalability. It could be used on top of other approaches to combine their advantages.

3.2 Global Locator, Local Locator, and Identifier Split (GLI-Split)

Most of the current proposals for a future routing and addressing architecture [197] implement a kind of Loc/ID split. They essentially separate core and edge routing, but local routing is still performed on IDs. When nodes change their position within a local routing domain or move from one edge network to another, they either require a new ID or the local routing system must account for that change. Replacing a node's ID breaks the function of an ID and adapting the local routing system makes routing more complex. A few proposals implement a true Loc/ID split, e.g. [195], but they take a clean-slate approach, i.e., they are not backward-compatible with today's Internet which makes them hard to deploy.

In this section, we propose GLI-Split as a new concept for future Internet routing and addressing. It splits the functionality of IP addresses into global locators, local locators, and identifiers and implements a true Loc/ID split with IDs that are independent of the current location and are never used for routing. IDs and locators are encoded in regular IPv6 addresses so that no new routing protocols are required. GLI-Split is backward-compatible with the IPv6 Internet and interworking is simple. But GLI-Split not only solves the scalability problem, it also has several other benefits. It facilitates provider changes, renumbering, multihoming, multipath-routing, traffic engineering, and provides improved mobility support. To take full advantage of all features, nodes in GLI-domains require upgraded networking stacks, but legacy nodes can also be accommodated in GLI-domains and enjoy benefits. This facilitates incremental deployability. For interworking between GLI-Split and the IPv4 Internet we can use existing mechanisms for IPv4 and IPv6 as GLI-Split is IPv6-compatible.

The content of this section is mainly taken from [10]. It is structured as follows. Section 3.2.1 presents fundamentals of GLI-Split and explains how GLI-Split works with upgraded and non-upgraded nodes in single-homed domains. Section 3.2.2 then introduces mechanisms required for the multi-homed case and describes how multipath transfer and traffic engineering can be supported. Sec-

tion 3.2.3 briefly introduces our implementation of the GLI-Split architecture and presents a proof-of-concept evaluation with respect to the round-trip-time. Finally, we summarize the benefits of GLI-Split and discuss deployment considerations in Section 3.2.4.

3.2.1 Fundamentals of GLI-Split

This section introduces the basic nomenclature, shows the GLI address structure, and explains their relation to DNS.

3.2.1.1 General Idea and Nomenclature

Edge networks like those of companies that implement GLI-Split are called GLI-domains while others are called *plain IPv6* domains. Nodes of a GLI-domain are GLI-nodes and its border routers are GLI-gateways. GLI-nodes with a special GLI-(networking-)stack are called *upgraded* while others are called *plain IPv6* nodes. GLI-nodes and GLI-gateways are identified by a globally unique identifier (ID). They have a local locator (LL) that describes their position within their GLI-domain and serves for local routing. Furthermore, each GLI-gateway has a globally unique global locator (GL) that describes its position in the IPv6 backbone. A global mapping system (MS) maps IDs to global locators and a local mapping system maps IDs to local locators. This setting is illustrated in Figure 3.6. IDs are denoted by integral numbers, local locators by lowercase letters, and global locators by uppercase letters. In the examples of this section, we refer to parts of the setting in this figure. We designate GLI-nodes by their IDs, i.e., node 1 is the node with ID 1.

3.2.1.2 GLI-Addressing

GLI-Split encodes ID and locator information in IPv6 addresses to be compatible with plain IPv6. The ID of a GLI-address is fixed, while the locator information can be replaced by GLI-hosts and -gateways on the path between source and

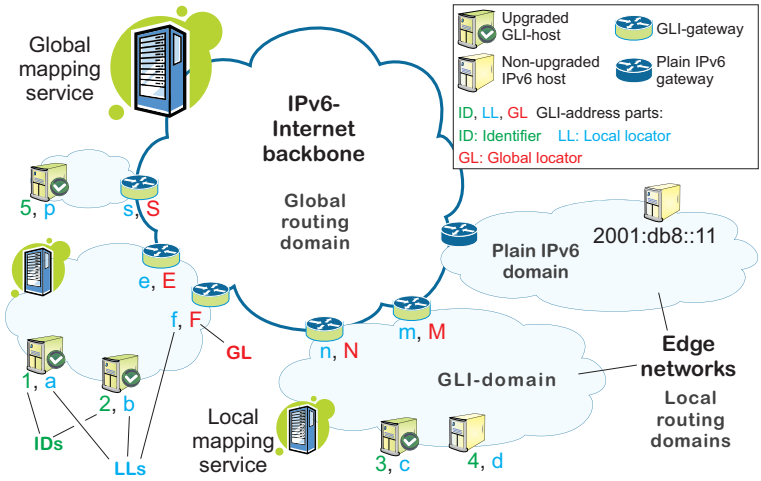


Figure 3.6: *GLI-nodes and GLI-gateways have an identifier (ID) for identification and a local locator (LL) for routing in edge networks; in addition, GLI-gateways have a global locator (GL) which is used for routing in the IPv6 backbone.*

destination. According to the included locator, we distinguish three types of addresses: identifier addresses, local addresses, and global addresses.

Transport layer protocols like TCP use source and destination IP addresses to map packets to flows. During an ongoing transport connection, these elements must not change; otherwise, packets cannot be mapped correctly to the existing connection. An *identifier address* is an endpoint identifier, independent of any locator information, which is used in the transport layer of upgraded GLI-nodes. This guarantees that regardless of the current location of the upgraded GLI-node, the transport layer sees the same address. On the network layer, locator addresses are used to encode the current location. An upgraded GLI-node translates between both address types when handing data up or down the protocol stack. This

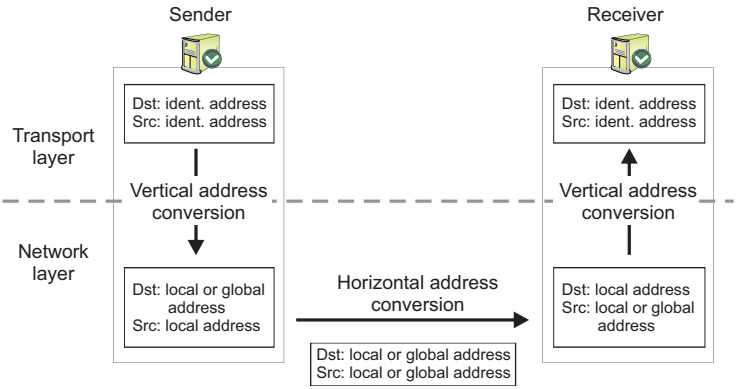


Figure 3.7: *GLI-nodes translate identifier addresses to local or global addresses when handing data from the transport layer to the network layer and vice-versa.*

principle is called vertical address translation and illustrated in Figure 3.7. Depending on the scope of the locator address, we distinguish between two different types. A *local address* is used for forwarding within a GLI-domain. As the local locator has only site-local meaning, a local address must never leave a GLI-domain. To ensure this requirement, a responsible GLI-gateway at the border of the host's GLI-domain translates between local addresses and global addresses. This property is called horizontal address translation and can also be seen in Figure 3.7.

A *global address* is mainly used for routing outside GLI-domains. The global locator belongs to a GLI-gateway of the host's GLI-domain and is allocated from the address space of the ISP that is connecting the GLI-domain to the Internet. Inside a GLI-domain, packets addressed to global addresses are usually forwarded to a default gateway. However, if the global locator in the destination address belongs to a GLI-gateway of the same GLI-domain, the packet is routed to this GLI-gateway.

Address format Figure 3.8 shows the encoding of the three address types reusing the 128-bit IPv6 address format. The 64 higher-order bits are used for routing and special tasks while the 64 lower-order bits contain an identifier. A similar separation is done by ILNP [198]. All GLI-addresses have a special n-bit GLI-prefix to differentiate them from other IPv6 addresses. Routing is based only on the higher-order bits and our assumption is that appropriate GLI-prefixes are announced in the IPv6 backbone. A marker (L, G) indicates whether the locator part contains a local or global locator. Global addresses have the GL followed by a GAP-bit which is used for multipath-routing, traffic engineering, and interworking (see Section 3.2.2.2). Identifier addresses have the locator field filled with padding zeros.

The remaining 16 bits are used for checksum compensation so that checksums calculated, e.g., by TCP, are still valid after locator changes. TCP uses a 16-bit checksum in its header and includes the source and destination address of the IP header in the computation.

Horizontal and vertical address translation in GLI-Split change source and destination addresses. When two GLI-nodes communicate with each other, checksum problems do not occur as GLI-nodes see only identifier addresses on the transport layer. However, checksum problems possibly occur for communication with non-upgraded nodes because they use locator-dependent GLI-addresses for checksum calculation which are subject to changes. GLI-Split solves this problem by compensating these bit changes with an additional checksum inside the GLI-address (see Figure 3.8). The 16-bits are computed like in the TCP header as the one's complement sum of the preceding three 16-bit words. Therefore, changing a GLI-address does not modify the TCP checksum. This makes translation of GLI-addresses invisible to TCP checksum operations.

Address assignment Global locators are IPv6 prefixes that are globally assigned to GLI-gateways from ISPs in a hierarchical way, just like regular IPv6 prefixes are assigned today in the Internet. IDs are also hierarchically assigned in a similar way, but they are independent of any routing information. The hierar-

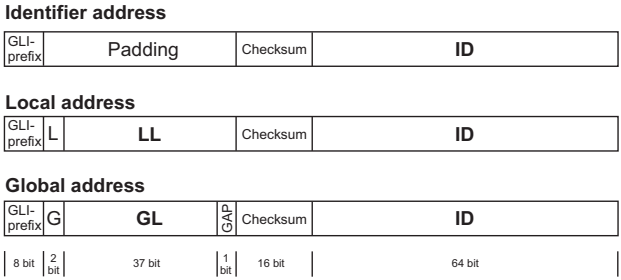


Figure 3.8: *Three types of GLI-addresses are encoded in an IPv6 address.*

chy here is important to improve the scalability of the mapping system, which can then work with ID-prefixes instead of individual IDs. HIP-like IDs may also be supported using the concept in [199]. Local locators are locally allocated according to a network's topology and management needs. They can be dynamically changed and re-assigned to IDs when nodes move within a GLI-domain.

The assignment of local locators to nodes inside a GLI-domain may be done by enhanced DHCP. This DHCP also communicates the information how to reach the mapping service. An upgraded GLI-node knows its ID, tells it to DHCP which returns a local locator as well as a set of global locators. The upgraded GLI-node registers the ID-to-LL and ID-to-GL mappings with the local and global mapping system including the information that the associated node has upgraded GLI-functionality. When an upgraded node changes its attachment point, it performs this procedure again. For non-upgraded nodes in GLI-domains, the assignment process works differently. The DHCP server knows by configuration the MAC address and the ID for every non-upgraded node in its area and assigns a local GLI-address to this node reflecting the ID of the node. Due to missing capabilities of non-upgraded nodes, the DHCP server is in charge of registering the appropriate ID-to-LL and ID-to-GL mapping with the local and global mapping system. As a consequence, stateless address auto configuration [200] cannot be used in this case.

Notation In our examples, local addresses are written as a combination of the local locator (a lower case character) and the ID (an integer number). Example: 'a.1'. Global addresses are written as a combination of the global locator (an upper case character) and the ID. Example: 'E.1'. The activated GAP-bit is denoted by a (g) after the global locator. Example: '$E(g)$.1'. Identifier addresses are denoted only by their IDs. Example: '.1'.

3.2.1.3 Name Resolution

To start a communication session, the initiating host resolves a DNS name (e.g. host3.other-gli-domain.net) into an IP address. If the returned IP address is a GLI-address, GLI-nodes or GLI-gateways possibly require an additional lookup to the mapping system to find an appropriate local or global locator for the ID.

Use of the DNS When a DNS name denotes a GLI-node in a multi-homed domain, it returns a global GLI-address with a set GAP-bit (see Section 3.2.2.2). If the DNS name belongs to a GLI-node in a single-homed domain, the GAP-bit is not set. In both cases, the returned GLI-address is globally routable and hosts outside of GLI-domains can use this address without modifications. This requires no changes to the currently used DNS servers in the Internet. More details about multi-homed domains are provided in Section 3.2.2.

Use of the mapping system The mapping system consists of a local and a global component. The local mapping system stores a set of local GLI-addresses for IDs residing within its local GLI-domain while the global mapping system stores a set of global GLI-addresses for any ID. Sets of addresses are required when routing alternatives exist, e.g., inside a GLI-domain when the ID is connected to several networks in the same GLI-domain, or, in the global mapping system when the ID belongs to a GLI-node in a multi-homed domain.

GLI-hosts with upgraded networking stacks are able to recognize when an IP address returned from the DNS belongs to a GLI-node. In that case, they extract the ID from that address and query the local mapping system for an appropriate

GLI-address. If the destination node resides in the same GLI-domain as the requesting node, the local mapping system returns a set of local GLI-addresses, otherwise it notifies the requesting node that the requested ID is not part of the same GLI-domain. Then, the GLI-node requests the global mapping system which returns a set of global GLI-addresses.

Like the DNS, the mapping system is queried only for the first occurrence of a new ID and the query result is locally cached for later use to avoid that the mapping system becomes a performance bottleneck [180]. We do not rely on any specific mapping system, but in Section 3.3, we propose a scalable, reliable, and secure mapping system called FIRMS which could be used in combination with GLI-Split. We also propose a classification of current mapping systems and compare FIRMS with many others, e.g. [177, 179, 201, 202], in detail.

3.2.1.4 Communication Between Upgraded GLI-Nodes

In this section, we describe the communication between two GLI-nodes with upgraded networking stacks and how networking details are hidden from the transport layer. In the examples, GLI-node 1 establishes communication with another GLI-node with the DNS name hostX.exampleY.net. GLI-node 1 queries the DNS and obtains an IPv6 address. As the prefix of the returned address indicates a GLI-address, GLI-node 1 extracts the ID from that address. We distinguish whether both GLI-nodes are in the same domain or in different domains.

Communication within a GLI-domain GLI-node 1 communicates with GLI-node 2 in the same GLI-domain (see Figure 3.6). Node 1 queries the local mapping system for a local GLI-address of ID 2. As both GLI-nodes are part of the same GLI-domain, the mapping system responds with one or several local GLI-addresses for node 2. Node 1 chooses one of them as destination address and its own local GLI-address as source address for communication with node 2.

Communication between GLI-domains GLI-node 1 communicates with GLI-node 3 in a different GLI-domain (see Figure 3.6). When node 1 queries

the local mapping system for local GLI-addresses of ID 3, it receives a negative answer. Then, node 1 queries the global mapping system for a global GLI-address of ID 3. Alternatively, the local mapping system can forward the request to the global mapping system which returns the global GLI-addresses so that GLI-node 1 needs to issue only a single query. Node 1 uses its own local GLI-address as source address and one of the returned global GLI-addresses of ID 3 as destination address for communication with node 3.

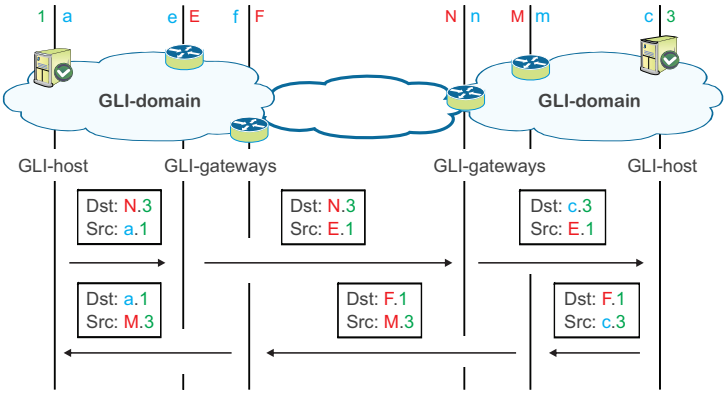


Figure 3.9: *Communication process with horizontal address translation between two GLI-domains. GLI-node 1 sends a packet to node 3 in a different GLI-domain and node 3 replies.*

Figure 3.9 shows how source and destination address fields of IP packets change on the path between GLI-nodes 1 and 3. Depending on the configuration of the local routing system, packets are forwarded either to a default GLI-gateway or to a specific GLI-gateway. We assume that packets are routed to the gateway with global locator E. When a GLI-gateway receives a packet destined to an outbound global address, it substitutes the local source address with the global source address reflecting its own global locator. Then, the packet contains globally routable source and destination addresses. It can be carried over the stan-

dard IPv6-Internet backbone towards the GLI-gateway whose global locator is reflected in the packet's global destination address. The GLI-gateway in the destination GLI-domain queries its local mapping system for a local GLI-address of ID 3 and substitutes the global destination GLI-address in the packet by a local destination GLI-address. Based on the local destination GLI-address, the packet is eventually delivered to GLI-node 3.

When GLI-node 3 sends a response back to node 1, it also queries the mapping system to obtain a global locator of ID 1. When GLI-domains are multi-homed, different GLI-gateways may be chosen. As a result, different global GLI-addresses may be used in the two directions of a single communication session (see Figure 3.9). This example demonstrates that GLI-Split is by design able to utilize different paths for domains with several gateways. However, if access control or filtering devices like firewalls require that packets enter and leave the GLI-domain through the same GLI-gateway, this can be supported by the mechanisms described in Section 3.2.2.2.

3.2.1.5 Interworking Between GLI-Domains and the Plain IPv6 Internet

For future Internet routing architectures, it is extremely important that the communication with the non-upgraded part of the Internet is possible and that it does not imply any restrictions. In the GLI-Split architecture, the interworking between single-homed GLI-domains and plain IPv6 domains is natively supported. In multi-homed networks, some additional mechanisms are required, which are explained in Section 3.2.2.

In the single-homed case, when node 11 (*2001:db8::11*) in the plain IPv6 Internet wants to initiate a communication to the GLI-node 3 in a GLI-domain (see topology in Figure 3.10), it uses its IPv6 address as source address and the global GLI-address of node 3, which was obtained through the DNS, as destination address. According to the global locator N in the global GLI-address, the packet is delivered to the GLI-gateway of the destination GLI-domain. There, the GLI-gateway queries the local mapping system for the local locator of the destination

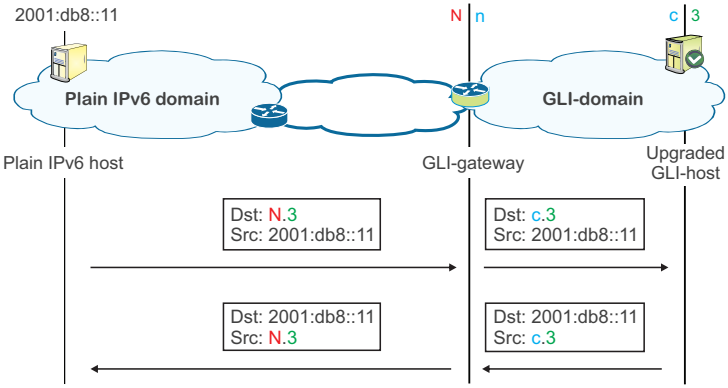


Figure 3.10: *IPv6 node with IP address 2001:db8::11 in a non-GLI domain communicates with GLI-node 3 in GLI-domain.*

GLI-node and replaces the global GLI-address in the destination field with the returned local GLI-address. The packet is then forwarded according to the local locator to the destination node. At GLI-node 3, the local GLI-address on the network layer is replaced by an identifier address before handing the packet up to the transport layer. This is the vertical address translation mechanism which is explained in Section 3.2.1.2. In return packets, GLI-node 3 simply swaps source and destination address on network layer and sends the packet to its default GLI-gateway. There, the local GLI-address in the source field is replaced with a global GLI-address indicating the global locator of the GLI-gateway. After the translation, the packet is forwarded according to the plain IPv6 address in the destination field and eventually arrives at node 11 in the plain IPv6 domain.

In the opposite direction, when a GLI-node initiates a communication to a node in the non-GLI IPv6 Internet, it uses its own identifier address as source and the conventional IPv6 address as destination on the transport layer. The source address is as usual replaced by a local address, but the destination address is not changed when passing the packet from the transport to the network layer.

The packet is carried to a GLI-gateway which then substitutes the local source address by the global address reflecting the global locator of that GLI-gateway. Eventually, the packet is delivered to the destination node. This node can respond to the packet by simply swapping source and destination address. The receiving GLI-node can map the packet as a response to its initial request because the identifier GLI-address and the non-GLI IPv6 address are used on the transport layer.

3.2.1.6 Mobility Support

In today's Internet, mobile IP is needed for communication with a mobile node (MN). The mobile node's home address serves as a stable reference address on the transport layer and for finding a rendezvous point with the mobile node on the network layer. If the mobile node leaves its home network, the mobile node's care-of-address indicates its location on the network layer. With upgraded GLI-nodes, locators in local or global addresses may change due to roaming without breaking transport connections because upgraded GLI-nodes use only identifier addresses on the transport layer so that mobile IP is no longer needed. However, GLI-Split allows improved mobility support only if two upgraded GLI-nodes communicate with each other and reside in GLI-domains. Any other communication patterns are supported by mobile IP.

Mobility support with GLI-Split works as follows. The DNS stores a static home address of the mobile node which is used for mobile IP. This is a GLI-address and contains the identifier in the usual position. Thus, GLI-nodes can extract the identifier and get an appropriate locator for the mobile node. When a mobile node roams into a GLI-domain, it receives new local and global locators and updates the global mapping system and the local mapping system in the new domain. Furthermore, it informs all GLI-upgraded corresponding nodes (CNs) about its new global locator with mobility update messages so that the corresponding nodes can reach the mobile node again. If the mobile node and the corresponding nodes are in the same GLI-domain, the corresponding nodes may query the local mapping system to obtain the local locator of the mobile node

to avoid triangle routing via the GLI-gateway (see Section 3.2.2.3). Then, both nodes communicate via a direct connection without triangle routing. A similar feature is provided by ILNP [198]. In contrast to GLI-Split, in ILNP an upgraded mobile node updates the DNS with its new address when roaming into a new network; interworking with non-upgraded nodes is not defined. In GLI-Split, the new global GLI-address of the mobile node is also used as care-of-address for communication with non-upgraded nodes using mobile IP.

We highlight the benefits of the new mobility support offered by GLI-Split compared to mobile IP. Corresponding nodes can always contact mobile nodes directly without triangle routing over a home agent. This is an advantage since home agents may be far away and increase the latency. With mobile IPv6, such route optimization can be done under some conditions, but the first contact with the mobile node is always via the home agent. Furthermore, GLI-Split makes local moves of mobile nodes almost invisible to corresponding nodes in other domains. If the mobile node moves only within a GLI-domain, it receives a different local locator but keeps the same global locator so that corresponding nodes in different domains can continue to send to the same global GLI-address as before. Hence, the communication is hardly impaired by the location change.

3.2.1.7 Multicast Support

Today, video streaming, IPTV, and content delivery networks account for a large and still growing fraction of the Internet traffic. In this area, IP multicast is gaining more and more popularity, therefore, we explain how IP multicast traffic is handled with GLI-Split. IPv6 offers native support for multicast and requires the Multicast Listener Discovery (MLD) protocol [203] and the Protocol Independent Multicast - Sparse Mode (PIM-SM) protocol [204]. MLD is used by hosts to join or leave a multicast group while PIM-SM is used between routers to build the multicast delivery tree state. PIM-SM creates unidirectional trees which are rooted at a group specific rendezvous point. The multicast delivery model can be either any-source multicast (ASM) or source-specific multicast (SSM). The ASM mode relates to many-to-many group communications while the SSM mode re-

lates to one-to-many group communications. In the ASM mode, a host uses the (*,G) state to join all sources in the multicast group with multicast IPv6 address G. In contrast in the SSM mode, a host uses the (S,G) state to join only the source with IPv6 address S in the multicast group with IPv6 multicast address G.

In PIM-SM, multicast group join requests are sent towards the rendezvous point for which the address may be obtained from the multicast group address [205]. In GLI-Split, the address of the rendezvous point is a global address so that also plain IPv6 nodes in non GLI-domains are able to join the multicast group. By sending the PIM-SM join request towards the rendezvous point, the required (S,G) state is established in all routers on the path from the receiver to the rendezvous point. In case the rendezvous point is in a GLI-domain, the GLI-gateway translates the address of incoming PIM-SM join requests to the local address of the rendezvous point. To sent multicast data to the multicast group, the source forwards the data to the rendezvous point which further distributes the multicast data according to the established (S,G) tree state. This way, no modifications are required for PIM-SM and MLD and multicast handling is the same as in the ordinary IPv6 architecture.

3.2.1.8 Security Concerns and Countermeasures

In this section, we consider new security threats that arise due to the separation of identity and location information, and show how GLI-Split handles these problems.

Potential new attacks The Loc/ID split-related issues can be generally classified as redirection attacks which constitute threats against confidentiality, integrity, and availability [206]. The general idea of these attacks is that an attacker manipulates the information in the mapping components like the mapping cache in order to redirect existing flows to a new target. There, the packets of the redirected flow may either be just dropped to cancel the availability of a service, they could be monitored to violate confidentially, or the contents of the packets could be changed to break the integrity of the transferred data. These issues apply to all

protocols where a mapping from an identifier to several locators is used. Examples are threats related to IPv6 multihoming [206], multipath TCP [207], or other new routing architectures like LISP [208].

The threats can be divided into control-plane and data-plane threats. Control-plane threats involve map request and map reply messages. Map request messages could be used for amplifying DoS attacks where an attacker requests mapping information under a spoofed locator address. This locator address is the address of the victim which might then receive a possibly large amount of mapping information. A small map request message from the attacker hence causes a possible large map reply message to the victim. Another threat which involves map request messages is the cache update mechanism which is for example used by the mobility extension LISP-MN [209]. This mechanism could be used to either insert false mapping information or to cause a cache overflow. These threats related to the mapping cache could also be achieved by utilizing the locator gleaning concept on the data-plane. Locator gleaning was proposed by LISP and should reduce the amount of necessary mapping lookups and hence speed up the initial communication establishment.

In the following, we provide more details about the threats related to locator gleaning and mobility update mechanisms and explain, how they can be avoided.

ID hijacking through locator gleaning Locator gleaning means that nodes store ID-to-GL mappings in their local caches when they see incoming packets with new ID/GL mappings. This possibly saves queries to the mapping service, but it causes a security problem so that locator gleaning should be avoided. Figure 3.11 illustrates how an attacker can hijack the ID of another node when GLI-hosts use locator gleaning. The attacker behind GLI-gateway X pretends to be node 1. It sends a packet with ID 1 in the source address to node 3. Node 3 receives the global GLI-address X.1 and updates its local cache with the mapping entry 1→X (“locator gleaning”). When node 3 contacts node 1 later, it uses the wrong locator from the local cache and the packets destined to node 1 will be delivered to the spoofing node behind X instead of the correct node behind E.

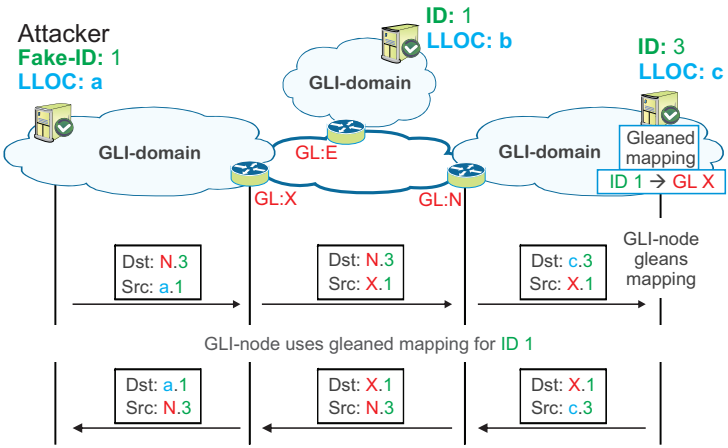


Figure 3.11: *When a GLI-host gleans locators from incoming source addresses, a malicious node can send a packet with a spoofed source ID and steal traffic intended for that ID.*

A countermeasure against that type of attack is implemented in the upgraded stack of GLI-nodes and GLI-gateways. When a packet is received with an unknown ID/GL combination in the source address, this mapping should be validated by a query to the mapping service before storing it in the local cache. Plain IPv6 nodes including those inside a GLI-domain are not affected by wrong mapping information since they are unaware of locators, identifiers, and mappings.

Flow interception through spoofed mobility updates When two upgraded GLI-nodes in different GLI-domains communicate with each other, a malicious GLI-gateway of another domain can deviate the flows to intercept them. This is illustrated in Figure 3.12. The attacking GLI-gateway sends a mobility update message to both GLI-nodes, saying that the locator of the other node has changed to the locator of the malicious GLI-gateway. Thus, the attacker attracts the traffic from both nodes and can forward it to the other node. Thereby, the

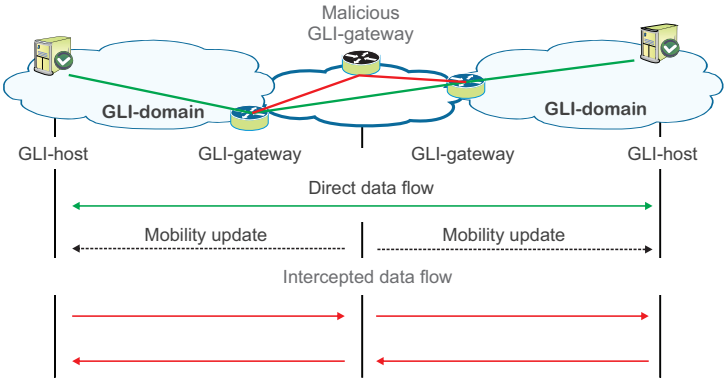


Figure 3.12: *A malicious GLI-gateway sends spoofed mobility update messages to two communicating GLI-nodes and intercepts their conversation.*

GLI-gateway can intercept the traffic although it is not on the path between the two communicating nodes.

This problem can be avoided if mobility update messages are signed by the sender and validated by the receiver. The use of a nonce has been proposed as a solution for that problem in ILNP [198].

3.2.2 Multihoming and Interworking

In Section 3.2.1, we have introduced the fundamental building blocks of the GLI-Split architecture considering only single-homed GLI-domains. In the multi-homed case, some additional mechanisms are required especially for supporting communications with plain IPv6 nodes either in plain IPv6 domains or in GLI-domains. In the following section, we first briefly explain the issues in the multi-homed case and then introduce mechanisms to solve these problems. After that, we introduce advanced mechanisms which facilitate several paths like multipath transfer and traffic engineering.

3.2.2.1 Issues with Plain IPv6 Nodes

As explained in Section 3.2.1.2, transport layer protocols use source and destination IP addresses including port numbers to map packets to flows. Moreover, bidirectional transport protocols or applications expect that packets flowing in the reverse direction (responses) have just interchanged source and destination IP addresses relative to packets flowing in the forward direction (requests) because receivers just swap these addresses when responding. In case of multihoming, this property can be easily violated as we observed in Section 3.2.1.4. When addresses of returning packets differ from the addresses used by the sender that initiated the connection, these packets cannot be mapped to the existing communication session. For upgraded GLI-nodes this is not a problem as only identifier addresses are used on the transport layer and changes on the network layer do not affect existing transport connections. However a plain IPv6 nodes sees these changes on the transport layer and hence, mechanisms are required to ensure that the addresses do not change when different paths are used.

3.2.2.2 Gateway Selection and Preservation

When edge networks are multi-homed, traffic may leave or enter through different gateways. First, we propose a mechanism for GLI-nodes to enforce a certain gateway for outgoing packets. Then, we suggest a method for GLI-gateways to preserve the global destination GLI-address of incoming traffic as source GLI-address in outgoing response packets. Both mechanisms require an address buffer to store a single additional GLI-address in the IPv6 header. This address buffer may be implemented by a new IPv6 extension header. It is only used inside a GLI-domain so that the size of external packets is not increased and, thus, cannot cause MTU issues in the Internet.

Gateway selection We assume a multi-homed GLI-domain with several GLI-gateways. When a GLI-node sends packets to a global address, the local routing system inside the local GLI-domain determines the outgoing GLI-

gateway to which the packets are forwarded. To enforce a certain GLI-gateway for outgoing traffic, the GLI-node stores the global destination address in the address buffer and sends the packet to the selected GLI-gateway, using a global address of the gateway as destination address. If the GLI-gateway receives a packet with an address buffer, it strips off the address buffer and substitutes the destination address of the packet by the address in the address buffer. As usual, the GLI-gateway also replaces the local source address with a global address, reflecting the gateway's global locator.

Global address preservation (GAP) When a destination GLI-domain is multi-homed, packets in the forward direction of a connection may take a different GLI-gateway than packets in the reverse direction. This may result in different global GLI-addresses at the initial sender and to the violation on the transport layer explained in Section 3.2.2.1. When the GAP-bit (see Figure 3.8) is activated in the global GLI-address of a packet's destination, the GAP-mechanism is triggered at the GLI-gateway of the destination domain to preserve the global destination address of request packets as the global source address of potential response packets. To that end, the GLI-gateway adds an address buffer to the packet storing the currently used global GLI-address of the destination before substituting this address by a local GLI-address. The destination node recognizes the activated GAP-bit of the global GLI-address in the address buffer and stores it. When response packets of the same connection are sent, the GLI-node uses gateway selection for these packets to the respective GLI-gateway. Thereby, the initial sender just sees swapped source and destination addresses and no violations on the transport layer occur.

3.2.2.3 GLI-Split with Plain IPv6 Nodes

The description of GLI-Split in Section 3.2.1 requires upgraded networking stacks for GLI-nodes. This is a major obstacle for its initial deployment. Upgrading the nodes can easily be achieved through system updates, which are frequently available for new equipment. However, it is hard to upgrade legacy

equipment for which updates are not offered anymore. Thus, for incremental deployability of GLI-Split within GLI-domains it is important to accommodate also plain IPv6 nodes without upgraded networking stacks. We describe additions to GLI-Split for that purpose. We show how the missing functionality of the plain IPv6 stacks can be compensated by modified behavior of the local DNS server and enhanced behavior of the GLI-gateways. We present an alternative mechanism for GAP based on stateful NAT which is used for interworking with the non-GLI Internet in the mulit-homed case. Furthermore, we propose a method to handle local traffic that mistakenly uses global GLI-addresses, which may happen when a global GLI-address was obtained for the destination from a DNS server outside the GLI-domain.

Modified behavior of local DNS servers The DNS is configured to return a global GLI-address with an activated GAP-bit for GLI-nodes in a multi-homed domain. When an upgraded GLI-node wants to contact another node, it receives its global address from the DNS, but uses only the integrated ID to query the local mapping system for the local or global GLI-addresses. Thereby, an upgraded GLI-node finds out whether the communication peer resides in the GLI-domain so that it uses a local GLI-address of the corresponding node for communication. Plain IPv6 nodes cannot query the mapping system and rely on the result from the DNS server. Therefore, the local DNS server should return local GLI-addresses for nodes inside its GLI-domain. However, local GLI-addresses should never leave a GLI-domain as they are not routable outside. Therefore, such a modified DNS server must be contacted only from within the GLI-domain. This may for instance be achieved by checking whether the source address is a local address.

Enhanced behavior of GLI-gateways Upgraded GLI-nodes directly register at the local mapping system and for plain IPv6 nodes, the registration is done, e.g., by an enhanced DHCP (see Section 3.2.1.2). Hence, the local mapping system knows which hosts inside its domain are upgraded and which are plain IPv6.

When a packet arrives at a GLI-gateway, the gateway asks the local mapping system for a local destination GLI-address of that packet. The local mapping system returns the requested address and upgrade information to the gateway. Thus, the gateway can behave differently depending on whether it forwards packets to upgraded or plain hosts.

NAT-based global address preservation NPTv6 [210] describes network address translation between IPv6 addresses. GLI-gateways could take advantage of this specification. In this light, GLI-Split resembles large-scale NAT for edge networks, but there are significant differences. The NAT operation performed by GLI-gateways is stateless and nodes in GLI-domains are reachable by global addresses. GLI-Split even improves their reachability beyond provider changes.

When a GLI-gateway receives packets with an active GAP-bit in the destination address, it must assure that source and destination address are simply swapped and do not change for responses. To achieve this, upgraded GLI-nodes implement GAP using gateway selection. Plain IPv6 nodes lack this feature. We show how it can be compensated through stateful network address translation (NAT) by GLI-gateways. The GLI-gateway keeps a NAT table that maps pairs of external source and destination addresses to pairs of internal source and destination addresses. Furthermore, a part of the ID space is reserved for private use inside GLI-domains that can be used by GLI-gateways to perform NAT. When a GLI-gateway receives an incoming packet for a plain node with the GAP-bit set in the global destination address, it substitutes the source and destination address according to the entries in its NAT table. When no matching entry is found in the NAT table, a new entry is established that maps the external address pair to the corresponding local destination address and a global source address that consists of the local locator of the gateway and a currently unused private ID. Response messages from the destination node are returned to the same GLI-gateway which replaces the source and destination addresses according to the entries in its NAT table so that leaving response messages have symmetric source and destination addresses relative to previous request messages.

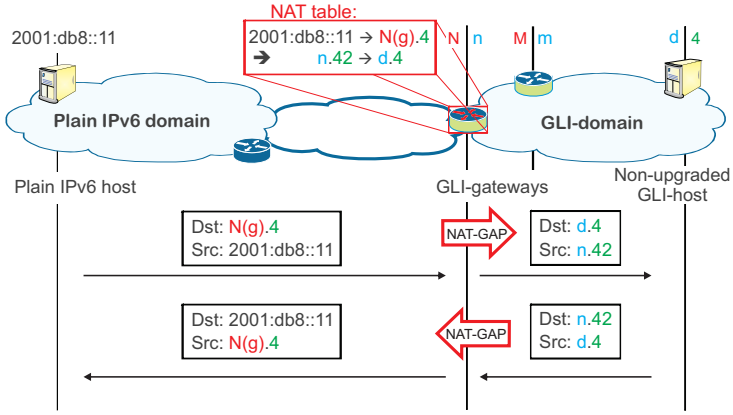


Figure 3.13: *IPv6 host 2001:db8::11 in a non-GLI-domain communicates with the non-upgraded GLI-node 4 using NAT-based GAP.*

Figure 3.13 illustrates this procedure. GLI-gateway N receives a packet with a global source address 2001:db8::11 and global destination GLI-address '$N(g).4$'. It queries the local mapping system for a local GLI-address of ID 4 and obtains '$d.4$' as well as the information that node 4 is a plain IPv6 node. Therefore, NAT-based GAP and gateway selection must be applied. The GLI-node searches its NAT-table but does not find a matching entry. Therefore, it picks a currently unused private ID (e.g. 42) and records the mapping (2001:db8::11, '$N(g).4$')→ ('$n.42$','$d.4$') in its NAT table. It translates the source and destination address of the packet accordingly and the packet is delivered to node 4. When response messages from node 4 return to the gateway N, it substitutes the source and destination address in the response packet according to the reverse entry in the NAT table.

Local traffic with global GLI-addresses When a plain IPv6 node in a GLI-domain wants to communicate with another node in the same domain, it

should receive a local GLI-address from the DNS. If it accidentally obtains a global GLI-address with a set GAP-bit for such a node from an external DNS, the GLI-gateways have to follow special rules to handle this correctly. We illustrate this using Figure 3.14. Nodes 3 and 4 are in the same GLI-domain. Node 4 wants to send a packet to node 3 and has obtained a global address of node 3. Unlike an upgraded host, the plain host 4 cannot contact the local mapping system to find out the correct local address. It just sends a packet with the global GLI-destination-address '*N(g).3*' and the packet is forwarded to the gateway N.

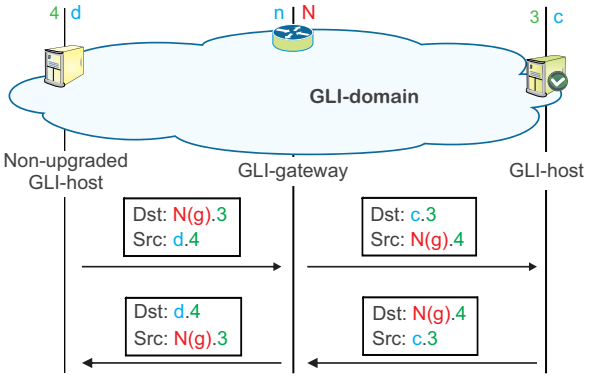


Figure 3.14: *Reflection of local traffic: plain IPv6-GLI-node 4 communicates with GLI-node 3 via global addresses. The gateway of their GLI-domain reflects the traffic between the hosts to ensure addressing symmetry.*

Gateway N recognizes that both sender and receiver of the packet are inside its own GLI-domain. It substitutes the global GLI-address of the destination with an appropriate local GLI-address. The local source address is replaced by the global source address '*N(g).4*', reflecting N as gateway, node 4 as source, and setting the GAP-bit. The packet reaches node 3 and when node 3 responds to the packet, the previously set GAP-bit ensures that source and destination address are simply

swapped. Thus, also the response message is returned to gateway N. Gateway N handles this packet like the one before so that node 4 receives response messages with the global address of node 3 in the source field. This way, bidirectional communication is possible. The described operation of the gateway is stateless. There is no need to build or store any mapping table. The gateway uses only the information in each packet to translate source and destination addresses.

3.2.2.4 Multipath Support

When an edge network is multi-homed, its nodes have multiple paths to destinations in other domains, but only a single path can be used in the current Internet. However, networking could benefit from using all available paths [211,212]. For example, a node could balance traffic over multiple paths to maximize its throughput or it could improve fault tolerance [3,213]. The Stream Control Transmission Protocol (SCTP) [214] takes advantage of this. Multipath support requires that hosts can determine through which gateway their traffic should be carried. If both the source and the destination network are multi-homed, multipath routing could enforce specific gateways both in the source and destination domain. With GLI-Split this can be achieved as follows. A GLI-node queries the global mapping system for the set of its own global GLI-addresses and the one of its corresponding node. Each combination of global source and destination GLI-addresses represents a different path. These paths are not necessarily entirely disjoint, but possibly on the last mile between the customer and the provider network which is often the slowest and most error-prone part of the path. To send traffic over a specific path, a GLI-node selects the appropriate GLI-gateway for its outgoing traffic (see Section 3.2.2.2) and uses the appropriate global GLI-address for the destination node to select a specific GLI-gateway in the destination domain.

3.2.2.5 Traffic Engineering Support

A GLI-domain might be multi-homed to two ISPs: a cheap ISP for carrying its best effort traffic and an expensive ISP for carrying its premium traffic from de-

manding applications such as games or live video. In our example in Figure 3.6, E may be the gateway to the cheap ISP and F may be the gateway to the expensive ISP. Thus, best effort traffic should be exchanged through gateway E while premium traffic should be exchanged through gateway F.

Gateway selection for self-initialized communication We assume that GLI-node 1 wants to establish a real-time connection with another node outside its own domain. It selects outgoing GLI-gateway F using the method described in Section 3.2.2.2. When sending the packet to F, it activates the GAP-bit in the global GLI-address '*F(g).**' to indicate that F should set the GAP-bit in the global source address. Thus, gateway F substitutes the local GLI-address '*a.1*' in the source field of the packet with the global GLI-address '*F(g).1*'. As a result, the corresponding node of node 1 will send return data to '*F(g).1*' and not to another global address of 1. This is important for destination nodes in GLI-domains with upgraded networking stacks as they could send return data to '*E.1*'. Hence, client node 1 has successfully selected gateway F for outgoing and incoming traffic.

Gateway selection for incoming traffic Gateway selection for incoming traffic requires support from the DNS and the global mapping system. A node may offer different services: one requires best effort transport and another requires premium transport. The DNS name for the best effort service should resolve, e.g., to E(g).1 and the name for the premium service should resolve, e.g., to F(g).10. Nodes without upgraded networking stacks use this information to contact the server. Nodes with upgraded networking stacks use just the destination ID 1 or 10 and query the local or global mapping system for an appropriate local or global address. Therefore, the global mapping system should be configured to return E.1 and F.10 as default and F.1 and E.10 as alternative to be used when the default values do not work. This ensures that GLI-nodes with upgraded networking stacks usually contact the best effort service through ID 1 and gateway E and the premium service through ID 10 and gateway F as desired.

3.2.3 GLI-Split Implementation

The previous sections described all building blocks, concepts, and algorithms that together form the GLI-Split architecture. As a first demonstration that this concept actually works, we implemented GLI-Split as a proof-of-concept simulation [36] in the INET Framework [215] for OMNeT++ [216]. In this section, we describe the simulation and use it to present a brief evaluation of the round-trip-time in different communication scenarios.

3.2.3.1 OMNeT++ Simulation

OMNeT++ is a modular simulation environment which can be applied to all problem domains where the discrete event based approach is suitable. Especially for communication networks, the INET Framework for OMNeT++ offers all required communication protocols to model TCP/IPv6 based networks.

We modeled different GLI- and plain domains, which are connected via a core network consisting of three provider networks. A global DNS module ensures DNS-name-to-global-locator address resolution. Each domain has an administration module (admin) which configures the identifier and local locator prefixes per domain and installs required routing entries. On the global area, a similar administration module configures the global locators per domain and installs inter-domain routing entries. The GLI modules in the implemented model of our GLI-Split architecture comprise the GLI-gateways as well as local and global mapping servers. Each GLI-domain has its own local mapping server and there is one global mapping server for the entire network which serves as interface to the mapping service.

In the following, this test network is used to present a brief evaluation with respect to the round-trip-time for communications between different domains.

3.2.3.2 Round-Trip Time Evaluation

For the evaluation, all inter-domain link delays in the test network have been set to *2 ms*. In the following, we take a brief look at the round-trip-time (RTT) in

all six possible combinations of source and destination nodes (see Table 3.1). In the first four scenarios, no mapping lookups are required and hence, the RTT just comprises the corresponding inter-domain link delays. This base delay applies to all six combinations and comprises the eight single link delays back and forth between source and destination domain. In the fifth scenario, a plain IPv6 host in a GLI-domain communicates with a GLI-host in another GLI-domain. In this case in the return direction, the GLI-host performs a mapping lookup because the returned address from the DNS is a GLI-address. This results in an additional delay of *12 ms* for the initial message. For subsequent message, the mapping has already been cached and no further lookups are required. The largest RTT occurs in the last scenario, where two GLI-hosts in different domains communicate. This time, a lookup at each domain is required which adds two times *12 ms* to the base delay. But again, this applies only to the first messages. This result may seem unfavorable, but this evaluation does not consider the advanced mechanisms which can be utilized by upgraded GLI-nodes. The next section explains some of these in detail.

Table 3.1: *Comparison of initial and second RTT between plain hosts (H_P) and GLI-hosts (H_{GLI}) in GLI-domains (D_{GLI}) or plain IP domains (D_P).*

Connection	**1st RTT**	**2nd RTT**
H_P in $D_P \rightarrow H_P$ in D_P	16ms	16ms
H_P in $D_P \rightarrow H_P$ in D_{GLI}	16ms	16ms
H_P in $D_P \rightarrow H_{GLI}$ in D_{GLI}	16ms	16ms
H_P in $D_{GLI} \rightarrow H_P$ in D_{GLI}	16ms	16ms
H_P in $D_{GLI} \rightarrow H_{GLI}$ in D_{GLI}	28ms	16ms
H_{GLI} in $D_{GLI} \rightarrow H_{GLI}$ in D_{GLI}	40ms	16ms

3.2.4 Benefits and Deployment Considerations of GLI-Split

GLI-Split improves the scalability of Internet core routing by removing the need for fine-grained provider-independent addresses and moreover provides many benefits for edge networks. We first summarize the full set of advantages for communication between upgraded GLI-hosts. Then, we analyze which subset thereof is also available for plain IPv6 nodes in GLI-domains. Finally, we present required changes to the existing IPv6 architecture and discuss deployment considerations.

3.2.4.1 Benefits for Upgraded GLI-Nodes

With GLI-Split, hosts are not configured with any global locators. This simplifies provider changes as it makes renumbering in terms of assigning new global locators obsolete. Renumbering nodes inside a GLI-domain means assigning new local locators. This is necessary for example when subnetworks need to be rearranged for administrative reasons. With GLI-Split, this is facilitated because local locators are automatically assigned to nodes and nodes outside a GLI-domain are unaware of the corresponding local addresses. Nodes inside a GLI-domain use only stable identifier addresses for configuration and connection establishment.

GLI-Split enables multihoming and takes advantage of all benefits associated with multihoming. When the connection from the local GLI-domain to its ISP fails, the local routing system reroutes the traffic to another GLI-gateway. When a destination is not reachable at its default locator, the source may be notified about a failure and may address the traffic to another global GLI-address.

This represents a host-based rerouting technique which is an alternative to network-based rerouting techniques as presented in [217]. GLI-hosts can select the GLI-gateways of the source and destination domain and thereby enable multipath routing which might be useful for host-based load balancing. Traffic engineering for outbound traffic can be performed by enforcing the GLI-gateway of the source domain with gateway selection. In addition, traffic engineering for

inbound traffic can be achieved by enforcing the GLI-gateway of destination domains. This is done by activating the GAP-bit in global GLI-addresses for certain services or nodes. Moreover, GLI-Split provides improved mobility support in the sense that corresponding nodes can contact mobile nodes directly without triangle routing over a home agent.

Most of these advanced networking features are not available in today's Internet or require provider-independent addresses. GLI-Split enables even smallest edge networks to use these features without increasing the routing tables in the DFZ. In contrast to many other future Internet routing proposals, GLI-Split does not suffer from potential problems due to increased packet sizes after encapsulation and it does not require special interworking techniques with the plain IPv6 Internet.

3.2.4.2 Incentives for Early Adopters

GLI-nodes of early adopters usually communicate with the plain Internet which reduces the set of advantages provided by GLI-Split. However, it still has appealing benefits. Multihoming is still possible. GLI-domains can change providers without renumbering, but global GLI-addresses communicated to external nodes need to be changed. Traffic engineering for outbound and inbound traffic can still be performed.

3.2.4.3 Benefits for Plain IPv6 Nodes in GLI-Domains

Plain IPv6 nodes can be accommodated in GLI-domains. This is a valuable feature for incremental deployability since equipment for which upgraded GLI-networking stacks are not yet available or legacy equipment for which GLI-networking stacks will not be provided anymore can be operated in GLI-domains.

Internal renumbering after a provider change is simplified because plain IPv6 nodes in GLI-domains know only their local GLI-address. Hence, provider changes are invisible to them like to nodes behind a NAT-gateway.

Multihoming is possible. When communicating with upgraded GLI-nodes,

they can perform host-based rerouting so that also plain IPv6 nodes in multi-homed GLI-domains get better resiliency. Traffic engineering is supported for incoming traffic but not for outbound traffic.

3.2.4.4 Implementation and Deployment Considerations

The GLI-Split functionality inside upgraded GLI-nodes requires host updates which could be distributed via operating system updates. The modification introduces an additional shim layer between the network layer and the transport layer which performs the vertical address translation (see Section 3.2.1.2) as well as mapping registration and mapping lookups for outgoing packets (see Section 3.2.1.3). The modifications inside hosts do not require changes to transport layer protocols like TCP as the horizontal address translation done by the shim layer is transparent to upper layers.

In the initial deployment phase, host updates are not necessarily required for all nodes in a GLI-domain as GLI-Split also accommodates plain IPv6 nodes (see Section 3.2.2.3). To achieve that, a modified DNS is required which returns either a local or a global address depending on whether source and destination host are in the same or different GLI-domains (see Section 3.2.2.3). The modification however does not require changes to the DNS protocol as it could be realized via configuring a split-horizon or split-view DNS that provides different DNS records depending on the location of the querying source host. The only protocol modification that is required is the extension of the DHCPv6 protocol which is used for the assignment of local addresses to hosts inside GLI-domains (see Section 3.2.1.2). The DHCP needs to return the address of the local and global mapping service which could be realized via a new DHCP options field like it is currently done for DNS server addresses. A new DHCP options field may also be used to return a set of global locators to upgraded GLI-nodes. For plain IPv6 nodes, the DHCP needs to know the ID and MAC address to compute local GLI-addresses with checksum compensation. In addition, the DHCP is responsible to register the advertised local GLI-addresses in the local mapping service and the corresponding global GLI-addresses in the global mapping service (see Section

3.2.1.2) which requires an extension of the DHCP server functionality.

The most critical entity in GLI-Split is the GLI-gateway which implements the main network functionality. It is responsible for the horizontal address translation (see Section 3.2.1.2) as well as for all mapping service related mechanisms like mapping registration or mapping lookups. Further, it realizes all mechanisms which are required to support plain IPv6 nodes in GLI-domains (see Section 3.2.2.3). The complexity of such a GLI-gateway is comparable to a common NAT-gateway, which also translates addresses and recomputes TCP checksums. NAT functionality is in the current Internet mostly used in residential home gateways or in carrier-grade or large scale NAT solutions in provider networks. Large scale NAT solutions are for instance used in the IPv6 transition mechanisms like NAT444 [218]. Hence we assume that the complexity of the GLI-Split mechanisms can be handled by available of-the-shelf routing hardware. In addition, we assume that GLI-gateways are located closer to the edge which also reduces the size of GLI-domains and hence the amount of traffic that must be handled by GLI-gateways.

3.3 A Mapping System for Future Internet Routing (FIRMS)

The locator/identifier (Loc/ID) split principle, as for example used in the GLI-Split presented in the previous section, is expected to overcome many challenges of the current routing architecture, in particular, scaling issues in the Internet [161,162]. Loc/ID split separates the functionality of current IP addresses into two different parts: the RLOC and the EID[1]. The EID addresses an endpoint and the RLOC addresses the network attachment point, i.e., the position of the endpoint in the Internet. A mapping system is required to bind both address spaces together. When a node or a whole network changes its point of attachment to the Internet or performs traffic engineering [219], the mapping system is updated with the new EID-to-RLOC information.

From the application layer's point of view, only EIDs are visible and used to address packets. Depending on the routing architecture, either a new layer in the IP stack of the source node or an intermediate node, e.g., a gateway router, queries the mapping system to obtain an RLOC for the destination EID and adds the RLOC to the packets to make them routable over the Internet. This mapping node may locally cache the EID-to-RLOC mappings to avoid unnecessary lookups.

Some routing proposals add the RLOC at the source node [195,198], but many others add it at some intermediate node [35,183,187,188,220–222]. If the mapping node is an intermediate node and additional packets addressed to the same EID arrive before an EID-to-RLOC-mapping is returned and stored in the cache, these packets may be buffered or dropped. To avoid extensive delay and packet loss, the mapping system should either be very fast or provide a packet relaying function, which can temporarily forward EID-addressed packets over the mapping system to the correct destination. If Loc/ID split becomes part of future Internet routing, mapping systems become a vital part of the Internet architecture. They must be resilient to outages, secure, fast, and should provide a packet relaying function.

[1] As FIRMS is not limited to GLI-Split, we adopt the RLOC/EID terminology used in the LISP context.

In this section, we present FIRMS, a distributed mapping system for future Internet routing. It supports routing architectures implementing the Loc/ID split. It is fast, scalable, resilient, secure, and it is able to relay packets.

The content of this section is mainly taken from [11, 12, 37–39] and it is organized as follows. First we analyze requirements for a mapping system in Section 3.3.1. The new FIRMS architecture is described in detail in Section 3.3.2 and Section 3.3.3 presents a rough calculation about the expected load on various system components. Section 3.3.4 describes a simulation of FIRMS in the OMNeT++ simulation framework and our proof-of-concept implementation in the G-Lab testbed. Section 3.3.5 provides a classification for mapping architectures and Section 3.3.6 uses this classification to give an overview over related work on mapping systems.

3.3.1 Requirements Analysis

Depending on whether the mapping lookup is performed in hosts or intermediate nodes, the requirements for a mapping system are slightly different. Therefore, a brief requirement analysis for mapping systems is given in this subsection.

3.3.1.1 Mapping Structure

The underlying future Internet routing architecture mandates the mapping structure to be supported by the mapping system. The granularity of EIDs and the way they are aggregated has a deep impact on the architecture.

3.3.1.2 Scalability

The Internet has been growing fast in the last three decades in terms of hosts and in terms of networks. We expect this process to continue due to the increasing ubiquity of Internet applications. Therefore, the number of EIDs and/or EID-prefixes will also increase in the future. As we could not foresee the tremendous growth of the Internet in the past, mapping systems must be able to handle a similar growth in the future.

3.3.1.3 Resilience

Today's Internet already uses a mapping service: the DNS. If the DNS fails, the reachability of other end systems is strongly compromised unless the user knows their IP addresses in addition to their DNS names. If the mapping service for a Loc/ID split routing architecture fails, there is mostly no way for the user to reach other end systems as the user knows only their EIDs. Therefore, the EID-to-RLOC mapping system must not fail. As a consequence, resilience requirements for the EID-to-RLOC mapping system are clearly stricter than for the DNS.

3.3.1.4 Security

For the same reasons as above, an EID-to-RLOC mapping system has to provide at least the same security as the existing DNS. It has to withstand direct and indirect security threats. An example of direct threats is denial-of-service attacks. These types of attacks are difficult to prevent. Examples of indirect threats are takeover of authoritative mapping sources and cache poisoning which happens when a mapping node accepts wrong mappings. A carefully designed security model should address as many threats as possible to minimize targets for attackers, and it should be extendable.

3.3.1.5 Relaying

ITRs may locally cache mappings including an expiration date to reduce the request loads [180, 223] for any type of mapping system. During regular operation of a mapping system, cache misses can occur. A cache miss means that requested information for a certain mapping has not been fetched from the authoritative source, yet. It is advantageous if the mapping system offers a relay service so that mapping nodes can forward packets with missing RLOCs over the mapping system to another node that can forward the packet to the destination. This avoids packet loss and extensive delay, but can lead to packet reordering when subsequent packets are forwarded directly.

3.3.2 The FIRMS Architecture

In this section we present FIRMS, a new mapping system for future Internet routing. We describe its architecture, specify its operation, and discuss its resilience and security features. FIRMS is not only applicable to the GLI-Split architecture presented in the previous section, but to many routing approaches that are based on the Loc/ID split. Thus, we use the widely used general nomenclature (EID, RLOC, ITR, ETR) in the following.

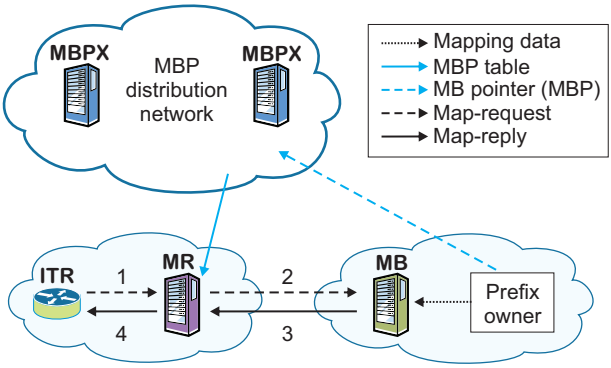


Figure 3.15: *Basic operation of FIRMS.*

3.3.2.1 General Idea

Figure 3.15 illustrates the basic structure and operation of FIRMS. We assume that EIDs are assigned to their owners in prefix blocks. Each prefix owner provides a map-base (MB) holding the EID-to-RLOC mappings for all its EIDs. The operation of the MB may be delegated to a specialized company. A map-base pointer (MBP) is a data structure containing information about the MB. The prefix owner registers this information in the global MBP distribution network which collects all MBPs and constructs a global MBP table. Each ITR is configured with

a map-resolver (MR). The MR registers at the MBP distribution network and receives a copy of the global MBP table. When the ITR requires an EID-to-RLOC mapping for an EID, it sends a map-request to its MR. The MR looks up the address of the responsible MB in its local copy of the MBP table and forwards the map-request to this MB. The MB returns a map-reply containing the desired EID-to-RLOC mapping to the MR which forwards it to the ITR. If a non-existing mapping is queried, a negative map-reply is returned. This design requires that MRs and MBs have globally reachable RLOC addresses. We present ITRs and MRs as two different entities because they have different functionality. However, the MR functionality may be integrated in an ITR which saves communication overhead and simplifies the design.

3.3.2.2 Map-Base Pointer Distribution Network

We explain how MBPs are distributed from prefix owners to MRs. We assume that EIDs are assigned in a similar way as IP addresses are assigned today; many routing proposals even assume that EIDs are IP addresses. IANA delegates IP address blocks to the five regional Internet registries (RIRs): AfriNIC, APNIC, ARIN, LACNIC, and RIPE NCC. They delegate subsets thereof to local Internet registries (LIRs). Both RIRs and LIRs partition the address space in prefix blocks and assign prefixes to organizations (prefix owners).

Every RIR or LIR runs a map-base pointer exchange node (MBPX). Figure 3.16 shows that the MBPX of a LIR (LIR-MBPX) is connected to the MBPX of its RIR (RIR-MBPX), and the RIR-MBPXs are fully meshed. This constitutes the MBP distribution network. The prefix owner adds, changes, or removes MBPs for its EID prefixes at the MBPX of its LIR or RIR. An LIR-MBPX forwards this data to its superordinate RIR-MBPX. The RIR-MBPX collects the MBPs for all EID prefixes under its control and compiles a regional MBP table. The MBP tables are exchanged among all RIR-MBPXs so that each of them has a copy of the global MBP table. They push this information to their subordinate LIR-MBPXs which forward it to all MRs that have registered for that service. An involvement of RIRs or LIRs for the support of Internet services is not uncommon. For

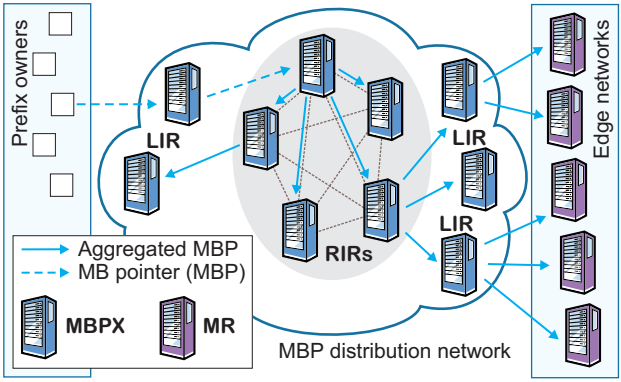


Figure 3.16: *Propagation of MBP updates in the map-base pointer distribution network.*

instance, RIRs and LIRs play an active role for reverse DNS lookup.

To facilitate incremental updates to MBP tables, the RIR-MBPX collects individual MBP updates from prefix owners over some time and provides sequentially numbered aggregated updates. It pushes them to the other RIR-MBPXs and its subordinate LIR-MBPXs. When an RIR-MBPX, LIR-MBPX, or MR receives such an update, it applies the changes to its local copy of the MBP table and forwards the updates to all its subordinate LIR-MBPXs or MRs. The numbering of the updates contributes to the consistency of all MBP tables. If an update is received with an unexpected number, missing updates are detected and their retransmission is requested.

3.3.2.3 Mapping Retrieval

To minimize query overhead, ITRs and MRs have local caches for EID-to-RLOC mappings. To avoid stale information, mappings are automatically purged from the caches after their time-to-live has expired. Figure 3.17 illustrates how EID-to-RLOC mappings are retrieved in combination with caches. When the ITR re-

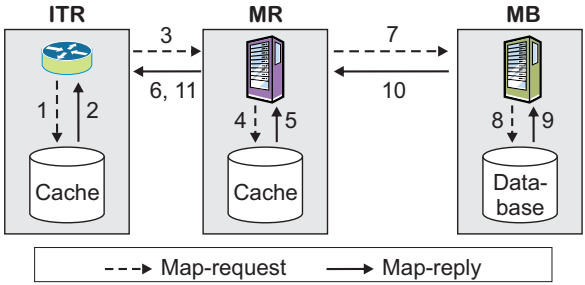


Figure 3.17: *Cascading mapping retrieval in FIRMS.*

quires a mapping, it first checks its cache (1) and can often retrieve the mapping immediately (2). In case of a cache miss, the ITR sends a map-request to the MR (3).

When the MR receives a map-request from an ITR, it first searches its cache (4) and, if successful (5), sends a map-reply back to the ITR (6). If unsuccessful, the MR searches its local copy of the MBP table using a longest prefix match with the requested EID and selects the appropriate MBP. It chooses a suitable MB from the MBP and sends a map-request to this MB (7). The MR keeps a state for the requested EID so that a map-reply can later be returned to the requesting ITR. The state is removed when the MR returns the requested information to the ITR or when a timer expires.

When the MB receives a map-request from a MR (7), it retrieves the EID-to-RLOC mapping from its database (8, 9) and sends it back to the MR in a map-reply (10). The MR stores the mapping of the map-reply in its cache and sends a map-reply back to the ITR (11) which also stores the mapping in its cache. The caches at the ITRs and MRs minimize the retrieval time for the mappings and reduce the frequency of map-requests. Performance issues of caches have been discussed in [180].

We propose several enhancements to improve the speed and scalability of the mapping retrieval.

- MRs and ITRs should limit the rate of map-requests for the same EID to avoid outgoing map-request storms.
- Every EID may have its own RLOC. If EIDs of a common prefix block share the same RLOC, their EID-to-RLOC mappings may be aggregated to a single EID-prefix-to-RLOC mapping. On the one hand, this saves storage in caches and databases. On the other hand, an EID-prefix-to-RLOC mapping covers the RLOCs for many EIDs. That makes additional map-requests redundant when the ITR needs a mapping for a new EID that is already covered by an EID-prefix-to-RLOC mapping in its cache. Thus, EID-prefix-to-RLOC mappings minimize the lookup delay and take load off the MR and the MB.
- If an MR serves only a single ITR, the caches of the MR and the ITR are likely to have the same content so that advantage cannot be taken from the cache at the MR. Hence, several ITRs should be configured with the same MR. Then, the MR may be able to serve an ITR's map-request from its cache with EID-to-RLOC mappings that have been requested earlier by other ITRs.
- Alternatively, the MR functionality may be integrated in ITRs. This saves communication overhead and simplifies the overall structure. Then, the MR is mainly an interface to logically separate ITR and MR functionality within the same physical node.

3.3.2.4 Packet Relaying

We first outline the motivation for a packet relaying service and then explain how it can be offered by FIRMS.

Motivation for packet relaying When an ITR receives a packet addressed to an outbound EID, it tries to retrieve the EID-to-RLOC mapping from its local

cache and, if successful, tunnels the packet to the ETR whose RLOC was given in the mapping. In case of a cache miss, the ITR retrieves the mapping over the network which is a time-consuming process. This can happen for the first packet of a communication session when a new flow to a previously not contacted EID is established. The arrival rate of such packets is most likely rather low. In contrast, when traffic is shifted from one ITR to another, the rate of packets with missing RLOCs can be very high. This can happen, for example, when the primary ITR of a networks fails, when the internal routing is changed, or when load balancing policies change. There are three options to handle such outbound packets until their mappings are available in the ITR cache: they can be dropped, stored, or relayed to another node that knows how to forward them.

When the ITR drops packets, many applications will resend them, and by then the mapping is hopefully available in the ITR's cache. This might work for the first packet of a communication, but especially this packet can be quite important, e.g., the initial SYN packet of a TCP connection setup. Losing the first packet can significantly impede the communication setup. When a large number of flows is shifted from another ITR, an immense number of packets is dropped until a mapping can be retrieved from the MS.

As an alternative, the ITR stores the packet until the requested mapping returns from the MR. Then, the ITR can add the RLOC to the packet and send it. This option requires a large buffer to store such packets. Additional logic is needed to continue the processing of the packets as soon as the missing mappings arrive or to drop them when a timer expires. The buffer may overflow and packets may be lost, especially when packets arrive at a high rate. This gives rise to potential attacks where attackers send packets to the ITR with yet unknown destination EIDs. Thus, this option requires complex engineering and still cannot avoid packet loss.

Packet relaying to another node that knows how to forward the packet seems a promising idea since it avoids the drawbacks of dropping and storing. Therefore, it has been proposed also for other mapping systems in the LISP context under the name "data probe" [177, 178, 224].

Packet relaying in FIRMS Figure 3.18 illustrates how packet relaying can be realized in FIRMS. Normally, the ITR has the EID-to-RLOC mapping in its cache (1, 2) and tunnels the packet to the ETR (3). In case of a cache miss, the ITR tunnels the packet to the MR (4). If the MR finds the required mapping in its cache (5, 6), it tunnels the packet to the ETR (7). Otherwise, the MR tunnels the packet to the appropriate MB (8). The MB has the mapping in its database (9, 10) and tunnels the packet to the ETR (11). This design has several nice properties.

- Only the MR and the MB are involved in the relay process. They are operated by the sender's network and the prefix owner or on behalf of the prefix owner so that these elements have economic incentives to forward the data. In particular, no elements of public infrastructure or other private networks are involved. This is different in other proposals where relayed packets are transmitted over an overlay network [177, 178, 224].
- If the MB is collocated with the destination network of the EID and near the ETR, the path of the relayed packets is hardly stretched.
- Relayed packets can be interpreted as implicit map-requests and save explicit map-requests. That means, MRs or MBs not only tunnel the relayed packets to ETRs when they have appropriate mappings, they also respond with map-replies. When an ITR relays multiple packets with the same EID, map-reply storms may occur and measures should be taken to avoid them (see Section 3.3.2.3).

3.3.2.5 Resilience Concept

We propose a protection concept for FIRMS based on simple replication so that the mapping service survives in case of any component failure. Moreover, additional LISP-specific resilience methods can also be applied with FIRMS.

Protection for FIRMS RLOCs can become unreachable. If an edge network is multihomed, it is reachable over alternative RLOCs that also appear in the EID-to-RLOC mappings. When an ITR detects problems with an RLOC, it marks the

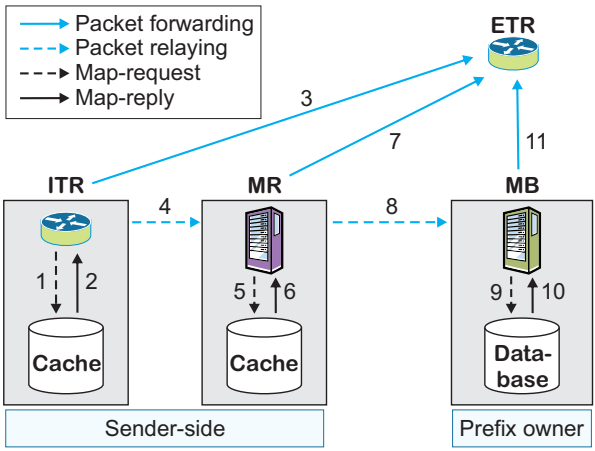


Figure 3.18: *Packet forwarding and relaying in FIRMS.*

particular RLOC in its cache as unreachable for a while and uses an alternative RLOC instead.

MRs can fail. ITRs can be configured with multiple MRs. When an ITR detects the failure of an MR, it marks the MR as unreachable for a while and contacts another configured MR.

MBs can fail. A prefix owner has multiple MBs with identical mappings and records their addresses in the MBP. When an MR detects the failure of an MB, it marks the MB as temporarily unreachable and contacts an alternative MB whose address is given in the MR's local copy of the MBP table.

MBPXs can fail. As a consequence, MRs do not receive updates for the MBP table in time. An MR can register with multiple MBPXs, and if one of them fails, the MR still receives updates from the other MBPXs.

LISP-specific protection The LISP encapsulation header reserves four bytes as "locator status bits" [184]. These bits correspond to an ordered list of

RLOCs in the EID-to-RLOC mapping and indicate which of them are operational. The prefix owner can change this information at the MBs to give the ITRs a hint which RLOCs are currently reachable.

In addition, database map-versioning is proposed. EID-to-RLOC mappings are equipped with version numbers to facilitate detection of outdated information. The ITR adds the current version number for the mapping of the source EID in the LISP encapsulation header. The ETR examines the version number in the encapsulation header of incoming packets and compares them with the version number in the corresponding mappings stored in the local cache of the collocated ITR. If the mapping in the local cache is outdated, the ITR sends a map-request for the respective EID to update the mapping in its cache. This mechanism helps to keep track of mapping changes.

3.3.2.6 Security Concept

In Loc/ID split based routing architectures it is crucial that EID-to-RLOC mappings are recent and authentic. In FIRMS, MBs are under the control of prefix owners who must make sure that their MBs respond with correct mappings for their EID range. The MR must be able to verify that mappings arrive from the queried MB, and that mappings are unaltered and recent, e.g., that they are not a replay of an old mapping entry. We first propose mechanisms which ensure that the MR can trust the information in its MBP table, and then we add functionality to achieve the other requirements.

Figure 3.19 visualizes the security concept of FIRMS. [225] proposes an extension to the ITU-T X.509 v3 standard for a public key infrastructure (PKI) that allows to bind a list of IP prefixes to the subject of a so-called resource certificate. It is already provided for use by APNIC [226]. We use it in FIRMS to transfer the right-to-use for IP prefixes from IANA through the RIRs and LIRs to prefix owners. Thus, prefix owners can authenticate themselves as the rightful owners of their EID prefixes. They use this feature for adding, modifying, or removing EID-to-RLOC mappings at the MBs, and for adding, modifying or removing MBPs at the MBPX of the LIRs/RIRs from which they received their EID space.

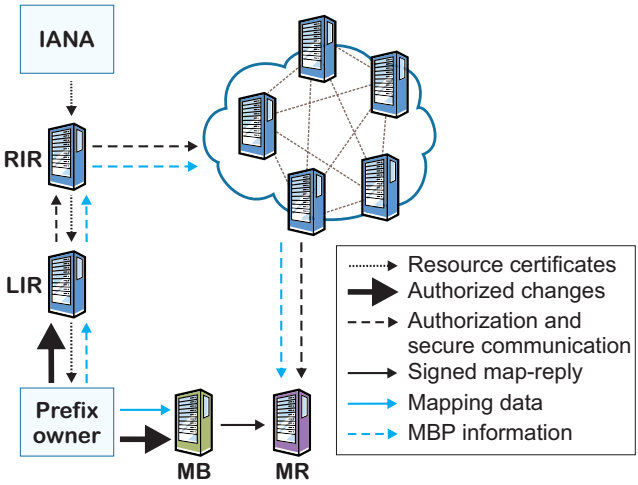


Figure 3.19: *Security concept for FIRMS.*

When MBP updates are propagated from subordinate MBPXs to superordinate MBPXs, transport layer security (TLS) [227] or datagram TLS (DTLS) [228] together with resource certificates ensure that an MBPX can propagate MBP changes over a secured connection only if it has the right-to-use for the corresponding EID ranges. The MBPX of an LIR trusts the MBPX of its RIR or superordinate LIR, and each MR trusts the MBPXs it is connected to. To receive trusted MBP updates from them, they just authenticate them and use a secured connection for data transport. As a result, the MR can trust the MBP information in its local MBP table.

An MR must be able to verify whether the mappings obtained from an MB are authentic and recent. To that end, an additional PKI for MBs is introduced. The MB includes a time stamp in the map-reply and signs the map-reply with its private key before sending the map-reply to the MR. The public key of each MB is included in the corresponding entry of the global MBP table whereof each

MR already stores a local copy. When the MR receives a map-reply, it uses the MB's public key to validate the message to be sure that the contained mapping is authentic, and checks the time stamp to be sure that the mapping is recent. The MR can immediately validate the obtained map-replies without verifying any trust chain which would generate extra delay and traffic. This was a major design goal of the FIRMS architecture and ensures that FIRMS is fast and scalable.

3.3.3 Performance Assessment

In this section we estimate the expected loads on various system components in FIRMS and show that they are in a manageable order of magnitude.

3.3.3.1 Record Sizes

We calculate the size of EID-to-RLOC and MBP records in FIRMS. We assume that both EIDs and RLOCs have the same format as IPv6 addresses which are 16 bytes long. Edge networks can take advantage of multihoming more easily with Loc/ID split, but being connected to more than 4 ISPs has only limited benefit [229]. Therefore, we assume that nodes are usually connected to the Internet over three providers which results in an average number of three RLOCs per EID-to-RLOC record. This record contains additional information like a time-to-live (5 bytes), some traffic engineering attributes (10 bytes), and a security signature with a timestamp (16+5 bytes) so that its average size is about 100 bytes.

MBPs consist of an EID prefix (8 bytes), the RLOCs (16 bytes each) and public keys (64 bytes each) of the corresponding MBs, and some additional attributes for traffic engineering (10 bytes). For resilience and load balancing purposes, each EID prefix should have two separate highly available MBs so that we assume two MBs per MBP. This sums up to an average size of 178 bytes per MBP record.

3.3.3.2 Storage Requirements

We estimate the storage requirements of a MB and for the MBP table in FIRMS. The current number of prefixes in the Internet is about $n_{\text{pref}} = 10^6$ [230] while the

current number of hosts is about 10^9 [231]. This leads to an average number of $n_{\text{pref}}^{\text{EIDs}} = 10^3$ hosts per prefix. With the Internet of things and novel applications, we assume that the number of hosts (and EIDs) per EID prefix will dramatically increase in the future. The same holds for the number of EID prefixes.

A MB needs to store on average $n_{\text{pref}}^{\text{EIDs}} = 10^3$ EID-to-RLOC mappings (100 Kbyte) today and a multiple of them in the future. That does not seem a critical value. The MBP table keeps $n_{\text{pref}} = 10^6$ MBP entries (178 Mbyte) and a multiple in the future. Also that seems feasible.

3.3.3.3 Update and Map-Request Loads

We calculate the update load in a MB and for the MBP table in FIRMS. The MB provider may be independent of the ISP of a network. In that case, a customer may change its ISP while keeping its MB provider. Thus, the validity of a MBP can outlast the contract between a customer network and an ISP. A study showed that only 32 percent of small and medium companies changed their provider in 2008 [232]. Thus, we assume that prefix owners change their MBPs every 3 years which also includes key updates for the MBs. With 10^6 prefixes, this leads to an average update rate for MBPs of 38 prefixes or 6.77 Kbytes per hour. Also a much larger multiple seems quite feasible in particular as MBP updates are aggregated and not sent individually as this rough calculation assumes. RIRs have between 1000 and 6500 subordinate LIRs. Hence, they need to push 1.05 Gbytes daily towards their LIRs. This is a large amount of data but breaks down to a continuous upload rate of 12.2 Kbytes/s so that even a large multiple is feasible.

EID-to-RLOC mappings are less stable when nodes become increasingly mobile. We assume that an EID changes its mapping once a month. This is rather an average over all nodes than a typical value since some devices are significantly more mobile than others. A MB in FIRMS which is responsible for a single EID prefix with 10^3 EID-to-RLOC mappings encounters 33 updates per day. This is more than feasible even if a MB stores the mappings for multiple EID prefixes and if the number of EID is orders of magnitude larger.

With FIRMS, the worldwide request load is not problematic since a MR han-

dles only the load originating at an ITR and the MB handles only the request load for a single or a few EID prefixes.

3.3.3.4 Resolution Delay

The resolution delay with FIRMS is rather small. The mappings for most packets are available in the local cache. In case of a cache miss, the MR of the source network queries the MB which should be located in a well accessible place in the Internet. When the map-reply returns, the MR can immediately validate the authenticity of the received data and forward it to the ITR for further use. Thus, FIRMS is rather fast as the resolution delay consists essentially of round trip time to the MB, assuming that local operations are fast.

3.3.3.5 Map-Request Loads

We estimate a lower bound for the expected rate of global map-requests that originate from edge networks due to cache misses. Currently, about 265 million domain names are registered. We use data from VeriSign (.com, .net) and Denic (.de) to estimate the worldwide DNS query load at second-level domains. VeriSign currently experiences an average load of 82 billion DNS queries per day for the 125 million registered .com and .net domains [233]. DENIC has about 11 billion queries for their 16 million .de domains [234]. We now sum up these values and extrapolate them for all 265 million domains, leading to a total average load of 175 billion queries per day, 2 million queries per second, or about 1.6 GBit/s when we assume the size of a map-request packet to be 100 bytes including all headers. Peak request rates are about twice as high. This is the minimum order of magnitude that a future mapping system must sustain for EID-to-RLOC requests. Results from DNS queries are usually cached on different levels of the resolution hierarchy so that the queries at the big registrars heavily underestimate the real number of DNS requests issued by hosts.

Most mapping systems do not have similar hierarchical caching systems and, therefore, the expected rate for worldwide map-requests that cannot be resolved

from local caches is by orders of magnitude larger. High loads of map-requests are problematic for mapping systems with strong hierarchies. Obviously, a single DMB requires a lot of CPU and transmission capacity to answer the worldwide map-requests. However, also map-request forwarding overlays with hierarchical structures such as LISP-ALT are likely to have a few nodes facing an extremely large load of map-requests. This is not only costly for the operator, it also creates a political issue: the worldwide mapping system should not be controllable by a few ISPs. It is better when critical infrastructure is in public hands and causes only moderate operational costs. FIRMS achieves both.

The MBPXs constitute the public infrastructure of FIRMS. They just distribute MBP updates which consume only relatively little capacity, and the MBPXs do not forward map-requests. The load of map-requests at the MBs is relatively low compared with the worldwide load and the expenses for operating the MBs are paid by the prefix owners who want their EIDs be reachable. This seems to be a model with the right economic incentives. A completely distributed mapping system like LISP-NERD is faced only with the load of map-requests that originate from the local network so that the worldwide rate of map-requests is irrelevant.

3.3.4 Proof of Concept

To validate the design and to evaluate the feasibility of FIRMS, we implemented a detailed simulation of FIRMS in the OMNeT++ framework. In addition to the detailed simulation, we also presented an implementation of the basic FIRMS mechanisms in the G-Lab testbed.

3.3.4.1 OMNeT++ Simulation

The following section provides an overview over the implementation of FIRMS in OMNeT++. To test the implementation independent from GLI-Split, we implemented a simple version of a generic map-and-encap Loc/ID split protocol as an operational test environment for FIRMS. The system supports encapsulation and forwarding using FIRMS capabilities of packet forwarding. It compromises

three basic components: ITR, ETR, and a client. Clients communicate over ITRs or ETRs. A listen-only client waits for packets arriving from the ETR and simply deletes them on reception. A sending-only client sends packets with randomly chosen receiver EIDs to its ITR in specific distributed intervals. The ITR receives packets from a client, performs a cache-lookup for the receiver's EID, encapsulates the packet, and eventually forwards the packet to the correct ETR. In case of a cache-miss, the original packet is attached to the solicited map-request message that is then sent to the MapResolver. The ETR receives packets from an ITR, performs decapsulation, and eventually forwards the packet to the correct client.

For the functional simulation of FIRMS, we implemented all important FIRMS nodes:

PrefixOwner The PrefixOwner is a very simple component of the FIRMS simulation model and models the role of a prefix owner in FIRMS. The PrefixOwner is configured with specific prefixes, locator lists, MapBases and MapBasePointerExchangeNodes. In predefined intervals, the PrefixOwner can register or unregister its prefix information in FIRMS.

MapBase The MapBase is the authoritative source for mapping information in FIRMS. It presents a simple query interface to the FIRMS components and is responsible for providing mapping information if requested by the MapResolver. The information base in a MapBase node is set directly by PrefixOwner nodes. In order to be reachable throughout the Internet, a MapBase has a globally accessible address.

MapBasePointerExchangeNode The MapBasePointerExchangeNode is responsible for the reliable propagation of both mapping update and structural update messages. The module can operate in two different modes: RIR-mode and LIR-mode. With these two modes we model a hierarchical system of MapBasePointerExchangeNodes. Both node modes are implemented in a single module and can be selected by configuration.

MapResolver The MapResolver combines the information of the MBP table with the MapBases. It must ensure that its copy of the MBP table is up-to-date and that map-requests are answered quickly. This component has to remain extremely stable and responsive under high system load. Therefore, cache, timer, and state management must be kept as simple as possible. The main responsibilities of the module are answering queries from stub resolvers and updating the internal MBP table.

Test scenarios To verify the design and the functionality of our FIRMS mapping architecture, we evaluated several test cases. We verified that our implementation works with and without caching by sending packets between different clients. We also made sure that the update-mechanisms of the FIRMS nodes works as expected. In additional scenarios, we could show that FIRMS continues to work when different MapBases fail.

3.3.4.2 Prototype Implementation in G-Lab

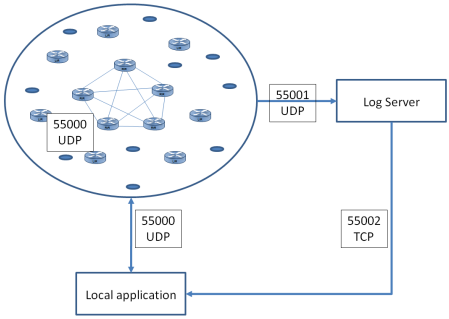


Figure 3.20: *Structure of FIRMS implementation in the G-Lab demonstrator.*

After we verified that the FIRMS architecture works in the simulator, we implemented the basic FIRMS functionality in a distributed Java program to enable

further tests and verification in a real world environment. For this purpose, we use the German Lab (G-Lab), which is an experimental facility with the goal to foster experimentally driven research to exploit future Internet technologies. Our Java-based implementation evaluated different parts of the architecture: We provide the mapping distribution structure, consisting of RIRs, LIRs, MRs, and MBs. In addition, we implemented a log server that receives status messages of all components to monitor the infrastructure. The components communicate among each other using UDP and TCP as depicted in Figure 3.20.

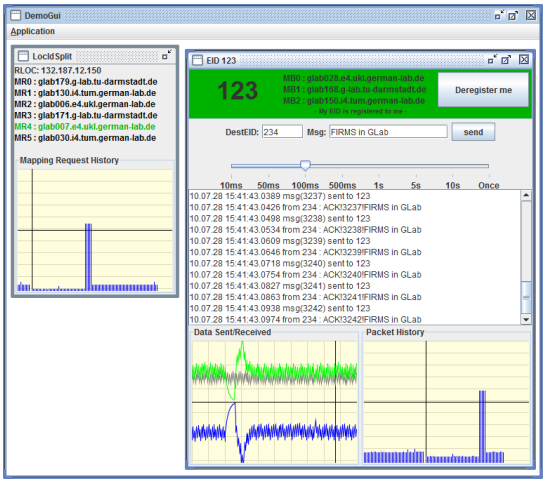


Figure 3.21: *Application-based Loc/ID split.*

For demonstration purposes, we provide a local application that can act both as prefix owner and register mapping for prefixes and also as an ITR that requests mappings from the FIRMS mapping system. As the demonstrator does only implement the FIRMS mapping system but not the GLI-Split, we use a simple application-based Loc/ID split to show the capabilities of FIRMS. The GUI of the application is shown in Figure 3.21. This applications runs on different

nodes and can register several EIDs in the mapping system, using the nodes IP address as locator. It shows the currently used MR and the mapping-lookup delay in the left LocIdSplit-frame. In the larger frame, an EID application is shown that communicates with another EID located on a different computer. Each message must first look up the locator for the destination ID. The round-trip time delay until an ACK message is received for each packet is plotted on the right side. The left chart shows the rate of incoming and outgoing packets to this EID.

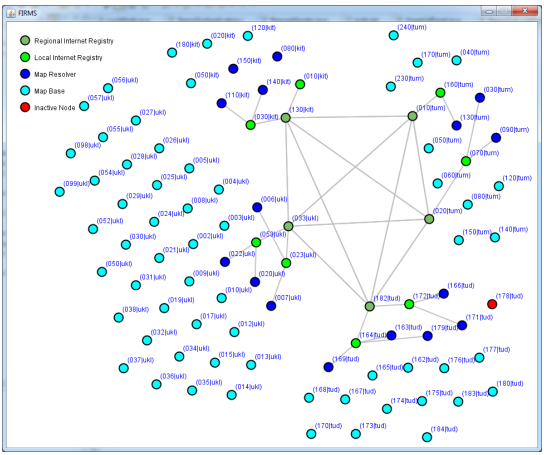


Figure 3.22: *FIRMS in the G-Lab experimental facility.*

Another application that connects to the log server and visualizes the status of the FIRMS nodes is depicted in Figure 3.22. The Java-based FIRMS prototype is deployed on 100 G-Lab nodes that are depicted in the GUI. Request activity on every node and the current status is also shown.

We demonstrated the FIRMS functionality [38] using different scenarios. First, we showed that in our example application, mobility of EIDs between nodes can easily be supported. We also demonstrated different failure scenarios. When either a MB or an MR fails, mapping continues successfully after a short delay

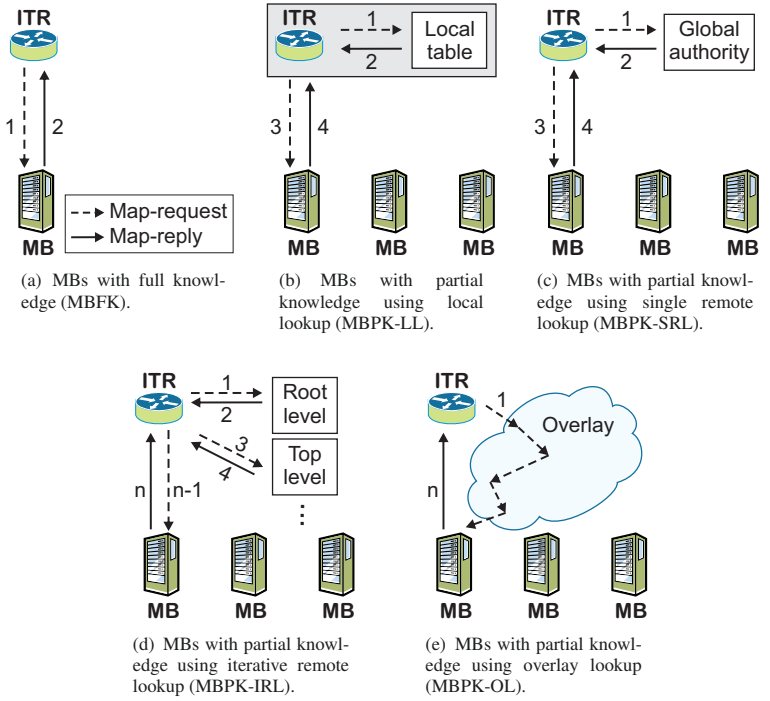
when built in resilience mechanisms kick-in.

During the simulation and implementation of FIRMS, several drawbacks of the architecture were discovered and new ideas evolved, so we often went back to our specifications and improved the architecture based on these insights.

3.3.5 Classification of Mapping Systems

Besides the development of our own FIRMS system, we analyzed all other proposed mapping architectures. For a better understanding of the general design-options, we suggest a taxonomy of different mapping systems. An architecture depends on the structure of EIDs. EIDs in the Loc/ID split context should be globally unique. Their uniqueness can be achieved through administrative or statistical means. IP or Ethernet addresses are examples for the first category. They are hierarchically assigned from number authorities to smaller and smaller organizations and finally configured to individual nodes. Alternatively, EIDs may be randomly created like in HIP [189]. If they are sufficiently long, the probability for the creation of the same EIDs is very small. These EIDs are unstructured and we call their address space flat. Also semi-structured addresses have been proposed, which combine hierarchically assigned prefixes and random suffixes [199, 222, 235]. A set of unstructured EIDs cannot be aggregated by a common prefix. Therefore, each EID in the set needs its own EID-to-RLOC mapping, even if all of them are located behind the same RLOC. When structured EIDs with a common prefix have the same RLOC, the mapping information can easily be aggregated to an EID-prefix-to-RLOC mapping. This is attractive as it saves memory and communication overhead. Aggregation is more difficult when multihoming or mobility of individual EIDs is to be supported. The length of EID prefixes in mappings (maximimum aggregatable EID blocks) could be minimized by allowing indicated exceptions, i.e., holes in the aggregated EID prefix [236].

In the following, we derive an abstract hierarchical classification of mapping systems that were recently proposed and discussed in a Loc/ID split context. We briefly explain their basic structure and operation.



(a) MBs with full knowledge (MBFK).

(b) MBs with partial knowledge using local lookup (MBPK-LL).

(c) MBs with partial knowledge using single remote lookup (MBPK-SRL).

(d) MBs with partial knowledge using iterative remote lookup (MBPK-IRL).

(e) MBs with partial knowledge using overlay lookup (MBPK-OL).

Figure 3.23: *Structure and operation of different mapping system classes.*

3.3.5.1 Map-Bases (MBFK, MBPK)

We define a map-base (MB) as a node or distributed system that is the authoritative source of EID-to-RLOC mappings. In general, MBs must have globally reachable RLOC addresses so that they can be contacted without another mapping lookup.

There are basically two options to implement MBs. One option is to implement a central MB to store the EID-to-RLOC mappings for all existing EIDs in a single

MB with full knowledge (MBFK). As an alternative to MBFKs, EID-to-RLOC mappings may be stored in distributed MBs each of which holds only partial knowledge (MBPK).

MBFKs may be replicated to multiple mirrors for resilience and load-balancing, and to bring the MBs closer to the ITRs. An ITR needs to be configured with the address of at least one MBFK. It sends map-requests and receives replies directly from that MB, as visualized in Figure 3.23(a). Changes of EID-to-RLOC mappings lead to a large amount of update traffic and frequent changes in the MBs. Therefore, frequent mapping changes should be avoided. Packet relaying with MBFKs is simple. After a cache miss, ITRs can immediately relay packets to the MBFK. The receiving MB can serve as a proxy ITR and encapsulate the received data packets towards their destinations.

In contrast, MBPKs help to keep mapping updates local and possibly facilitate mobility support using Loc/ID split [27, 209]. They may be operated on behalf of EID owners or on behalf of networks or autonomous systems (ASes). This approach also has the advantage that only local MB operators have control over the mappings.

3.3.5.2 Discovery Options for MBPKs

When an ITR needs a mapping for a certain EID, it needs to know which MBPK to query. There are several discovery options to find the correct MBPK.

Local lookup (MBPK-LL) The fastest option to find the appropriate MB is to configure ITRs with a table that stores EID(-prefix)-to-MB information. It points to the MB that stores EID-to-RLOC mappings for certain EID-prefixes. Only one local lookup suffices to find an appropriate MB. The actual EID-to-RLOC mapping is obtained with a single query over a direct path between ITR and MB which keeps the mapping delay short. This is visualized in Figure 3.23(b).

The challenge is the composition and distribution of the EID-to-MB table, and to keep it up to date. To keep the table small, mapping aggregation is crucial, i.e., EID-to-RLOC mappings for EIDs from a common prefix should be provided by

the same MB. The EID(-prefix)-to-MB information in the local tables of ITRs is relatively stable so that the update load is low.

With MBPK-LLs, a packet relaying service can easily be implemented. After a cache miss, the ITR looks up the address of the appropriate MB to send a map-request. The ITR may then relay packets to this MB which stores the matching EID-to-RLOC mapping and can forward the packet like an ITR.

Single remote lookup (MBPK-SRL) Instead of a local table, a global authority can be used. In this approach, the knowledge about which MB is responsible for which EIDs is maintained and stored by a global authority. To retrieve an EID-to-RLOC mapping for a specific EID, an ITR must first query the global authority for an EID(-prefix)-to-MB mapping and then send a map-request to the respective MB. This is visualized in Figure 3.23(c).

Single remote lookups for MB discovery introduce more communication overhead than local lookups, but ITRs may store the obtained EID(-prefix)-to-MB information in a cache to reduce communication overhead and delay for future requests. This is similar to the caching of EID-to-RLOC information. EID(-prefix)-to-MB information is expected to change less frequently than EID-to-RLOC information, so that this information can be cached for a longer time.

With MBPK-SRLs, the implementation of a packet relaying service is problematic because packets need to be stored or dropped until the ITR has retrieved the EID-to-MB information from the authority. As soon as the ITR knows the appropriate MB, it can relay packets to the MB that may forward the packets to their destination.

Iterative remote lookup (MBPK-IRL) The MB that is responsible for a certain EID may be found iteratively, i.e., in a similar way as in the Domain Name System (DNS). A level-0 authority (root level in DNS) returns a EID(-prefix)-to-level-1 mapping to the ITR. A level-1 authority (top level in DNS) returns a EID(-prefix)-to-level-2 mapping to the ITR, etc. Finally, the EID-to-level-n mapping designates the actual MB. This is visualized in Figure 3.23(d). Again, caching of

entries is possible to save communication overhead for future map-requests.

Packet relaying is difficult with MBPK-IRLs for the same reasons as for MBPK-SRLs. A slow, remote query is required to discover the appropriate MB. During this time, packets need to be stored or dropped by the ITR.

Overlay lookup (MBPK-OL) Our last proposed class of MBPK discovery options is the usage of overlay networks. In architectures that fall into this class, the ITR finds the appropriate MB through an overlay network. A map-request is sent into an overlay network where it is forwarded to the appropriate MB which responds to the ITR with a map-reply. Each ITR must be configured with at least one entry node in the overlay network. There are many different implementation options for the overlay network, which are classified in the next section (Section 3.3.5.3).

The resolution delay of MBPK-OL is rather large by design for two reasons. Map-requests are sent over possibly multiple hops within an overlay network to the appropriate MB instead of using the direct path as other mapping systems do. Moreover, most MBPK-OL implementations require that overlay nodes process the map-requests on the application layer to forward them to the appropriate next-hop, which is time-consuming. As a consequence, transport over the overlay network is usually much slower than over a direct path.

In most MBPK-OLs, a packet relaying service can be implemented. After a cache miss, ITRs may send EID-addressed packets through the overlay network where they eventually reach a MB that can forward the packets to the destination. When first packets are relayed over the overlay network and subsequent packets are tunneled by the ITR, packet reordering can occur due to the large difference in the transportation time over the different paths. This may cause problems for some applications.

The overlay network is a vital part of the mapping system and should be run on a trusted infrastructure. Operators of nodes participating in an MBPK-OL may control transiting sensitive traffic from other participants. This could be a threat to participants whose map-requests are carried over nodes that do not have at

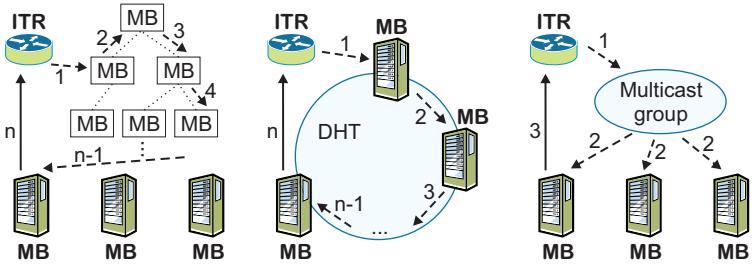
least indirect business relations with them. They require that these nodes reliably process and forward their map-requests. Carrying traffic from participants with whom they do not have business relations is also a burden for the operator of an overlay node. The node is expected to process and forward the requests without receiving revenues from them although the data rate may be high, especially if also packets are relayed. Thus, new business models are needed for the deployment of MBPK-OLs.

Parts of the overlay network can fail or be attacked. Since customers cannot simply increase the availability of the overlay network by replication, MBPK-OLs require special backup concepts to avoid service degradation in failure cases.

Despite these shortcomings, several MBPK-OLs have been proposed in the past. Many of them do not need an infrastructure that is managed by a global authority, which is very appealing especially in the prototype stage.

3.3.5.3 Lookup Overlays for Discovery of MBPKs

In the following, we describe different approaches to implement MBPK-OL.



(a) MBs with partial knowledge using hierarchically structured overlay (MBPK-HSO).

(b) MBs with partial knowledge using distributed hash table (MBPK-DHT).

(c) MBs with partial knowledge using multicast overlay (MBPK-MCO).

Figure 3.24: *Special cases of lookup overlays.*

Hierarchically structured overlay (MBPK-HSO) In a hierarchically structured overlay network, nodes represent EID-prefixes and are arranged in a hierarchical manner with regard to these EID-prefixes. The MBs are connected to the nodes with the most-specific EID-prefixes. Figure 3.24(a) visualizes the forwarding of a map-request in such a overlay network. At first, map-requests travel up the hierarchy if needed and then down towards the nodes with the most specific EID-prefixes over which they finally reach the appropriate MB.

Distributed hash tables (MBPK-DHT) A distributed hash table (DHT) consists of connected MBs and uses a function that determines at which MB the EID-to-RLOC mapping for a special EID is stored. When map-requests are injected into the DHT, each node knows how to forward them to neighboring nodes so that the map-request eventually reaches the appropriate MB. This is visualized in Figure 3.24(b).

Multicast overlay (MBPK-MCO) In a multicast overlay network, the EID-space is partitioned and multicast groups are created for each of the EID-subsets. MBs with EID-to-RLOC mappings subscribe to all groups which cover one of their EIDs. The ITR is configured with all multicast groups and sends map-requests to the multicast group the EID belongs to. Thus, the map-request is carried to all MBs containing EIDs for the same multicast group. The MBs with the matching mapping send a map-reply back to the ITR. This is illustrated in Figure 3.24(c).

3.3.5.4 Overview

Figure 3.25 shows a summary of our proposed classification. Mapping systems consist of MBs with either full or partial knowledge. For the latter, we distinguished the way how ITRs determine the appropriate MB for a given EID. The options are local lookup, single remote lookup, iterative remote lookup, and overlay lookup. The overlay network may be a hierarchically structured overlay, a distributed hash table, or a multicast overlay. This classification does not claim

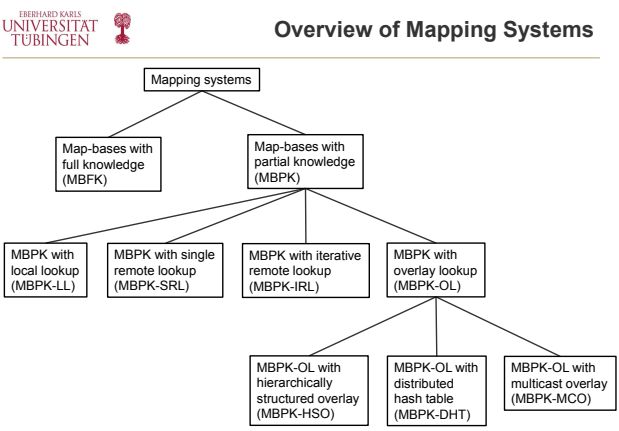


Figure 3.25: *Hierarchical classification of mapping retrieval.*

to be complete, but it describes the existing proposals that we review in the next section well.

3.3.6 Related Work on Mapping Systems

We review mapping systems that were presented in the context of Loc/ID split based Internet routing. We classify them into the categories presented in Section 3.3.5 and discuss their properties.

3.3.6.1 Map-Bases with Full Knowledge (MBFK)

LISP-NERD The "Not-so-novel EID to RLOC Database" for LISP (LISP-NERD) [181] is a mapping system that is primarily designed to avoid packet drops. To achieve this, ITRs get all mapping information in advance. One or several authorities assign EIDs to organizations and run a MB (called NERD) with authoritative mappings. An ITR is configured with the addresses of possibly several authoritative NERDs and pulls the entire mapping information from them upon system start. To facilitate incremental updates, changes to the NERD are

associated with a version number so that ITRs need to download only new information to their database. All information sent from the NERDs to the ITRs is digitally signed using X.509 certificates. As all mappings are locally available at the ITRs, cache misses cannot occur. Querying delay and packet loss cannot happen so that a packet relay service is not needed. In an architecture with a fine mapping granularity, the distribution of all mapping-updates to all NERDS raises scalability concerns.

APT "A Practical Tunneling architecture" (APT) [220] is an architecture with its own mapping distribution system [176] that maps EID prefixes to RLOCs. APT's mapping system assumes that each ISP has a default mapper (DM), i.e., a mirror containing the global mapping information. DMs of neighboring ASes know each other and exchange mapping information via a mapping dissemination protocol using signed messages. The prefix owners inject the mapping information into the DMs of their ISPs, from where information is pushed to neighboring DMs. When an ITR encounters a cache miss for a packet destined to an unknown EID, the ITR sends the packet to the DM of its own domain. The DM chooses a single RLOC, returns the appropriate mapping to the ITR, and tunnels the packet to an appropriate ETR. Thus, APT provides a packet relay service. The DMs can also be used to protect against the failure of ETRs. By announcing a high-cost route towards every ETR inside its AS, the DM attracts traffic addressed to a non-working ETR, and re-encapsulates it towards an alternative ETR.

3.3.6.2 Map-Bases with Partial Knowledge using Local Lookup (MBPK-LL)

Our proposed FIRMS architecture is the only mapping system with partial knowledge that uses local lookup.

3.3.6.3 Map-Bases with Partial Knowledge using Single Remote Lookup (MBPK-SRL)

The "Hierarchical Internet Mapping Architecture" HiiMap [237] is the only mapping system in this category. It assumes that EIDs are under the control of a region which may be, e.g., a country. Each regional authority has a MB that stores all EID-to-RLOC mappings for all its EIDs. A single global authority stores an EID-to-MB mapping, a so-called "regional prefix" for each EID. An ITR queries the global authority for the "regional prefix" and after its reception, the ITR queries the regional authority for the EID-to-RLOC mapping.

HiiMap can support flat EID spaces as it does not take advantage of EID prefix aggregation. However, this feature makes it only as scalable as MBFKs. The global authority stores a regional prefix per EID which leads to large storage requirements. Furthermore, the global authority may be a performance bottleneck for updates and requests. If the ITR has no regional prefix for an EID in its local cache, it must query both the global and the regional authority to obtain the EID-to-RLOC mapping which leads to increased lookup delay.

3.3.6.4 Map-Bases with Partial Knowledge using Iterative Remote Lookup (MBPK-IRL)

DNS has been proven to be a powerful and scalable architecture, and recently, it has also been made secure [238]. Clients can trust the received data when they are signed by a trusted authoritative DNS server. However, if the client does not trust the public key of the authoritative DNS, it must first validate that key before it can validate the actual data. This iterative validation process up the trust chain to a common trust anchor adds delay to the mapping lookup. None of the following MBPK-IRLs support packet relaying.

One-phase lookup using reverse DNS The one-phase lookup as presented in [239, Scheme 1] assumes that the reverse DNS reply contains both a pointer resource record (PTR-RR) and an A-RR for the ETR of the queried IP

number, i.e., an RLOC for the requested EID. Normally, A-RRs provide IPv4 addresses for the DNS names. In this context, they provide the IPv4 RLOCs of the ETRs for the EIDs. The prefix owner can set up an authoritative DNS server returning the A-RR with the RLOC information for his EID prefix and register the address of this delegation server with the authority from which it has received his EID prefix. Thereby, the prefix owner has still control over the mappings. This idea has been sketched for the LISP context [239, 240] but it did not prevail since the existing DNS should not be burdened with another heavy service.

Two-phase lookup using reverse DNS Intermediate nodes may perform a two-phase lookup to retrieve an EID-to-RLOC mapping. An ITR first makes a reverse lookup to get the DNS name for an EID (which must be an IP address), and then it makes a forward lookup for the RLOC of that DNS name. This requires the use of reverse DNS, a PTR-RR that provides a DNS name for each EID, and the definition of a new locator resource record (L-RR). This approach is quite slow since it requires two lookups and it raised security concerns.

Mapping lookup using only DNS names The Identifier/Locator Network Protocol (ILNP) [241, 242] defines four new resource record (RR) types for DNS [243]: ID, L32, L64, and LP. The ID-RR stores an EID for a special DNS name. RLOCs are stored as L32-RRs and L64-RRs, depending on whether ILNPv4 or ILNPv6 locators are to be retrieved. LP-RR stands for locator pointer RR and is used to provide multihoming and mobility in ILNP [244]. Hosts use the fully qualified domain name to retrieve the ID-RR from the DNS. With the ID-RR, hosts further query the DNS to retrieve the corresponding L32-RR or L64-RR. With this information, the hosts compose the appropriate IP numbers. In case of multihoming or mobility, the ID-RR is used to retrieve the LP-RR which is further used and resolved to L32-RRs or L64-RRs.

This principle works well with ILNP but not with other routing architectures where intermediate nodes add locator information as the lookup requires a DNS name which is not contained in the IP packet header.

Use of DNS for HIT-to-IP lookup in HIP HIP requires a mapping system to find an IP address for a given HIT. The authors of [245] propose to use the DNS system to find the IP addresses for HITs. For reverse DNS, the authors of [246] postulate a "hit-to-ip.arpa" domain in which HITs are denoted like IPv6 addresses within "ipv6.arpa". Since HITs are not hierarchically structured, all HITs need to be known by top-level servers that are run by authorities. The authors give evidence that DNS servers are powerful enough for their purpose. Since improved mobility is an objective of HIP, HIT-to-IP mappings are likely to change often. As updates of DNS records take orders of magnitude longer than retrievals, a two-level hierarchy is introduced. The entries in the top-level DNS servers refer to second-level DNS servers. These entries are likely to remain stable for a long time. Thus, top-level servers experience fewer updates which reduces the infrastructure expenses for authorities. This also provides direct control over the actual HIT-to-IP mapping to the HIT owner which is important to support mobility.

LISP-TREE LISP-TREE [201] uses DNS technology, but stores EID-to-RLOC mappings on a different infrastructure. It assumes that the EID address space is partitioned among regional EID registrars (RERs) which allocate parts of their EID space to local EID registrars (LERs). LERs further allocate EID space to other LERs or customers. To be compliant with LISP, EID-to-RLOC mappings are stored by authoritative ETRs which serve as MBs for EID-prefixes.

LISP-TREE uses a tree-like overlay structure of LISP-TREE servers (LTSs). They run a DNS service for EIDs and assist ITRs to find the authoritative MB for a given EID. The root LTSs are run by the RERs and store pointers to the LTSs for their /8 prefixes (at most 256). Lower level LTSs are managed by the corresponding LERs and hold pointers from more specific EID-prefixes to lower level LTSs that are responsible for a smaller subset of the EID address space defined by the delegated prefix. LISP-TREE uses the generic LISP-MS mapping system interface so that MSes constitute the leaves of the LTS tree.

ITRs are configured with the root LTSs and iteratively or recursively query LTSs to eventually receive the address of the correct MB. Then, they query it to

get the EID-to-RLOC mapping from the MB. Intermediate results from LTSs are cached so that the ITR must query the root LTSs only rarely. The authors have shown that only the iterative mode is scalable. Security in LISP-TREE is provided by the use of DNSSEC [238]. The layered mapping system (LMS) discussed in [247] and presented in [248] is very similar to LISP-TREE and therefore not further discussed here.

LISP delegated database tree LISP Delegated Database Tree (LISP-DDT) [249] is a hierarchical mapping system and the logical successor of LISP-TREE. In LISP-DDT, LTS are called DDT nodes and do not use DNS technology. However, the abstract structure of LISP-TREE is preserved, i.e., the DDT nodes form a hierarchical EID-prefix tree. At the lowest level of the database tree, DDT map-servers hold EID-prefix-to-MB mappings. DNS security mechanisms cannot be used directly because LISP-DDT does not use DNS technology. However, security is provided using pre-shared keys between DDT nodes, which works similar to DNSSEC and LISP-SEC [250] mechanisms. LISP-DDT in combination with LISP-MS is currently the preferred mapping system for LISP.

Distributed real time mapping system for IVIP and LISP IVIP [186, 251] is an alternative Loc/ID split architecture with its own "Distributed Real Time Mapping system" (DRTM) [252]. The EID space is partitioned in mapped address blocks (MABs) by MAB operating companies (MABOCs). The resulting MABs are assigned to user organizations who can further partition their MABs into micronets which are arbitrarily long EID prefixes.

MABOCs store EID(-prefix)-to-RLOC mappings on behalf of the prefix owners on authoritative query servers (QSAs). Each QSA is only authoritative for a subset of all MABs. Resolver query servers (QSRs) use a DNS-based mechanism to resolve the EID(-prefix)-to-QSA mapping and then query the appropriate QSA for the EID(-prefix)-to-RLOC mapping. ITRs communicate with QSRs directly or use a cascade of caching query servers (QSCs) to speedup consecutive lookups. QSAs internally store the last mapping requesters. In case of a mapping

change, this enables QSAs to flush the cache entries in the QSRs, QSCs, and ITRs to force a new mapping lookup, and to reduce signaling complexity.

In contrast to other approaches, only one RLOC is stored per micronet. IVIP assumes that edge networks hire third parties to effect real-time updates to the mapping system to take advantage of multihoming for inbound traffic engineering and service restoration in case of ITR/ETR failures. The author of [252] states further that resolution process of DRTM is fast enough so that ITRs can buffer initial packets of a flow without experiencing buffer overflow.

ID mapping system The ID mapping system (IDMS) [253] uses an extended version of DNS. IDs in IDMS are hierarchically structured and have a similar format like e-mail addresses, i.e., *hostname@authority*. EID-suffix-to-MB mappings are stored in the extended DNS. MBs are implemented as so-called ID mapping servers and provide EID(-suffix)-to-RLOC mappings. Each authority provides ID mapping servers and updates the EID-to-RLOC mapping in real-time. Mappings stored in DNS are stable while mappings in ID mapping servers may change frequently due to host mobility.

The scalability of the system is limited by DNS and the ID mapping server implementations. The latter gives local authorities the freedom to choose a scalable solution for their own EID space, i.e., each local authority runs its own ID mapping server implementation. Local authorities are also responsible to implement appropriate resilience mechanisms for their servers. Security is provided through PKI and digital signatures.

Mapping lookup for intercepted DNS queries Before a host starts communication with a remote system, it usually resolves a DNS name to an IP address. ITRs may intercept these DNS queries, query a new EID-to-RLOC mapping for the contained DNS names, and store the DNS reply in their cache so that packets sent to corresponding EIDs do not encounter a cache miss. This idea is not a mapping system but a DNS scheme for LISP [239, Scheme 2], i.e., a prefetching technique.

3.3.6.5 Map-Bases with Partial Knowledge using Hierarchically Structured Overlay (MBPK-HSO)

LISP+ALT In LISP Alternative Logical Topology (LISP+ALT) [177] so-called ALT routers build a semi-hierarchically structured overlay network: the ALT. ALT routers are associated with EID prefixes and connected in a semi-hierarchical manner with respect to these prefixes. Shortcuts are possible on the same hierarchy level. A leaf ALT router connects to all MBs that store EID-to-RLOCs for its EID prefix. Even though the architecture is strongly aggregation oriented, there are no root nodes. ALT routers communicate to neighboring ALT routers via BGP and exchange aggregated EID prefixes that can be reached through them. In contrast to regular BGP, ALT routers possibly aggregate prefixes received via BGP before forwarding them, which makes the ALT scalable.

The ITR addresses map-requests to the queried EID and sends them to an ALT router. The map-request is forwarded through the ALT overlay based on the EID. Eventually, the map-request reaches the appropriate MB which returns a map-reply directly to the ITR.

The operation of LISP+ALT is very efficient. The ALT routers are directly connected over tunnels using generic routing encapsulation (GRE). ALT routers simply forward packets addressed to EIDs according to their routing tables that are composed with the help of BGP on the basis of the EID prefixes associated with the ALT routers. Thus, ALT routers do not need to process the packets on the application layer like it is done in other MBPK-OL proposals. In case of a cache miss at the ITR, packets can also be carried over the ALT, but this is not recommended in the current LISP proposal [177, Section 3.3].

LISP-CONS The "Content distribution Overlay Network Service for LISP" (LISP-CONS) [178] was a predecessor to LISP+ALT. LISP-CONS does not necessarily use BGP for communication between nodes of the hierarchy. Map-replies are returned from the ETRs back to the ITRs over the overlay network which is also different from LISP+ALT. LISP-CONS also allows carrying mapping requests and packets over the overlay network.

A hierarchical mapping system for LISP LISP-HMS [254] is a hierarchical mapping system which combines BGP and one-hop DHTs. MBs are called mapping servers and are responsible for a pre-defined mapping domain, i.e., a set of EID-prefixes. The MBs form a one-hop DHT called destination mapping server (DMS) which stores all mapping information of an AS. To provide mappings between ASes, forwarders further aggregate the mappings of the DMSs, and resolvers exchange EID-prefix-to-AS mapping information using BGP. Thus, the resolvers form an overlay network for inter-AS mapping information.

The resolution process works as follows. ITRs are configured with MBs. If the MB does not know a requested mapping, the request is forwarded to the associated DMS. If the requested EID belongs to the same AS, the appropriate MB in the DMS replies directly to the ITR. Otherwise, the map-request is forwarded to the AS' resolver which then forwards the request to the correct AS, DMS, and MB which replies to the ITR. Then, the ITR can start sending packets.

When a new device joins an AS, it registers at its ITR. The ITR registers the EID-to-RLOC mapping at its MB. The MB updates the mappings at its DMS and reports its aggregated mappings to the forwarder. The forwarder aggregates its information base further and reports the mappings to its resolver. Then, the resolver propagates the new mapping information to the other ASes using BGP. Depending on the granularity of the aggregation, mapping information at upper levels stays stable. As BGP is used between ASes, security mechanisms of BGP can be utilized to protect the dissemination of mapping information.

ID/Locator distributed mapping server The mapping and relaying system presented in [255] has AS-specific DHT-based MBs that store the EID-to-RLOC mappings for the EIDs hosted in the AS. ITRs query their local MB in case of cache misses. Map-requests that cannot be served directly are forwarded to a so-called border server. Border servers of different ASes exchange the EID-prefixes under their control via BGP and build a hierarchically structured overlay for which EID aggregation is prerequisite for scalability.

IRON The "Internet Routing Overlay Network" (IRON) [256, 257] assumes that the global EID space is partitioned among virtual service providers (VSPs) in aggregated prefixes (AP) which are further partitioned in client prefixes (CP) and eventually delegated to end-hosts. A VSP forms an IRON instance which comprises of IRON agents, i.e., IRON servers, IRON clients and at least one IRON relay. IRON clients fulfill the function of ITRs and ETRs. IRON servers store the EID(-prefix)-to-RLOC information and announce their stored mappings to their IRON relays using eBGP, i.e., they are the MBs. Each IRON relay connects to the Internet as an AS using BGP and forms an overlay network with the other IRON relays. They aggregate the mapping information of all MBs of their respective IRON instance and internally store the EID(-prefix)-to-MB information. On the IRON relay overlay network, EID(-prefix)-to-IRON-relay information is exchanged between the IRON relays using iBGP.

When an ITR encounters a cache miss, it relays the packet without locator information to the MB of its VSP which forwards it to one of its IRON relays. The IRON relay natively forwards the packet to the IRON relay of the destination VSP which further tunnels it to the appropriate MB that tunnels it to the ETR of the destination network. Explicit map-requests do not exist. However, the MB responsible for the destination EID sends a "route optimization" message to the ITR so that the ITR can tunnel further packets directly to the ETR. Thus, route optimization messages are similar to map-replies.

Realm zone bridging server RZBS is the mapping system of MILSA [258, 259]. Apparently, RZBS designates both the name of the mapping system architecture and a MB. The system shares some similarities with DNS. IDs in RZBS are structured like URIs, and the ID space is partitioned in domains, which are called realms in RZBS. Each realm can be further partitioned in subrealms, subsubrealms and so on. MBs are responsible to store EID-suffix-to-MB and EID-to-RLOC information. Thus, MBs form a hierarchically structured forest of realm trees. To connect the realm trees, DNS is used.

RZBS achieves scalability and resilience through replication of MBs at dif-

ferent hierarchical levels. Security is realized through trust relationships between realm trees, i.e., each subtree trusts its root node and may trust its neighboring tree directly. The trust relationships influence how signaling messages are forwarded inside the overlay network.

3.3.6.6 Map-Bases with Partial Knowledge using Distributed Hash Tables (MBPK-DHT)

LISP-DHT LISP-DHT stores mappings in a distributed hash table (DHT) [179]. MBs join a modified Chord ring as so-called service nodes to build a DHT. They have an ID that determines their position within the ring structure. The ID of a service node is the highest EID of the EID prefix for which it is responsible. Thus, service node IDs and EIDs are taken from the same number space. The unhashed EIDs of map-requests are used for message forwarding in the DHT. Thus, a map-request is carried within the DHT over several hops to the service node with the smallest ID that is at least as large as the requested EID. This service node then replies the mapping to the requesting ITR.

Prefix owners keep control over the mappings as they are kept local in the service nodes. If a service node is responsible for several EID prefixes, it has several IDs and is connected to the Chord ring at several positions. To prevent malicious nodes from EID prefix hijacking, joining service nodes must be authenticated as the rightful owners of their EID prefixes using X.509 resource certificates [226]. To inject map-requests, ITRs join the Chord ring as stealth nodes which do not participate in message forwarding or other critical tasks.

ER+MO In [260], a mapping and relaying system is presented which combines techniques similar to LISP+ALT and LISP-DHT. A customer network stores the mappings for its EIDs in a MB which is part of a mapping overlay (MO) very similar to LISP-DHT. However, Kademlia is used instead of Chord as DHT, and mappings are stored per EID instead of per EID prefix. Thus, a MB joins the DHT as a service node once for each EID under its control. This induces significant management overhead when multiple nodes join or leave. The relaying system

works similarly as LISP+ALT. It consists of EID routers (ER) which learn EID-prefixes from ETRs via BGP and relay packets if needed.

DHT-MAP DHT-MAP [202] also supports a flat identifier space. Each AS operates a MB – called resolver – which stores the AS-specific EID-to-RLOC mappings for the EIDs supported within the AS. Resolvers of different ASes are connected to a content addressable network (CAN) which is a special type of DHT in which EID-to-MB mappings are stored. ITRs of an AS are connected to a resolver. When an ITR encounters a cache miss, it sends a map-request including the packet to the resolver. If the resolver knows the EID-to-RLOC mapping, it tunnels the packet to the ETR and returns a map-reply to the ITR; otherwise, it sends the map-request including the packet into the CAN. The CAN node that is responsible for the requested EID may have different EID-to-MB mappings, chooses one of them, and forwards the map-request to this resolver. This resolver has an appropriate EID-to-RLOC mapping, tunnels the packet to the ETR, and sends a map-reply to the requesting resolver which forwards it to the requesting ITR. All subsequent packets are tunneled directly to the destination ETR.

When a new device joins an AS, it registers at the ITR. The ITR registers the new EID-to-RLOC mapping at the resolver and the resolver registers the EID-to-resolver mapping in the CAN. Therefore, DHT-MAP's resolvers know only the RLOCs of one AS. Since the CAN node that is responsible for the EID forwards map-requests only to a single resolver, the ITRs receives only the RLOCs of a single AS for an EID. This is a strong limitation for multihoming.

DHT-MAP is able to relay packets, but the extensive path stretch for first packets causes long delays and mapping lookups. This leads to more relayed traffic than for short mapping lookups so that substantial forwarding capacity is needed in the CAN. DHT-MAP relies on the resilience features of the DHT to carry the map-request to backup nodes in the DHT and on resilience features in the AS to carry map-requests to backup resolvers.

HIP-DHT HIP-DHT [261] proposes to use a DHT for looking up HIP-related information based on a HIT (see Section 3.3.6.4). The authors specify how their concept works with OpenDHT. However, the authors also list security concerns pointing out potential map-reply spoofing attacks leading to stale information or mapping pollution since authentication is not required to register new or already existing mappings in the system.

RANGI In the "Routing Architecture for the Next Generation Internet" (RANGI) [222], the name space of host identifiers (HI) is partitioned by prefixes among administrative domains (ADs). HIs consist of two parts: the globally unique AD ID which is assigned by a central authority like IANA and a cryptographical part that is generated as a hash containing the AD ID and a public key value like in HIP. An AD takes care that the HIs under its control are unique. RANGI uses a hierarchical DHT to map HIs to RLOCs. A top-level DHT guides map-requests to bottom-level DHTs using the AD ID in the HI. The bottom-level DHTs uses the unstructured cryptographical part of the HI to resolve the mapping and send map-replies to ITRs.

CoDoNS CoDoNS stands for cooperative domain name system [262]. It is proposed as a substitute for the DNS and it is implemented based on a DHT called Pastry and enhanced using a proactive caching layer called Beehive. CoDoNS replicates mapping information across the DHT to achieve an access time of practically $O(1)$. Large organizations should participate in CoDoNS with at least two nodes. These nodes store data from other organizations and the organization's own data are probably stored on nodes of other organizations. This property is hard to accept in practice which is also a general argument against the straightforward use of DHTs as a mapping system.

LISP-SHDHT The main objectives of LISP single-hop DHT mapping overlay (LISP-SHDHT) are fast lookup and load balancing [263]. MBs are called SHDHT nodes and form a single-hop DHT. The system internally uses two dif-

ferent namespaces: node IDs and partition IDs. Node IDs designate MBs. Partition IDs correspond to the hashed EID space and designate end-hosts. Each MB has at least one partition ID assigned which indirectly defines the partition ID range the MB is responsible for. IDs in a partition range are by definition called resource IDs. Each MB knows all mappings between partition IDs and node IDs, i.e., each MB can resolve EID-to-MB mappings in one hop.

When a new mapping is registered in LISP-SHDHT, an ETR sends a map-register message to a known MB. The MB generates the resource ID from the to-be-registered EID using a hash function. The resource ID is matched to the closest MB using the MB's internal node routing table. If the current node is responsible for the resource ID, it stores the mapping. Otherwise, the it forwards the map-register message to the responsible MB which registers the mapping.

The lookup procedure in LISP-SHDHT works similarly. An ITR sends a map-request message to a known MB. The MB generates the resource ID from the requested EID using a hash function. The resource ID is matched to the closest MB using the MB's internal node routing table. If the current node is responsible for the resource ID, it replies the mapping. Otherwise, the current MB forwards the map-request message to the responsible MB which then replies the mapping.

MDHT The Multi-Level DHT (MDHT) has been proposed as a name resolution service for information-centric networks [264] and maps flat object identifiers to network addresses. It could be reused for EID-to-RLOC mappings, so we describe it here using the LISP nomenclature. In MDHT, all EIDs located in a specific access network are stored in a MB called access node. MBs are grouped in a nested, hierarchical structure of DHT areas, e.g., a POP DHT holds EID-to-MB pointers for all EIDs within the point of presence, and an AS DHT holds EID-to-MB pointers for all EIDs inside the AS.

To retrieve a mapping, an ITR queries its MB. If the MB holds the requested mapping, it can locally retrieve the mapping, otherwise it queries its peers in the DHT area. If the mapping is not retrievable in the current DHT area, the query is recursively forwarded to the next higher DHT area until it can be answered. The

answering DHT area returns an EID-to-MB mapping to the requesting MB. Finally, the requesting MB sends a map-request to the authoritative MB and returns the mapping to the ITR. Most likely, the presented mapping system works only within a single AS. A global DHT is unlikely to scale. To alleviate this, the paper suggests object identifier prefixes and a global resolution exchange system.

3.3.6.7 Map-Bases with Partial Knowledge using Multicast Overlay (MBPK-MCO)

The "EID mappings Multicast Across Cooperating Systems" for LISP [224] (EMACS-LISP) is another mapping mechanism that had been proposed for LISP. MBs join multicast groups for all EID prefixes they are responsible for. If that prefix is X.Y.A.B/16, the address of the corresponding multicast group is, e.g., 238.1.X.Y. In case of a cache miss for EID X.Y.A.B, the ITR sends the data packet to the corresponding multicast group so that all MBs of that group receive it. All MBs having appropriate mappings for the requested EID can respond with a map-reply. However, when data packets are relayed over this structure, only one of these MBs should deliver the packet to avoid duplicates at the destination. This approach has several drawbacks. Up to 2^{16} multicast groups need to be maintained in BGP and a large amount of unnecessary extra traffic is generated through multicast delivery.

3.4 Lessons Learned

The scalability issues of the Internet routing protocols could hamper future growth of the Internet. We realized that a new Internet routing architecture which improves routing scalability must also provide scalable support for traffic engineering, multihoming, and mobility. Renumbering of an entire AS should be simplified, and routing quality and security are very important. Last but not least, deployability of a solution is a prerequisite for its adoption in practice. Optimally, changes are incremental, backwards compatible and provide advantages for early

adopters. This way, an incentive driven deployment will eventually lead to a replacement of current architectures towards a new, scalable Internet architecture.

The Loc/ID split principle is a very promising solution for the routing in the future Internet. Many previous Loc/ID split based architectures still use the ID for routing in local routing domains. This is a straightforward solution but it violates the general Loc/ID principle of having a routing-independent ID. We developed our GLI-Split architecture that implements the Loc/ID split concept within today's IPv6 Internet. The outstanding feature of the GLI-Split architecture is that it really splits identification information from all routing information, while still being backwards compatible to plain IPv6 communication. It solves the scalability problem and provides many benefits to users in GLI-domains. They can change providers without internal renumbering, multihoming is facilitated even for smallest GLI-domains and can be exploited for multipath forwarding, traffic engineering, improved reliability and mobility support. GLI-Split is incrementally deployable on a per-domain basis and also within a single domain the migration from non-upgraded GLI-nodes to upgraded GLI-nodes can be done gradually. We implemented a proof-of-concept simulation of GLI-Split in OMNeT++ and demonstrated that our architecture is working as expected and also showed its incremental deployability. The simulation showed that communication with a GLI-host has similar round-trip times as communication flows using plain IPv6. Only for the first packet in a communication session, the GLI-gateways at the sending and the receiving side perform a mapping lookup that both add 12ms to the total round-trip time. Although the full set of benefits is available only for communications among GLI-nodes with upgraded networking stacks, GLI-Split provides advantages for upgraded GLI-nodes when communicating with the plain IPv6 Internet and even for plain IPv6 nodes in GLI-domains.

Besides the GLI-Split, several other routing architectures implementing the Loc/ID split have been proposed for the Internet. In many of them, an intermediate node queries a mapping system for ID-to-Loc mappings. We classified all proposed mapping system approaches according to their internal structure and condensed our resulting insights into the development of the Future Internet

Routing Mapping System (FIRMS). It includes security and resilience features and can relay packets when intermediate nodes encounter cache misses for required ID-to-Loc mappings. Our performance assessment showed that storage requirements, update loads, and resolution delays for FIRMS are manageable. We verified the design with an OMNet++ simulation of the mapping system that can be used for systems research. We also implemented a proof-of-concept for FIRMS in the G-Lab experimental facility and demonstrated its operation [38]. Coincidental real failures of hardware in the testbed during the demonstration showed that the FIRMS architecture is resilient against outages and still works in unforeseen situations.

During the development and design of our new naming, addressing, and routing scheme, we learned that ideas must be re-evaluated over and over again, and must be put in new contexts. Many of the ideas that went into our design have been proposed in one or another form before. We combined and completed them, and created a new architecture with some outstanding benefits. Nevertheless, the main value is not the proposed architecture itself, but the discussion of its features within the research community and standardization bodies. We hope that some of our ideas find their way into new standards that provide a better Internet.

4 Conclusion

The only true wisdom is in knowing you know nothing.
(Socrates)

The Internet is constantly growing and evolving. Its modular architecture facilitates that new mechanisms and protocols can be added to incorporate and enable new services and technologies. At the center of the Internet's protocol stack stands the Internet Protocol (IP) as a common denominator that enables all communication. Routing of IP packets is the central task of all networks attached to the Internet. To make routing efficient, resilient, and scalable, several aspects must be considered. Care must be taken that traffic is well balanced to make efficient use of the existing network resources, both in failure free operation and in failure scenarios. Another important factor which must be considered is the size of the forwarding tables inside the routers. The larger these tables get, the longer it takes to decide where to forward a packet to. The high growth rate of current routing table sizes in the Internet could impair the scalability of the Internet in the near future. In this monograph we studied the optimization of intradomain routing, focusing mainly on link loads, resilience, and add-on protocols, and we developed a new protocol to limit future growth of routing table sizes.

Finding the optimal routing in a network is an NP-complete problem. Therefore, routing optimization is usually performed using heuristics. In Section 2.1, we proposed speedup techniques for our optimization heuristic, new objective functions, and combined optimization of a primary and secondary objective function. The effectiveness of the proposed speedup techniques depends on the ap-

plied objective function. The considered functions can be used to improve the network in different ways. While some of them lower the maximum link utilization, others also try to minimize the path lengths. We showed that a routing optimized with one objective function is often not good when looking at another objective function. It can even be worse than unoptimized routing with respect to that objective function. Therefore, a traffic engineer must carefully decide which optimization goal is most important. Our combined optimization approach can improve several objectives at the same time. It is applicable for optimization of both resilient and non-resilient IP routing and very effective if a few severe network-inherent bottlenecks prohibit an effective improvement of the routing.

In Section 2.2 we extended our optimization and analysis to include the loop-free alternate (LFA) IP fast reroute (IP-FRR) mechanism. It is simple and fast, and the only IP-FRR mechanism that is already standardized. However, LFAs usually cannot protect all traffic in a network even against single link failures and some LFAs may create extra-loops in case of node and multiple failures. LFAs may be applied for different applications: to reduce lost traffic between the detection of a failure and the completion of IP rerouting, to improve the availability for some traffic aggregates, or to protect all traffic on a link to delay IP routing if that link fails. We looked at LFA coverage in several test networks from the point of view of these applications. Therefore, metrics of interests are traffic loss due to missing LFAs, percentage of fully protected traffic, and percentage of fully protected links. Moreover, we differentiated between general LFAs and those that avoid extra-loops under any condition. We showed that administrative IP link costs can be optimized so that LFA coverage can be significantly increased. The achievable LFA coverage highly depends on the network structure. In a few networks all traffic can be protected by LFAs after routing optimization, but only if extra-loops are acceptable in case of unlikely failures. When allowing only LFAs that avoid extra-loops for protection, LFA coverage is reduced, and 100% LFA coverage cannot be achieved in any network. In such a case, the choice of the right objective function for routing optimization has a large influence on the resulting LFA coverage. Link costs that maximize the percentage of protected

destinations often produce bad results in the light of the metrics that are motivated by the considered applications. As some traffic aggregates may be more important than others with regard to fast protection, we developed a method that preferentially protects such traffic and demonstrated its usefulness. We observed that optimizing link costs to improve only LFA coverage can lead to a huge imbalance of traffic in the network so that traffic may be lost due to overload. This is counterproductive as minimizing traffic loss is a major motivation for the use of LFAs. To solve that problem, we proposed Pareto-optimization yielding a set of link costs that are Pareto-optimal with regard to traffic loss due to missing LFAs and maximum relative link load. Some link costs among them perform well with regard to both metrics.

In Section 2.3 we analyzed and optimized the pre-congestion notification (PCN) mechanism. It marks packets when PCN traffic exceeds configured admissible or supportable rate thresholds (*AR*, *SR*) on a link of the PCN domain. This feedback allows simple and scalable admission control (AC) and flow termination (FT) in IP networks. Two different options with different benefits and drawbacks are proposed, which need to be understood. One class of methods requires two different marking mechanisms (dual marking PCN architecture, DM-PCN) and its *AR*- and *SR*-thresholds can be chosen independently of each other. Another class requires only a single marking mechanism (single marking PCN architecture, SM-PCN) and its *SR*-thresholds must be a fixed multiple of the *AR*-thresholds for all links in the PCN domain. We configured the link-specific *AR*- and *SR*-thresholds for a PCN domain and optimized its routing so that the admissible protected high-priority traffic is maximized. We showed that this is complex for SM-PCN, due to the fixed backup factor, whose impact was illustrated in detail for an example network. Our results for a large set of random networks showed that DM-PCN can support 50% more protected traffic than SM-PCN when unoptimized routing with uniform link costs is used. Routing optimization improves the throughput for both SM- and DM-PCN tremendously. With routing optimization, DM-PCN can support even 100% more protected traffic than SM-PCN, at least in large networks. These results reveal that SM-PCN uses network

resources less efficiently for resilient AC than DM-PCN and also show that the difference is significant. The algorithms presented in this work can be used to configure PCN rate thresholds and to optimize IP routing for PCN networks in practice.

In Chapter 2, we analyzed and optimized different routing protocols and add-on protocols to optimize the load distribution and failure tolerance in a network. These issues can and must be managed inside each autonomous system. In contrast, there is a problem that can only be resolved on a global scale. Chapter 3 explains that the scalability of the Internet is at risk since a major and intensifying growth of the interdomain routing tables has been observed. We analyzed several protocols and architectures that can be used to make interdomain routing more scalable. The most promising approach is the locator/identifier (Loc/ID) split architecture which separates routing from host identification. This way, changes in connectivity, mobility of end hosts, or traffic-engineering activities are hidden from the routing in the core of the Internet and the routing tables can be kept much smaller. The Loc/ID split concept also imposes changes to the Internet routing architecture, but often only to a very limited number of border gateways. It is a promising candidate to effectively cope with the scalability problem.

All of the currently proposed Loc/ID split approaches have their downsides. In particular, we considered the fact that most architectures use the ID for routing outside the Internet's core as a poor design, which inhibits many of the possible features of a new routing architecture. To better understand the problems and to provide a solution for a scalable routing design that implements a true Loc/ID split, we developed the GLI-Split protocol in Section 3.2, which provides separation of global and local routing and uses an ID that is independent from any routing decisions. GLI-Split implements the Loc/ID split concept within today's IPv6 Internet. Thereby, it can solve the scalability problem for a future IPv6 Internet when prefixes of global GLI-addresses are adopted for core routing. In addition, it provides many benefits to users in GLI-domains. They can change providers without internal renumbering, multihoming is facilitated even for smallest GLI-domains and can be exploited for multipath forwarding, traffic engineering, im-

proved reliability and mobility support. GLI-Split is incrementally deployable on a per-domain basis and the migration within a single domain from non-upgraded to upgraded GLI-nodes can be done gradually. GLI-gateways perform simple address rewriting without the need for session state. This also holds for interworking with the plain IPv6 Internet. In contrast to many other proposals, GLI-Split does not need triangle routing via extra devices for that purpose. We demonstrated the functionality of GLI-Split by simulating its operation in OMNeT++ and showed that short additional delays occur only for first packets of a communication session. This is a small cost compared to the benefits the GLI-split provides.

Besides GLI-Split, several other new routing architectures implementing Loc/ID split have been proposed for the Internet. Most of them assume that a mapping system is queried for EID-to-RLOC mappings by an intermediate node at the border of an edge network. When the mapping system is queried by an intermediate node, packets are already on their way towards their destination, and therefore, the mapping system must be fast, scalable, secure, resilient, and should be able to relay packets without locators to nodes that can forward them to the correct destination. We developed a classification for all proposed mapping system architectures and showed their similarities and differences. To condense our knowledge of mapping systems and to gain a better understanding for occurring problems when designing a mapping system, we developed the fast two-level mapping system FIRMS, which is presented in Section 3.3. It includes security and resilience features as well as a relay service for initial packets of a flow when intermediate nodes encounter a cache miss for the EID-to-RLOC mapping. We simulated FIRMS using OMNeT++ to demonstrate its functionality. We also implemented a proof-of-concept of FIRMS in the G-Lab experimental facility and showed its operation. Our performance assessment predicts that FIRMS scales significantly better than centralized mapping systems with respect to storage requirements and update rates. We proposed four categories of mapping systems and used them to provide a comprehensive review. FIRMS has structures in common with many other mapping system, but clearly differs in its overall design and has many benefits that could promote its deployment.

In the course of this monograph, we developed mechanisms that make routing in the Internet more efficient, more reliable, and more scalable. In particular, we improved the understanding of the aspects that must be regarded when routing is optimized inside a network, and developed new protocols that allow future growth of the Internet without risking its scalability due to unrestrained routing- and forwarding-table growth. In this respect, the work presented in this monograph improves the understanding of limiting factors of routing protocols in general and it represents an important step towards a stable and reliable future Internet.

Nomenclature and Acronyms

OPTIMIZATION OF IP-BASED ROUTING PROTOCOLS

$\mathcal{E}$	The set of directed links (edges) of a network topology
$\mathcal{V}$	The set of nodes (vertices) of a network topology
$\mathcal{V}_A$	The set of nodes removed from some topologies to make them two-connected for resilience analysis
$\mathbf{c}$	Bandwidth (capacity) $\mathbf{c}(l)$ of all links $l \in \mathcal{E}$
$\mathbf{k}$	Administrative cost $\mathbf{k}(l)$ of all links $l \in \mathcal{E}$, which determine the shortest path routing
$\mathcal{D}$	The traffic matrix consists of traffic aggregates (demands) $d_{v,w} \in \mathcal{D}$ between a source node v and a destination node $w \in \mathcal{V}$; its rate is given by $r(d_{v,w})$
$\mathcal{S}$	The set of considered failure scenarios comprised by sets of failed network elements $s \subseteq (\mathcal{V} \cup \mathcal{E})$; the failure-free scenario is denoted by $s = \emptyset$; usually, we consider the failure free scenario and all single link failures $\mathcal{S} = \{\emptyset\} \cup \{\{l\} : l \in \mathcal{E}\}$ or failures of both directions of a link $\mathcal{S} = \{\emptyset\} \cup \{\{l_{u,v}, l_{v,u}\} : l \in \mathcal{E}\}$
$u_s^{\mathbf{k}}(l, v, w)$	Function that indicates the fraction of traffic from v to w that is carried over link l in failure scenario s when link costs $\mathbf{k}$ apply

$\rho(\mathbf{k}, l, s)$	The utilization of a link l in a failure scenario s for given link costs $\mathbf{k}$
$\rho_{\mathcal{E}}^{\max}(\mathbf{k}, s)$	The utilization of the highest loaded link in given failure scenarios $s \in \mathcal{S}$ for given link costs $\mathbf{k}$
$\rho_{\mathcal{S}}^{\max}(\mathbf{k}, l)$	The maximum utilization of link l in all failure scenarios $s \in \mathcal{S}$ for given link costs $\mathbf{k}$
$\rho_{\mathcal{S},\mathcal{E}}^{\max}(\mathbf{k})$	The maximum utilization of all links $l \in \mathcal{E}$ in all failure scenarios $s \in \mathcal{S}$ for given link costs $\mathbf{k}$
$\rho_{\emptyset}^{\max}$	Maximum link utilization; the utilization of the most loaded link in the failure free topology
$\rho_{\mathcal{S}}^{\max}$ or $\rho^{\max}$	The maximum utilization of the highest loaded link in the worst failure scenario for given link costs $\mathbf{k}$
$\Phi_{\emptyset}$	General Fortz function applied to failure free routing; reflects load level of all links
$\Phi_{\mathcal{S}}^{\text{avg}}$	Equal-weighted average of Fortz-values over all failure scenarios
$\Phi_{\mathcal{S}}^{\text{weighted}}$	Weighted average of Fortz-values over all failure scenarios, given half of the weight to the failure free scenario
$\Phi_{\mathcal{S}}^{\text{max,in}}$	Fortz-value of artificial scenario where links have their maximum utilization over all considered failures
$\Phi_{\mathcal{S}}^{\text{max,out}}$	Fortz-value $\Phi(s)$ of the worst failure scenario s
π^{loss}	Percentage of lost traffic due to missing LFAs
π^{dest}	Percentage of protected destinations
π^{full}	Percentage of fully protected traffic
π^{link}	Percentage of fully protected links
$\Phi(s)$	General Fortz function for a failure scenario $s \in S$
Ψ	Placeholder for any objective function
$\Psi_1 + \Psi_2$	Objective function Ψ_2 applied as secondary objective during optimization for primary objective Ψ_1

$\mathbf{k}^{\Psi}$	Best obtained link cost vector after link cost optimization for objective function Ψ
$\mathbf{k}_{\text{u}}$	Uniform link costs, e.g., $\mathbf{k}_{\text{u}} = \mathbf{1}$
$\mathbf{k}^{\text{dest}}$	Link costs optimized for π^{dest}
$\mathbf{k}^{\text{full}}$	Link costs optimized for π^{full}
$\mathbf{k}^{\text{link}}$	Link costs optimized for π^{link}
$\mathbf{k}^{\text{loss}}$	Link costs optimized for π^{loss}
$\mathbf{k}^{\rho}$	Link costs optimized for ρ^{max}
$\mathbf{k}^{\text{Par}}$	Pareto-optimal link costs
LAC	Loop avoidance class
LP-LAC	LAC that uses all (general) LFAs
ND-LAC	LAC that uses only LFAs which avoid extra-loops in case of node failures and multiple failures
NP-LAC	LAC that uses only LFAs which avoid extra-loops in case of node failures
AC	Admission control
FT	Flow termination
$AR(l)$	Admissible rate threshold for PCN
$SR(l)$	Supportable rate threshold for PCN
$r(l)$	PCN traffic rate on a link
SM-PCN	Single marking PCN architecture using only one bit in the IP header
DM-PCN	Dual marking PCN architecture using two bits in the IP header
NP	No pre-congestion; new flows can be admitted
AS	Admission stop; no new flows should be admitted
ASR	Rate of AS-marked traffic
ET	Excess traffic; traffic rate above SR threshold in DM-PCN architecture

ETR	Rate of ET-marked traffic
b	Backup factor in SM-PCN that controls the relation between primary and backup capacity: $SR(l) = b \cdot AR(l)$
$\sigma(\mathbf{k})$	Scaling factor which expresses the multiple of the traffic matrix that can be admitted as protected priority traffic

DESIGN OF A NEW ADDRESSING AND ROUTING PROTOCOL

IRTF	Internet research task force
RRG	Routing research group; group inside the IRTF
FIB	Forwarding information base of a router
AS	Autonomous system; independent routing domain
DFZ	Default-free zone; area where routers must know a next-hop for each destination
PI addresses	Provider-independent IP addresses
PA addresses	Provider-aggregatable IP addresses
Loc/ID split	Locator / identifier split
ITR	Ingress tunnel router
ETR	Egress tunnel router
EID	Endpoint identifier
RLOC	Routing locator
GLI-Split	Global locator, local locator and identifier split
GL	Global locator
LL	Local locator
ID	Identifier
GAP	Global address preservation
FIRMS	Future Internet routing mapping system
EID	Endpoint identifier

RLOC	Routing locator
MB	Map-base
MBP	Map-base pointer
MR	Map-resolver
MBPX	Map-base pointer exchange node
RIR	Regional Internet registry
LIR	Local Internet registry

MBFK	Map-base with full knowledge
MBPK	Map-base with partial knowledge
MBPK-LL	MBPK using local lookup
MBPK-SRL	MBPK using single remote lookup
MBPK-IRL	MBPK using iterative remote lookup
MBPK-OL	MBPK using overlay lookup
MBPK-HSO	MBPK using hierarchically structured overlay
MBPK-DHT	MBPK using distributed hash tables
MBPK-MCO	MBPK using multicast overlay

Bibliography of the Author

— Journal Papers —

[1] M. Hartmann, D. Hock, and M. Menth, "Routing Optimization for IP Networks Using Loop-Free Alternates," *under submission*, 2014.

[2] M. Menth and M. Hartmann, "Threshold Configuration and Routing Optimization for PCN-Based Resilient Admission Control," *Computer Networks*, vol. 53, no. 11, pp. 1771 – 1783, Jul. 2009.

[3] M. Menth, R. Martin, M. Hartmann, and U. Spörlein, "Efficiency of Routing and Resilience Mechanisms in Packet-Switched Communication Networks," *European Transactions on Telecommunications (ETT)*, vol. 21, no. 2, pp. 108 – 120, Mar. 2010.

[4] M. Menth, M. Hartmann, R. Martin, T. Čičić, and A. Kvalbein, "Loop-Free Alternates and Not-Via Addresses: A Proper Combination for IP Fast Reroute?" *Computer Networks*, vol. 54, no. 8, pp. 1300 – 1315, Jun. 2010.

[5] T. Čičić, A. F. Hansen, A. Kvalbein, M. Hartmann, R. Martin, M. Menth, S. Gjessing, and O. Lysne, "Relaxed Multiple Routing Configurations: IP Fast Reroute for Single and Correlated Failures," *IEEE Transactions on Network and Service Management (IEEE TNSM)*, vol. 6, no. 1, pp. 1 – 14, Mar. 2009.

[6] C. Żukowski, A. Tomaszewski, M. Pióro, D. Hock, M. Hartmann, and M. Menth, "Compact Node-Link Formulations for the Optimal Single Path MPLS Fast Reroute Layout," *Advances in Electronics and Telecommunications*, vol. 2, no. 3, pp. 55 – 60, Sep. 2011.

[7] D. Hock, M. Hartmann, M. Menth, M. Pióro, A. Tomaszewski, and C. Żukowski, "Comparison of IP-Based and Explicit Paths for One-to-One Fast Reroute in MPLS Networks," *Telecommunication Systems (TS) Journal*, vol. 52, no. 2, pp. 947 – 958, Feb. 2013.

[8] D. Hock, M. Hartmann, C. Schwartz, and M. Menth, "ResiLyzer: Ein Werkzeug zur Analyse der Ausfallsicherheit in paketvermittelten Kommunikationsnetzen," *PIK - Praxis der Informationsverarbeitung und Kommunikation*, vol. 34, no. 3, pp. 158 – 159, Aug. 2011.

[9] M. Menth, M. Hartmann, D. Klein, and P. Tran-Gia, "Future Internet Routing: Motivation and Design Issues," *it - Information Technology*, vol. 5, no. 6, pp. 358 – 366, Dec. 2008.

[10] M. Menth, M. Hartmann, and D. Klein, "Global Locator, Local Locator, and Identifier Split (GLI-Split)," *Future Internet*, vol. 5, no. 1, pp. 67 – 94, Mar. 2013.

[11] M. Menth, M. Hartmann, and M. Hoefling, "FIRMS: A Mapping System for Future Internet Routing," *IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Internet Routing Scalability*, vol. 28, no. 8, pp. 1326 – 1331, Oct. 2010.

[12] M. Hoefling, M. Menth, and M. Hartmann, "A Survey of Mapping Systems for Locator/Identifier Split Internet Routing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1842 – 1858, Nov. 2013.

[13] P. Tran-Gia, T. Hoßfeld, M. Hartmann, and M. Hirth, "Crowdsourcing and its Impact on Future Internet Usage," *it - Information Technology*, vol. 55, no. 4, pp. 139 – 145, Jul. 2013.

[14] D. Klein, P. Tran-Gia, and M. Hartmann, “Aktuelles Schlagwort: Big Data,” *Informatik-Spektrum*, vol. 36, no. 3, pp. 319 – 323, Jun. 2013.

— Conference Papers —

[15] M. Hartmann, D. Hock, M. Menth, and C. Schwartz, “Objective Functions for Optimization of Resilient and Non-Resilient IP Routing,” in *7th International Workshop on Design of Reliable Communication Networks (DRCN)*, Washington, D.C., USA, Oct. 2009.

[16] M. Menth, M. Hartmann, and R. Martin, “Robust IP Link Costs for Multilayer Resilience,” in *6th IFIP-TC6 International Networking Conference (NETWORKING)*, Atlanta, GA, USA, May 2007.

[17] M. Duelli, M. Hartmann, M. Menth, R. Hülsermann, and M. Düser, “Performance Evaluation of IP over Cost-Optimized Optical Multilayer Networks with SRLGs,” in *ITG Workshop on Photonic Networks*, Leipzig, Germany, Apr. 2008.

[18] S. Gebert, D. Hock, M. Hartmann, J. Spoerhase, T. Zinner, and P. Tran-Gia, “Including Energy Efficiency Aspects in Multi-Layer Optical Network Design,” in *5th International Conference on Communications and Electronics (ICCE 2014)*, Da Nang, Vietnam, Jul. 2014.

[19] T. Čičić, A. F. Hansen, A. Kvalbein, M. Hartmann, R. Martin, and M. Menth, “Relaxed Multiple Routing Configurations for IP Fast Reroute,” in *11th IEEE Network Operations and Management Symposium (NOMS)*, Salvador de Bahia, Brazil, Apr. 2008.

[20] D. Hock, M. Hartmann, M. Menth, and C. Schwartz, “Optimizing Unique Shortest Paths for Resilient Routing and Fast Reroute in IP-Based Networks,” in *12th IEEE Network Operations and Management Symposium (NOMS)*, Osaka, Japan, Apr. 2010.

[21] D. Hock, M. Hartmann, T. Neubert, and M. Menth, "Loop-Free Convergence using Ordered FIB Updates: Analysis and Routing Optimization," in *8th International Workshop on Design of Reliable Communication Networks (DRCN)*, Krakow, Poland, Oct. 2011.

[22] M. Pióro, A. Tomaszewski, C. Żukowski, D. Hock, M. Hartmann, and M. Menth, "Optimized IP-Based vs. Explicit Paths for One-to-One Backup in MPLS Fast Reroute," in *14th International Telecommunications Network Strategy and Planning Symposium (NETWORKS), Best Paper Award*, Warsaw, Poland, Sep. 2010.

[23] C. Żukowski, A. Tomaszewski, M. Pióro, D. Hock, M. Hartmann, and M. Menth, "Compact Node-Link Formulations for the Optimal Single Path MPLS Fast Reroute Layout," in *1st European Teletraffic Seminar (ETS)*, Poznan, Poland, Feb. 2011.

[24] D. Hock, M. Hartmann, C. Schwartz, and M. Menth, "Effectiveness of Link Cost Optimization for IP Rerouting and IP Fast Reroute ," in *15th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB) and Dependability and Fault Tolerance (DFT)*, Essen, Germany, Mar. 2010.

[25] D. Hock, M. Menth, M. Hartmann, C. Schwartz, and D. Stezenbach, "ResiLyzer: A Tool for Resilience Analysis in Packet-Switched Communication Networks," in *15th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB) and Dependability and Fault Tolerance (DFT)*, Essen, Germany, Mar. 2010.

[26] D. Klein, M. Hartmann, and M. Menth, "NAT Traversal for LISP Mobile Node," in *ACM CoNEXT workshop Re-Architecting the Internet (ReArch)*, Philadelphia, PA, USA, Nov. 2010.

[27] M. Menth, D. Klein, and M. Hartmann, "Improvements to LISP Mobile Node," in *22nd International Teletraffic Congress (ITC)*, Amsterdam, The

Netherlands, Sep. 2010.

[28] D. Klein, M. Höfling, M. Hartmann, and M. Menth, "Integration of LISP and LISP-MN into INET," in *5th International Workshop on OMNeT++*, Desenzano, Italy, Mar. 2012.

[29] D. Stezenbach, M. Hartmann, and K. Tutschku, "Parameters and Challenges for Virtual Network Embedding in the Future Internet," in *First IEEE Workshop on Algorithms and Operating Procedures for Federated Virtualized Networks (FEDNET) - Collocated with IEEE NOMS*, Maui, Hawaii, USA, Jun. 2012.

[30] J. Inführ, D. Stezenbach, M. Hartmann, K. Tutschku, and G. R. Raidl, "Using Optimized Virtual Network Embedding for Network Dimensioning," in *IEEE Conference on Networked Systems (NetSys)*, Stuttgart, Germany, Mar. 2013.

[31] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, "Pareto-Optimal Resilient Controller Placement in SDN-based Core Networks," in *22nd International Teletraffic Congress (ITC)*, Shanghai, China, Sep. 2013.

— Others —

[32] M. Menth, R. Martin, M. Hartmann, and U. Spörlein, "Efficiency of Routing and Resilience Mechanisms in Packet-Switched Networks," University of Würzburg, Institute of Computer Science, Technical Report, No. 425, May 2007.

[33] M. Menth, M. Hartmann, and R. Martin, "Robust IP Link Costs for Multilayer Resilience," University of Würzburg, Institute of Computer Science, Technical Report, No. 427, May 2007.

[34] R. Martin, M. Menth, M. Hartmann, T. Čičić, and A. Kvalbein, "The Effect of Combining Loop-Free Alternates and Not-Via Addresses in IP Fast

Reroute," University of Würzburg, Institute of Computer Science, Technical Report, No. 432, Sep. 2007.

[35] M. Menth, M. Hartmann, and D. Klein, "Global Locator, Local Locator, and Identifier Split (GLI-Split)," University of Würzburg, Institute of Computer Science, Technical Report, No. 470, Apr. 2010.

[36] M. Menth, M. Hartmann, and D. Klein, "Demonstration of Global Locator, Local Locator, and Identifier Split (GLI-Split)," in *Würzburg Workshop on IP: Visions of Future Generation Networks (EuroView)*, Würzburg, Germany, Jul. 2009.

[37] M. Menth, M. Hartmann, and M. Hoefling, "Mapping Systems for Loc/ID Split Internet Routing," University of Würzburg, Institute of Computer Science, Technical Report, No. 472, May 2010.

[38] M. Hartmann, D. Hock, M. Hoefling, T. Neubert, and M. Menth, "Demonstration of the Future InteRnet Mapping System (FIRMS) in the G-Lab Experimental Facility," in *Würzburg Workshop on IP: Visions of Future Generation Networks (EuroView)*, Würzburg, Germany, Aug. 2010.

[39] M. Menth, M. Hartmann, and M. Hoefling, "Demo: a Future InteRnet Mapping System (FIRMS)," in *Würzburg Workshop on IP: Visions of Future Generation Networks (EuroView)*, Würzburg, Germany, Jul. 2009.

[40] D. Klein, M. Hartmann, M. Höfling, and M. Menth, "Demo: Improvements to LISP Mobile Node Including NAT Traversal," in *Würzburg Workshop on IP: Visions of Future Generation Networks (EuroView)*, Würzburg, Germany, Aug. 2010.

[41] D. Hock, S. Gebert, M. Hartmann, T. Zinner, and P. Tran-Gia, "Demonstration of POCO: A Framework for the Pareto-Optimal Resilient Controller Placement in SDN-based Core Networks," in *14^{th} IEEE Network Operations and Management Symposium (NOMS)*, Krakow, Poland, May 2014.

[42] D. Hock, M. Hartmann, S. Gebert, T. Zinner, and P. Tran-Gia, "Demonstration of POCO-PLC: Enabling Dynamic Pareto-Optimal Resilient Controller Placement in SDN Networks," in *33rd IEEE International Conference on Computer Communications (INFOCOM)*, Toronto, Canada, Apr. 2014.

General References

[43] M. Menth, "Design and Performance Evaluation of Control Mechanisms for the Future Internet," Professorial thesis, University of Würzburg, Faculty of Computer Science, 2011.

[44] M. Pióro, Á. Szentesi, J. Harmatos, A. Jüttner, P. Gajowniczek, and S. Kozdrowski, "On Open Shortest Path First Related Network Optimisation Problems," *Performance Evaluation*, vol. 48, no. 1 – 4, May 2002.

[45] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage, "California Fault Lines: Understanding the Causes and Impact of Network Failures," in *ACM SIGCOMM*, New Delhi, India, Aug. 2010.

[46] B. Fortz and M. Thorup, "Robust Optimization of OSPF/IS-IS Weights," in *International Network Optimization Conference (INOC)*, Paris, France, Oct. 2003.

[47] A. Basu and J. G. Riecke, "Stability Issues in OSPF Routing," in *ACM SIGCOMM*, San Diego, CA, USA, Aug. 2001.

[48] M. Shand and S. Bryant, "RFC5714: IP Fast Reroute Framework," http://www.rfc-editor.org/rfc/rfc5714.txt, Jan. 2010.

[49] A. Atlas and A. Zinin, "RFC5286: Basic Specification for IP Fast Reroute: Loop-Free Alternates," http://www.rfc-editor.org/rfc/rfc5286.txt, Sep. 2008.

[50] P. Francois and O. Bonaventure, "An Evaluation of IP-Based Fast Reroute Techniques," in *1st ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, Toulouse, France, Oct. 2005.

[51] A. F. Hansen, T. Cicic, and S. Gjessing, "Alternative Schemes for Proactive IP Recovery," in *2nd Conference on Next Generation Internet Design and Engineering (NGI)*, Valencia, Spain, Apr. 2006.

[52] M. Gjoka, V. Ram, and X. Yang, "Evaluation of IP Fast Reroute Proposals," in *2nd IEEE International Conference on Communication System Software and Middleware (COMSWARE)*, Bangalore, India, Jan. 2007.

[53] S. Shenker, "Fundamental Design Issues for the Future Internet," *IEEE Journal on Selected Areas in Communications (J-SAC)*, vol. 13, no. 7, Sep. 1995.

[54] D. M. Johnson, "QoS Control versus Generous Dimensioning," *British Telecom Technology Journal*, vol. 23, no. 2, Apr. 2005.

[55] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An Approach to Alleviate Link Overload as Observed on an IP Backbone," in *22nd IEEE International Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, April 2003.

[56] M. Menth, "Efficient Admission Control and Routing in Resilient Communication Networks," PhD thesis, University of Würzburg, Faculty of Computer Science, July 2004.

[57] IETF Working Group, "Congestion and Pre-Congestion Notification (pcn)," http://datatracker.ietf.org/doc/charter-ietf-pcn/, Feb. 2007.

[58] P. Eardley (Ed.), "RFC5559: Pre-Congestion Notification (PCN) Architecture," http://www.rfc-editor.org/rfc/rfc5559.txt, Jun. 2009.

[59] B. Fortz and M. Thorup, “Internet Traffic Engineering by Optimizing OSPF Weights,” in *19th IEEE International Conference on Computer Communications (INFOCOM)*, Tel-Aviv, Israel, Mar. 2000.

[60] S. Balon, F. Skivée, and G. Leduc, “How Well Do Traffic Engineering Objective Functions Meet TE Requirements?” in *5th IFIP-TC6 International Networking Conference (NETWORKING)*, Coimbra, Portugal, May 2006.

[61] B. Fortz and M. Thorup, “Increasing Internet Capacity Using Local Search,” *Computational Optimization and Applications*, vol. 29, no. 1, Oct. 2004.

[62] S. Köhler, D. Staehle, and U. Kohlhaas, “Optimization of IP Routing by Link Cost Specification,” in *15thITC Specialist Seminar*, Würzburg, Jun. 2002.

[63] A. Sridharany, R. Guérin, and C. Diot, “Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF/IS-IS Networks,” in *22nd IEEE International Conference on Computer Communications (INFOCOM)*, San Francisco, CA, USA, Apr. 2003.

[64] C. Lopes and A. de Sousa, “Heuristics for the MPLS Network Design with Single Path Minimum Weight Routing,” in *3rd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NET)*, West Yorkshire, UK, Jul. 2005.

[65] Y. Wang, Z. Wang, and L. Zhang, “Internet Traffic Engineering without Full Mesh Overlaying,” in *20th IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, Apr. 2001.

[66] K. G. Ramakrishnan and M. A. Rodrigues, “Optimal Routing in Shortest-Path Data Networks,” *AT&T Bell Laboratories Technical Journal*, Spring 2001.

[67] M. P. Petterson, R. Szymankek, and K. Kuchcinski, "A CP-LP Hybrid Method for Unique Shortest Path Routing Optimization," in *International Network Optimization Conference (INOC)*, Spa, Belgium, Jul. 2007.

[68] A. Tomaszewski, M. Pioro, M. Dzida, and M. Zagozdzon, "Optimization of Administrative Weights in IP Networks Using the Branch-and-Cut Approach," in *International Network Optimization Conference (INOC)*, Lisbon, Portugal, Mar. 2005.

[69] A. Tomaszwski, M. Pióro, M. Dzida, M. Mycek, and M. Zagożdżon, "Valid Inequalities for a Shortest-Path Routing Optimization Problem," in *International Network Optimization Conference (INOC)*, Spa, Belgium, Apr. 2007.

[70] A. Bley, "An Integer Programming Algorithm for Routing Optimization in IP Networks," in *16th Annual European Symposium on Algorithms (ESA)*, Karlsruhe, Germany, Sep. 2008.

[71] M. Ericsson, M. Resende, and P. Pardalos, "A Genetic Algorithm for the Weight Setting Problem in OSPF Routing," *Journal of Combinatorial Optimization*, vol. 6, no. 3, Sep. 2002.

[72] E. Mulyana and U. Killat, "An Alternative Genetic Algorithm to Optimize OSPF Weights," in *15thITC Specialist Seminar*, Würzburg, Germany, Jul. 2002.

[73] E. Mulyana and U. Killat, "A Hybrid Genetic Algorithm Approach for OSPF Weight Setting Problem," in *2ndPolish-German Teletraffic Symposium (PGTS)*, Gdansk, Poland, Sep. 2002.

[74] A. Riedl, "A Hybrid Genetic Algorithm for Routing Optimization in IP Networks Utilizing Bandwidth and Delay Metrics," in *IEEE Workshop on IP Operations and Management (IPOM)*, Dallas, TX, USA, Oct. 2002.

[75] A. Riedl, “Optimized Routing Adaptation in IP Networks Utilizing OSPF and MPLS,” in *IEEE International Conference on Communications (ICC)*, Anchorage, AK, USA, May 2003.

[76] C. Reichert and T. Magedanz, “A Fast Heuristic for Genetic Algorithms in Link Weight Optimization,” in *5th International Workshop on Quality of future Internet Services (QofIS)*, Barcelona, Spain, Sep. 2004.

[77] N. Wang, K.-H. Ho, and G. Pavlou, “Adaptive Multi-Topology IGP Based Traffic Engineering with Near-Optimal Network Performance,” in *7th IFIP-TC6 International Networking Conference (NETWORKING)*, Singapore, May 2008.

[78] J. Harmatos, “A Heuristic Algorithm for Solving the Static Weight Optimisation Problem in OSPF,” in *IEEE Global Communications Conference (GLOBECOM)*, San Antonio, TX, USA, Nov. 2001.

[79] S. Balon and G. Leduc, “Combined Intra- and Inter-domain Traffic Engineering using Hot-Potato Aware Link Weights Optimization,” in *ACM SIGMETRICS (short paper)*, Annapolis, MD, USA, Jun. 2008.

[80] A. Riedl and D. A. Schupke, “Routing Optimization in IP Networks Utilizing Additive and Concave Link Metrics,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, Oct. 2007.

[81] D. Xu, M. Chiang, and J. Rexford, “Link-State Routing with Hop-by-Hop Forwarding can Achieve Optimal Traffic Engineering,” in *27th IEEE International Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, USA, Apr. 2008.

[82] D. Yuan, “A Bi-Criteria Optimization Approach for Robust OSPF Routing,” in *3rd IEEE Workshop on IP Operations and Management (IPOM)*, Kansas City, MO, USA, Oct. 2003.

[83] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot, "IGP Link Weight Assignment for Transient Link Failures," in *18th International Teletraffic Congress (ITC)*, Berlin, Germany, Sep. 2003.

[84] A. Nucci, S. Bhattacharyya, N. Taft, and C. Diot, "IGP Link Weight Assignment for Operational Tier-1 Backbones," *IEEE/ACM Transactions on Networking*, vol. 15, no. 4, Aug. 2007.

[85] A. Sridharan and R. Guerin, "Making IGP Routing Robust to Link Failures," in *4th IFIP-TC6 International Networking Conference (NETWORKING)*, Waterloo, Canada, May 2005.

[86] M. Dzida, M. Zagożdżon, and M. Pióro, "Optimization of Resilient IP Networks with Shortest Path Routing," in *6th International Workshop on the Design of Reliable Communication Networks (DRCN)*, La Rochelle, France, Oct. 2007.

[87] B. Fortz, J. Rexford, and M. Thorup, "Traffic Engineering with Traditional IP Routing Protocols," *IEEE Communications Magazine*, vol. 40, no. 10, Oct. 2002.

[88] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World," *IEEE Journal on Selected Areas in Communications (J-SAC)*, vol. 20, no. 4, May 2002.

[89] P. Batchelor et al., "Ultra High Capacity Optical Transmission Networks. Final Report of Action COST 239," Jan. 1999.

[90] "SNDlib," http://sndlib.zib.de, 2015.

[91] "The GEANT website," http://www.geant.net/, 2008.

[92] D. Hock, "Analysis and Optimization of Resilient Routing in Core Communication Networks," PhD thesis, University of Würzburg, 2014.

[93] P. Francois and O. Bonaventure, "Avoiding Transient Loops during the Convergence of Link-State Routing Protocols," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, Dec. 2007.

[94] M. Shand, S. Bryant, S. Previdi, C. Filsfils, P. Francois, and O. Bonaventura, "RFC6976: Framework for Loop-Free Convergence Using the Ordered Forwarding Information Base (oFIB) Approach," http://www.rfc-editor.org/rfc/rfc6976.txt, Jul. 2013.

[95] P. Francois, M. Shand, and O. Bonaventure, "Disruption-Free Topology Reconfiguration in OSPF Networks," in *26th IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, May 2007.

[96] F. Clad, P. Merindol, J.-J. Pansiot, P. Francois, and O. Bonaventure, "Graceful Convergence in Link-State IP Networks: A Lightweight Algorithm Ensuring Minimal Operational Impact," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, Feb. 2014.

[97] Juniper Networks, "Understanding and deploying loop-free alternate feature," http://kb.juniper.net/library/CUSTOMERSERVICE/GLOBAL_JTAC/technotes/8010056-001-EN.pdf, 2009.

[98] G. Retvari, J. Tapolcai, G. Enyedi, and A. Csaszar, "IP Fast ReRoute: Loop Free Alternates Revisited," in *30th IEEE International Conference on Computer Communications (INFOCOM)*, Shanghai, China, Apr. 2011.

[99] H. Trong Viet, P. Francois, Y. Deville, and O. Bonaventure, "Implementation of a Traffic Engineering Technique that Preserves IP Fast Reroute in COMET," in *Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (ALGOTEL)*, Carry Le Rouet, France, Jun. 2009.

[100] G. Retvari, L. Csikor, J. Tapolcai, G. Enyedi, and A. Csaszar, "Optimizing IGP Link Costs for Improving IP-level Resilience," in *8th International Workshop on the Design of Reliable Communication Networks (DRCN)*, Krakow, Poland, Oct. 2011.

[101] Cisco Systems, "Cisco asr 9000 series aggregation services router mpls configuration guide, release 4.0," http://www.cisco.com/en/US/docs/routers/asr9000/software/asr9k_r4.2/mpls/configuration/guide/b_mpls_cg42asr9k_chapter_01.html, 2010.

[102] S. Rai, B. Mukherjee, and O. Deshpande, "IP Resilience within an Autonomous System: Current Approaches, Challenges, and Future Directions," *IEEE Communications Magazine*, vol. 43, no. 10, Oct. 2005.

[103] A. Raj and O. Ibe, "A Survey of IP and Multiprotocol Label Switching Fast Reroute Schemes," *Computer Networks*, vol. 51, no. 8, Jun. 2007.

[104] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, "Multiple Routing Configurations for Fast IP Network Recovery," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, Apr. 2009.

[105] S. Nelakuditi, S. Lee, Y. Yu, Z.-L. Zhang, and C.-N. Chuah, "Fast Local Rerouting for Handling Transient Link Failures," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, Apr. 2007.

[106] S. Bryant, S. Previdi, and M. Shand, "RFC6981: A Framework for IP and MPLS Fast Reroute Using Not-Via Addresses," http://www.rfc-editor.org/rfc/rfc6981.txt, Aug. 2013.

[107] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica, "Achieving Convergence-Free Routing using Failure-Carrying Packets," in *ACM SIGCOMM*, Kyoto, Japan, Aug. 2007.

[108] P.-K. Tseng and W.-H. Chung, "Joint Coverage and Link Utilization for Fast IP Local Protection," *Computer Networks*, vol. 56, no. 15, Oct. 2012.

[109] C. Filsfils, Ed., P. Francois, Ed., M. Shand, B. Decraene, J. Uttaro, N. Leyman, and M. Horneffer, "RFC6571: Loop-Free Alternate (LFA) Applicability in Service Provider (SP) Networks," http://www.rfc-editor.org/rfc/rfc6571.txt, Jun. 2012.

[110] S. Litkowski, B. Decraene, C. Filsfils, K. Raza, M. Horneffer, and P. Sarkar, "Operational management of Loop Free Alternates," http://tools.ietf.org/id/draft-ietf-rtgwg-lfa-manageability, Mar. 2015.

[111] A. Maghbouleh, Cariden, "IPFRR LFA Coverage Evaluation via Cariden MATE," in *Cisco live!*, London, UK, Jan. 2011.

[112] L. Csikor, J. Tapolcai, and G. Retvari, "Optimizing IGP link costs for improving IP-level resilience with Loop-Free Alternates," *Computer Communications*, vol. 36, no. 6, Mar. 2013.

[113] S. S. Lor and M. Rio, "Enhancing Repair Coverage of Loop-Free Alternates," in *London Communications Symposium*, London, UK, Aug. 2010.

[114] S. Cevher, T. Chen, I. Hokelek, J. Kang, V. Kaul, Y. Lin, M. Pang, M. Rodoper, S. Samtani, C. Shah, J. Bowcock, G. Rucker, J. Simbol, and A. Staikos, "An Integrated Soft Handoff Approach to IP Fast Reroute in Wireless Mobile Networks," in *International Conference on COMmunication Systems and NETworks (COMSNETS)*, Bangalore, India, Jan. 2010.

[115] S. Bryant, C. Filsfils, S. Previdi, M. Shand, and N. So, "RFC7490: Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)," http://www.rfc-editor.org/rfc/rfc7490.txt, Apr. 2015.

[116] L. Csikor and G. Retvari, "IP Fast Reroute with Remote Loop-Free Alternates: the Unit Link Cost Case," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, St. Petersburg, Oct. 2012.

[117] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications (J-SAC)*, vol. 29, no. 9, Oct. 2011.

[118] "The Internet Toplogy Zoo," http://www.topology-zoo.org/, 2015.

[119] M. Roughan, "Simplifying the Synthesis of Internet Traffic Matrices," *ACM SIGCOMM Computer Communications Review*, vol. 35, no. 5, Oct. 2005.

[120] A. Nucci, A. Sridharan, and N. Taft, "The Problem of Synthetically Generating IP Traffic Matrices: Initial Recommendations," *ACM SIGCOMM Computer Communications Review*, vol. 35, no. 3, Jul. 2005.

[121] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery*, 1st ed. Morgan Kaufmann / Elsevier, 2004.

[122] P. Cholda, A. Mykkeltveit, B. E. Helvik, O. J. Wittner, and A. Jajszczyk, "A Survey of Resilience Differentiation Frameworks in Communication Networks," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 4, Fourth Quarter 2007.

[123] A. Autenrieth and A. Kirstädter, "Engineering End-to-End IP Resilience Using Resilience-Differentiated QoS," *IEEE Communications Magazine*, vol. 40, no. 1, Jan. 2002.

[124] M. Brunner, G. Nunzi, T. Dietz, and I. Kazuhiko, "Customer-Oriented GMPLS Service Management and Resilience Differentiation," *eTransactions on Network and Service Management*, Dec. 2004.

[125] C. S. Ou, S. Rai, and B. Mukherjee, "Extension of Segment Protection for Bandwidth Efficiency and Differentiated Quality of Protection in Optical/MPLS Networks," *Optical Switching and Networking*, vol. 1, no. 1, Jan. 2005.

[126] M. Menth, R. Martin, and J. Charzinski, "Capacity Overprovisioning for Networks with Resilience Requirements," in *ACM SIGCOMM*, Pisa, Italy, Sep. 2006.

[127] M. Menth, F. Lehrieder, B. Briscoe, P. Eardley, T. Moncaster, J. Babiarz, A. Charny, X. J. Zhang, T. Taylor, K.-H. Chan, D. Satoh, R. Geib, and G. Karagiannis, "A Survey of PCN-Based Admission Control and Flow Termination," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, Third Quarter 2010.

[128] B. Briscoe et al., "An Edge-to-Edge Deployment Model for Pre-Congestion Notification: Admission Control over a DiffServ Region," http://tools.ietf.org/id/draft-briscoe-tsvwg-cl-architecture, Oct. 2006.

[129] J. Babiarz, X.-G. Liu, K. Chan, and M. Menth, "Three State PCN Marking," http://tools.ietf.org/id/draft-babiarz-pcn-3sm, Nov. 2007.

[130] A. Charny, F. L. Faucheur, V. Liatsos, and J. Zhang, "Pre-Congestion Notification Using Single Marking for Admission and Pre-emption," http://tools.ietf.org/id/draft-charny-pcn-single-marking, Nov. 2007.

[131] L. Westberg, A. Bhargava, A. Bader, G. Karagiannis, and H. Mekkes, "LC-PCN: The Load Control PCN Solution," http://tools.ietf.org/id/draft-westberg-pcn-load-control, Nov. 2008.

[132] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, Aug. 1993.

[133] B. Braden et al., "RFC2309: Recommendations on Queue Management and Congestion Avoidance in the Internet," http://www.rfc-editor.org/rfc/rfc2309.txt, Apr. 1998.

[134] K. Ramakrishnan, S. Floyd, and D. Black, "RFC3168: The Addition of Explicit Congestion Notification (ECN) to IP," http://www.rfc-editor.org/rfc/rfc3168.txt, Sep. 2001.

[135] N. Spring, D. Wetherall, and D. Ely, "RFC3540: Robust Explicit Congestion Notification (ECN) – Signaling with Nonces," http://www.rfc-editor.org/rfc/rfc3540.txt, Jun. 2003.

[136] K. Nichols, S. Blake, F. Baker, and D. L. Black, "RFC2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," http://www.rfc-editor.org/rfc/rfc2474.txt, Dec. 1998.

[137] S. Floyd, "RFC4774: Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field," http://www.rfc-editor.org/rfc/rfc4774.txt, Feb. 2007.

[138] W. Almesberger, T. Ferrari, and J.-Y. Le Boudec, "SRP: A Scalable Resource Reservation for the Internet," *Computer Communications*, vol. 21, no. 14, Nov. 1998.

[139] I. Stoica and H. Zhang, "Providing Guaranteed Services without per Flow Management," in *ACM SIGCOMM*, Boston, MA, USA, Sep. 1999.

[140] R. Szábó, T. Henk, V. Rexhepi, and G. Karagiannis, "Resource Management in Differentiated Services (RMD) IP Networks," in *International Conference on Emerging Telecommunications Technologies and Applications (ICETA 2001)*, Kosice, Slovak Republic, Oct. 2001.

[141] R. J. Gibbens and F. P. Kelly, "Distributed Connection Acceptance Control for a Connectionless Network," in *16th International Teletraffic Congress (ITC)*, Edinburgh, UK, Jun. 1999.

[142] F. Kelly, P. Key, and S. Zachary, "Distributed Admission Control," *IEEE Journal on Selected Areas in Communications (J-SAC)*, vol. 18, no. 12, Dec. 2000.

[143] M. Karsten and J. Schmitt, “Admission Control based on Packet Marking and Feedback Signalling – Mechanisms, Implementation and Experiments,” Darmstadt University of Technology, Technical Report 03/2002, 2002.

[144] M. Karsten and J. Schmitt, “Packet Marking for Integrated Load Control,” in *IFIP/IEEE Symposium on Integrated Network Management (IM)*, Nice, France, May 2005.

[145] D. J. Songhurst, P. Eardley, B. Briscoe, C. di Cairano Gilfedder, and J. Tay, “Guaranteed QoS Synthesis for Admission Control with Shared Capacity,” BT, technical report TR-CXR9-2006-001, Feb. 2006.

[146] M. Menth, S. Kopf, J. Charzinski, and K. Schrodi, “Resilient Network Admission Control,” *Computer Networks*, vol. 52, no. 14, Oct. 2008.

[147] B. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, “RFC2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification,” http://www.rfc-editor.org/rfc/rfc2205.txt, Sep. 1997.

[148] B. Braden, D. Clark, and S. Shenker, “RFC1633: Integrated Services in the Internet Architecture: an Overview,” http://www.rfc-editor.org/rfc/rfc1633.txt, Jun. 1994.

[149] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine, “RFC2998: A Framework for Integrated Services Operation over Diffserv Networks,” http://www.rfc-editor.org/rfc/rfc2998.txt, Nov. 2000.

[150] J. Wroclawski, “RFC2211: Specification of the Controlled-Load Network Element Service,” http://www.rfc-editor.org/rfc/rfc2211.txt, Sep. 1997.

[151] M. Menth and F. Lehrieder, “Performance Evaluation of PCN-Based Admission Control,” in *16th International Workshop on Quality of Service (IWQoS)*, Enschede, The Netherlands, Jun. 2008.

[152] X. Zhang and A. Charny, "Performance Evaluation of Pre-Congestion Notification," in *16th International Workshop on Quality of Service (IWQoS)*, Enschede, The Netherlands, Jun. 2008.

[153] P. Eardley, "Traffic Matrix Scenario," http://www.ietf.org/mail-archive/web/pcn/current/msg00831.html, Oct. 2007.

[154] M. Menth, J. Milbrandt, and S. Kopf, "Capacity Assignment for NAC Budgets in Resilient Networks," in *11th International Telecommunication Network Strategy and Planning Symposium (NETWORKS)*, Vienna, Austria, Jun. 2004.

[155] D. Nace and M. Pioro, "Max-Min Fairness and Its Applications to Routing and Load-Balancing in Communication Networks: A Tutorial," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, Fourth Quarter 2008.

[156] M. Menth, S. Gehrsitz, and J. Milbrandt, "Fair Assignment of Efficient Network Admission Control Budgets," in *18th International Teletraffic Congress (ITC)*, Berlin, Germany, Sep. 2003.

[157] V. Erramilli, M. Crovella, and N. Taft, "An Independent-Connection Model for Traffic Matrices," in *6th ACM Internet Measurements Conference (IMC)*, Rio de Janeiro, Brazil, Oct. 2006.

[158] M. Handley, "Why the Internet Only Just Works," *British Telecom Technology Journal*, vol. 24, no. 3, Jul. 2006.

[159] X. Zhao, D. Pacella, and J. Schiller, "Routing Scalability: An Operator's View," *IEEE Journal on Selected Areas in Communications (J-SAC)*, vol. 28, no. 8, Oct. 2010.

[160] G. Huston, "IPv4 Address Report, generated daily," http://www.potaroo.net/tools/ipv4/, 2013.

[161] D. Meyer, L. Zhang, and K. Fall, "RFC4984: Report from the IAB Workshop on Routing and Addressing," http://www.rfc-editor.org/rfc/rfc4984.txt, Sep. 2007.

[162] B. Quoitin, L. Iannone, C. de Launois, and O. Bonaventure, "Evaluating the Benefits of the Locator/Identifier Separation," in *2nd ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*, Kyoto, Japan, Aug. 2007.

[163] O. Bonaventure, "Reconsidering the Internet Routing Architecture," in *Routing Research Group Seminar at IETF-68*, Prague, Czech Republic, Mar. 2007.

[164] T. Li, "RFC6227: Design Goals for Scalable Internet Routing," http://www.rfc-editor.org/rfc/rfc6227.txt, May 2011.

[165] Internet Research Task Force (IRTF), "Routing Research Group (RRG)," http://irtf.org/rrg, 2008.

[166] "BGP Routing Analysis Reports," http://bgp.potaroo.net/, May 2015.

[167] Y. Afek, A. Bremler-Barr, and S. Schwarz, "Improved BGP Convergence via Ghost Flushing," *IEEE Journal on Selected Areas in Communications (J-SAC)*, vol. 22, no. 10, Dec. 2004.

[168] J. Chandrashekar, Z. Duan, Z.-L. Zhang, and J. Krasky, "Limiting Path Exploration in BGP," in *24th IEEE International Conference on Computer Communications (INFOCOM)*, Miami, FL, USA, Mar. 2005.

[169] D. Pei, M. Azuma, D. Massey, and L. Zhang, "BGP-RCN: Improving BGP Convergence through Root Cause Notification," *Computer Networks*, vol. 48, no. 2, Jun. 2005.

[170] W. Sun, Z. M. Mao, and K. G. Shin, "Differentiated BGP Update Processing for Improved Routing Convergence," in *14th IEEE International*

Conference on Network Protocols (ICNP), Santa Barbara, CA, USA, Nov. 2006.

[171] D. Krioukov, K. Claffy, K. Fall, and A. Brady, "On Compact Routing for the Internet," *ACM SIGCOMM Computer Communications Review*, vol. 37, no. 3, Jul. 2007.

[172] X. Zhang, P. Francis, J. Wang, and K. Yoshida, "Scaling IP Routing with the Core Router-Integrated Overlay," in *14th IEEE International Conference on Network Protocols (ICNP)*, Santa Barbara, CA, USA, Nov. 2006.

[173] W. Herrin, "Tunneling Route Reduction Protocol (TRRP)," http://bill.herrin.us/network/trrp.html, 2008.

[174] D. Jen, M. Meisel, H. Yan, D. Massey, L. Wang, B. Zhang, and L. Zhang, "Towards a New Internet Routing Architecture: Arguments for Separating Edges from Transit Core," in *7th ACM Workshop on Hot Topics in Networks (HotNets)*, Calgary, Alberta, Canada, Oct. 2008.

[175] D. Massey, L. Wang, B. Zhang, and L. Zhang, "A Scalable Routing System Design for Future Internet," in *ACM International Workshop on IPv6 and the Future of the Internet (IPv6)*, Kyoto, Japan, Aug. 2007.

[176] D. Jen, M. Meisel, D. Massey, L. Wang, B. Zhang, and L. Zhang, "APT: A Practical Transit Mapping Service," http://tools.ietf.org/id/draft-jen-apt, Nov. 2007.

[177] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "RFC6836: Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)," http://www.rfc-editor.org/rfc/rfc6836.txt, Jan. 2013.

[178] S. Brim, N. Chiappa, D. Farinacci, V. Fuller, and D. Lewis, "LISP-CONS: A Content distribution Overlay Network Service for LISP," http://tools.ietf.org/id/draft-meyer-lisp-cons, Apr. 2008.

[179] L. Mathy and L. Iannone, "LISP-DHT: Towards a DHT to Map Identifiers onto Locators," in *ACM CoNEXT workshop Re-Architecting the Internet (ReArch)*, Madrid, Spain, Dec. 2008.

[180] L. Iannone and O. Bonaventure, "On the Cost of Caching Locator/ID Mappings," in *3rd ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, New York, NY, USA, Dec. 2007.

[181] E. Lear, "RFC6837: NERD: A Not-so-novel EID to RLOC Database," http://www.rfc-editor.org/rfc/rfc6837.txt, Jan. 2013.

[182] S. Paul, J. Pan, and R. Jain, "Architectures for the future networks and the next generation Internet: A survey," *Computer Communications*, vol. 34, no. 1, Jan. 2011.

[183] D. Meyer, "The Locator Identifier Separation Protocol (LISP)," *The Internet Protocol Journal*, vol. 11, no. 1, Mar. 2008.

[184] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "RFC6830: Locator/ID Separation Protocol (LISP)," http://www.rfc-editor.org/rfc/rfc6830.txt, Jan. 2013.

[185] D. Lewis, D. Meyer, D. Farinacci, and V. Fuller, "RFC6832: Interworking LISP with IPv4 and IPv6," http://www.rfc-editor.org/rfc/rfc6832.txt, Jan. 2013.

[186] R. Whittle, "IVIP - A New Routing and Addressing Architecture for the Internet," http://www.firstpr.com.au/ip/ivip/, 2008.

[187] C. Vogt, "Six/One Router: A Scalable and Backwards Compatible Solution for Provider-Independent Addressing," in *3rd ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*, Seattle, WA, USA, Aug. 2008.

[188] S. Schuetz, R. Winter, L. Burness, P. Eardley, and B. Ahlgren, "Node Identity Internetworking Architecture," http://tools.ietf.org/id/draft-schuetz-nid-arch, Sep. 2007.

[189] R. Moskowitz and P. Nikander, "RFC4423: Host Identity Protocol (HIP) Architecture," http://www.rfc-editor.org/rfc/rfc4423.txt, May 2006.

[190] R. Atkinson, S. Bhatti, and S. Hailes, "A Proposal for Unifying Mobility with Multi-Homing, NAT, & Security," in *5th ACM International Workshop on Mobility and Wireless Access (MobiWac)*, Chania, Crete Island, Greece, Oct. 2007.

[191] R. Atkinson, S. Bhatti, and S. Hailes, "Mobility as an Integrated Service through the Use of Naming," in *2nd ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*, Kyoto, Japan, Aug. 2007.

[192] R. Atkinson, S. Bhatti, and S. Hailes, "Evolving the Internet Architecture Through Naming," *IEEE Journal on Selected Areas in Communications (J-SAC)*, vol. 28, no. 8, Oct. 2010.

[193] M. O'Dell, "GSE - An Alternate Addressing Architecture for IPv6," http://tools.ietf.org/id/draft-ietf-ipngwg-gseaddr, Feb. 1997.

[194] L. Zhang, "An Overview of Multihoming and Open Issues in GSE," *IETF Journal*, vol. 2, no. 2, Autumn 2006.

[195] A. Feldmann, L. Cittadini, W. Mühlbauer, R. Bush, and O. Maennel, "HAIR: Hierarchical Architecture for Internet Routing," in *ACM CoNEXT workshop Re-Architecting the Internet (ReArch)*, Rome, Italy, Dec. 2009.

[196] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "RFC5201: Host Identity Protocol," http://www.rfc-editor.org/rfc/rfc5201.txt, Apr. 2008.

[197] T. Li (ed.), "RFC6115: Recommendation for a Routing Architecture," http://www.rfc-editor.org/rfc/rfc6115.txt, Feb. 2011.

[198] R. Atkinson, S. Bhatti, and S. Hailes, "ILNP: Mobility, Multi-Homing, Localised Addressing and Security through Naming," *Telecommunication Systems*, vol. 42, no. 3 – 4, Dec. 2009.

[199] S. Jiang, "Hierarchical Host Identity Tag Architecture," http://tools.ietf.org/id/draft-jiang-hiprg-hhit-arch, May 2010.

[200] S. Thomson, T. Narten, and T. Jinmei, "RFC4862: IPv6 Stateless Address Autoconfiguration," http://www.rfc-editor.org/rfc/rfc4862.txt, Sep. 2007.

[201] L. Jakab, A. Cabellos-Aparicio, F. Coras, D. Saucez, and O. Bonaventure, "LISP-TREE: A DNS Hierarchy to Support the LISP Mapping System," *IEEE Journal on Selected Areas in Communications (J-SAC), Special Issue on Internet Routing Scalability*, vol. 28, no. 8, Oct. 2010.

[202] H. Luo, Y. Qin, and H. Zhang, "A DHT-Based Identifier-to-Locator Mapping Scheme for a Scalable Internet," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, Oct. 2009.

[203] H. Holbrook, B. Cain, and B. Haberman, "RFC4604: Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast," http://www.rfc-editor.org/rfc/rfc4604.txt, Aug. 2006.

[204] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "RFC4601: Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)," http://www.rfc-editor.org/rfc/rfc4601.txt, Aug. 2006.

[205] P. Savola and B. Haberman, "RFC3956: Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address," http://www.rfc-editor.org/rfc/rfc3956.txt, Nov. 2004.

[206] E. Nordmark and T. Li, "RFC4218: Threats Relating to IPv6 Multihoming Solutions," http://www.rfc-editor.org/rfc/rfc4218.txt, Oct. 2005.

[207] M. Bagnulo, "RFC6181: Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses," http://www.rfc-editor.org/rfc/rfc6181.txt, Mar. 2011.

[208] D. Saucez, L. Iannone, and O. Bonaventure, "LISP Threats Analysis," http://tools.ietf.org/id/draft-ietf-lisp-threats, Mar. 2015.

[209] D. Farinacci, D. Lewis, D. Meyer, and C. White, "LISP Mobile Node," http://tools.ietf.org/id/draft-meyer-lisp-mn, Jan. 2015.

[210] M. Wasserman and F. Baker, "RFC6296: IPv6-to-IPv6 Network Prefix Translation," http://www.rfc-editor.org/rfc/rfc6296.txt, Jun. 2011.

[211] D. Wischik, M. Handley, and M. Bagnulo Braun, "The Resource Pooling Principle," *ACM SIGCOMM Computer Communications Review*, vol. 38, no. 5, Oct. 2008.

[212] J. He and J. Rexford, "Towards Internet-wide Multipath Routing," *IEEE Network Magazine*, vol. 22, no. 2, Mar. 2008.

[213] L. Gyarmati, T. Cinkler, and T. A. Trinh, "Path-based multipath protection: resilience using multiple paths," *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 7, Nov. 2012.

[214] R. Steward (ed.), "RFC4960: Stream Control Transmission Protocol," http://www.rfc-editor.org/rfc/rfc4960.txt, Sep. 2007.

[215] A. Varga, "INET Framework for the OMNeT++ Discrete Event Simulator," http://github.com/inet-framework/inet, 2012.

[216] A. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment," in *First International Conference on Simulation Tools and*

Techniques for Communications, Networks and Systems (Simutools), Marseille, France, Mar. 2008.

[217] P. Francois, C. Filsfils, and O. Bonaventure, "Achieving Sub-50 Milliseconds Recovery Upon BGP Peering Link Failures," *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, Dec. 2007.

[218] I. Yamagata, Y. Shirasaki, A. Nakagawa, J. Yamaguchi, and H. Ashida, "NAT444," http://tools.ietf.org/id/draft-shirasaki-nat444, Jul. 2012.

[219] K. Li, S. Wang, S. Xu, and X. Wang, "ERMAO: An Enhanced Intradomain Traffic Engineering Approach in LISP-Capable Networks," in *IEEE Global Communications Conference (GLOBECOM)*, Houston, TX, USA, Dec. 2011.

[220] D. Jen, M. Meisel, D. Massey, L. Wang, B. Zhang, and L. Zhang, "APT: A Practical Tunneling Architecture for Routing Scalability," UCLA Computer Science Department, Tech. Rep. 080004, Mar. 2008.

[221] O. Hanka, C. Spleiss, G. Kunzmann, and J. Eberspächer, "A Novel DHT-Based Network Architecture for the Next Generation Internet," in *8th International Conference on Networking (ICN)*, Cancun, Mexico, Mar. 2009.

[222] X. Xu, "Routing Architecture for the Next Generation Internet (RANGI)," http://tools.ietf.org/id/draft-xu-rangi, Aug. 2010.

[223] J. Kim, L. Iannone, and A. Feldmann, "A Deep Dive into the LISP Cache and What ISPs Should Know about It," in *10th IFIP-TC6 International Networking Conference (NETWORKING)*, Valencia, Spain, May 2011.

[224] S. Brim, D. Farinacci, D. Meyer, and J. Curran, "EID Mappings Multicast Across Cooperating Systems for LISP," http://tools.ietf.org/id/draft-curran-lisp-emacs, Nov. 2007.

[225] C. Lynn, S. Kent, and K. Seo, "RFC3779: X.509 Extensions for IP Addresses and AS Identifiers," http://www.rfc-editor.org/rfc/rfc3779.txt, Jun. 2004.

[226] G. Huston, "Resource Certification," *The Internet Protocol Journal*, vol. 12, no. 1, Mar. 2009.

[227] T. Dierks and E. Rescorla, "RFC5246: The Transport Layer Security (TLS) Protocol Version 1.2," http://www.rfc-editor.org/rfc/rfc5246.txt, Aug. 2008.

[228] E. Rescorla and N. Modadugu, "RFC4347: Datagram Transport Layer Security," http://www.rfc-editor.org/rfc/rfc4347.txt, Apr. 2006.

[229] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A Measurement-Based Analysis of Multihoming," in *ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003.

[230] RIPE NCC, "RIS Statistics Report," http://www.ris.ripe.net/weekly-report/, 2013.

[231] Internet System Consortium, "The ISC Domain Survey," https://isc.org/solutions/survey, 2013.

[232] P. Martin, "Zen Internet UK Small Medium Enterprise (SME) survey," Shape the Future Limited, Tech. Rep., Nov. 2008.

[233] Verisign, "The Domain Name Industry Brief," Sep. 2013.

[234] DENIC.de, "DENIC Domainzähler – Domainentwicklung .de-Domains," http://www.denic.de/hintergrund/statistiken.html, Dec. 2013.

[235] T. Narten, R. Draves, and S. Krishnan, "RFC4941: Privacy Extensions for Stateless Address Autoconfiguration in IPv6," http://www.rfc-editor.org/rfc/rfc4941.txt, Sep. 2007.

[236] K. Sriram, Y.-T. Kim, and D. Montgomery, "Enhanced Efficiency of Mapping Distribution Protocols in Scalable Routing and Addressing Architectures," in *19th IEEE International Conference on Computer Communications and Networks (ICCCN)*, Zurich, Switzerland, Aug. 2010.

[237] O. Hanka, G. Kunzmann, C. Spleiß, J. Eberspächer, and A. Bauer, "HiiMap: Hierarchical Internet Mapping Architecture," in *First International Conference on Future Information Networks (ICFIN)*, Beijing, China, Oct. 2009.

[238] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "RFC4033: DNS Security Introduction and Requirements," http://www.rfc-editor.org/rfc/rfc4033.txt, Mar. 2005.

[239] D. Farinacci, D. Oran, V. Fuller, and J. Schiller, "Locator/ID Separation Protocol (LISP2) [DNS-based Version]," Nov. 2006. [Online]. Available: http://www.dinof.net/~dino/ietf/lisp2.ppt

[240] C. Vogt, "DNS Map – A DNS-Based Resolution System for IP Address Mappings," Ericsson, Technical Report, Feb. 2008.

[241] R. Atkinson, S. Bhatti, and S. Hailes, "ILNP - Identifier/Locator Network Protocol," http://ilnp.cs.st-andrews.ac.uk/, 2009.

[242] R. Atkinson and S. Bhatti, "An Introduction to the Identifier-Locator Network Protocol (ILNP)," in *London Communications Symposium (LCS)*, London, UK, Jul. 2006.

[243] R. Atkinson and S. Rose, "DNS Resource Records for ILNP," http://tools.ietf.org/id/draft-rja-ilnp-dns, Jul. 2011.

[244] R. Atkinson, S. Bhatti, and U. S. Andrews, "RFC6740: Identifier-Locator Network Protocol (ILNP) Architectural Description," http://www.rfc-editor.org/rfc/rfc6740.txt, Nov. 2012.

[245] P. Nikander and J. Leganier, "RFC5205: Host Identity Protocol (HIP) Domain Name System (DNS) Extension," http://www.rfc-editor.org/rfc/rfc5205.txt, Apr. 2008.

[246] O. Ponomarev and A. Gurtov, "Embedding Host Identity Tags Data in DNS," http://tools.ietf.org/id/draft-ponomarev-hip-hit2ip, Mar. 2009.

[247] S. Letong, "Layered Mapping System," http://www.ietf.org/mail-archive/web/rrg/current/msg05491.html, Dec. 2009.

[248] S. Letong, Y. Xia, W. Z. Liang, and W. Jianping, "A Layered Mapping System for Scalable Routing," http://tinyurl.com/LeXiLi09, Dec. 2009, Tsinghua University.

[249] V. Fuller, D. Lewis, V. Ermagan, and A. Jain, "LISP Delegated Database Tree," http://tools.ietf.org/id/draft-ietf-lisp-ddt, Apr. 2015.

[250] F. Maino, V. Ermagan, A. Cabellos, and D. Saucez, "LISP-Security (LISP-SEC)," http://tools.ietf.org/id/draft-ietf-lisp-sec, Apr. 2015.

[251] R. Whittle, "Ivip (Internet Vastly Improved Plumbing) Architecture," http://tools.ietf.org/id/draft-whittle-ivip-arch, Mar. 2010.

[252] R. Whittle, "DRTM - Distributed Real Time Mapping for Ivip and LISP," http://tools.ietf.org/id/draft-whittle-ivip-drtm, Mar. 2010.

[253] J. H. Wang, Y. Wang, M. Xu, and J. Yang, "Separating Identifier from Locator with Extended DNS," in *IEEE International Conference on Communications (ICC)*, Ottawa, Canada, Jun. 2012.

[254] H. Zhang and Z. Zhang, "A Hierarchical Mapping System for LISP," http://tools.ietf.org/id/draft-zhang-lisp-hms, Dec. 2012.

[255] F. Hu and J. Luo, "ID/Locator Distributed Mapping Server," http://tools.ietf.org/id/draft-hu-lisp-dht, Oct. 2009.

[256] F. Templin (Ed.), "RFC6179: The Internet Routing Overlay Network (IRON)," http://www.rfc-editor.org/rfc/rfc6179.txt, Mar. 2011.

[257] F. Templin, "The Intradomain Routing Overlay Network (IRON)," http://tools.ietf.org/id/draft-templin-ironbis, Mar. 2014.

[258] J. Pan, S. Paul, R. Jain, and M. Bowman, "MILSA: A Mobility and Multihoming Supporting Identifier Locator Split Architecture for Naming in the Next Generation Internet," in *IEEE Global Communications Conference (GLOBECOM)*, New Orleans, LA, USA, Nov. 2008.

[259] J. Pan, R. Jain, S. Paul, M. Bowman, X. Xu, and S. Chen, "Enhanced MILSA Architecture for Naming, Addressing, Routing and Security Issues in the Next Generation Internet," in *IEEE International Conference on Communications (ICC)*, Dresden, Germany, Jun. 2009.

[260] G. Chen et al., "An Incremental Deployable Mapping Service for Scalable Routing Architecture," http://tools.ietf.org/id/draft-chen-lisp-er-mo, Jul. 2009.

[261] J. Ahrenholz, "HIP DHT Interface," http://tools.ietf.org/id/draft-ahrenholz-hiprg-dht, Nov. 2009.

[262] V. Ramasubramanian and E. G. Sirer, "The Design and Implementation of a Next Generation Name Service for the Internet," in *ACM SIGCOMM*, Portland, OR, USA, 2004.

[263] L. Cheng and J. Wang, "LISP Single-Hop DHT Mapping Overlay," http://tools.ietf.org/id/draft-cheng-lisp-shdht, Jul. 2013.

[264] M. D'Ambrosio, C. Dannewitz, H. Karl, and V. Vercellone, "MDHT: A Hierarchical Name Resolution Service for Information-centric Networks," in *ACM SIGCOMM Workshop on Information-centric Networking (ICN)*, Toronto, ON, Canada, Aug. 2011.

ISSN 1432-8801