Bayerische Julius-Maximilians-Universität Würzburg
Institut für Informatik
Lehrstuhl für Verteilte Systeme
Prof. Dr. P. Tran-Gia

Resilience, Provisioning, and Control for the Network of the Future

Rüdiger Martin

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Bericht 03/08

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Herausgeber

Prof. Dr. P. Tran-Gia
Universität Würzburg
Institut für Informatik
Lehrstuhl für Verteilte Systeme
Am Hubland
D-97074 Würzburg

Tel.: +49-931-888-6630
Fax.: +49-931-888-6632
email: trangia@informatik.uni-wuerzburg.de

Satz

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

ISSN 1432-8801

Resilience, Provisioning, and Control for the Network of the Future

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Bayerischen Julius–Maximilians–Universität Würzburg

vorgelegt von

Rüdiger Martin

aus

Bamberg

Würzburg 2008

Eingereicht am: 13.06.2008
bei der Fakultät für Mathematik und Informatik
1. Gutachter: Prof. Dr.-Ing. P. Tran-Gia
2. Gutachter: Prof. Dr. Peder J. Emstad
Tag der mündlichen Prüfung: 25.07.2008

Danksagung

Die vorliegende Dissertationsschrift ist das Ergebnis mehrjähriger Arbeit. Im Laufe dieser Jahre habe ich vielfältige Unterstützung von verschiedenen Seiten erfahren, wofür ich mich herzlich bedanken möchte.

Dank gilt meinem Doktorvater Prof. Dr.-Ing. Phuoc Tran-Gia, der an seinem Lehrstuhl hervorragende Voraussetzungen für kreatives Arbeiten mit aktuellen und relevanten Fragestellungen geschaffen hat. Die wertvolle Umgebung zeichnet sich aus durch eine ausgewogene Mischung aus wissenschaftlich interessanten Industriekooperationen, Projekten im internationalen Kontext und dem direkten Zugang zur weltweiten Forschergemeinde.

Herzlicher Dank gebührt auch Herrn Prof. Dr. Peder J. Emstad für die Bereitschaft, das Zweitgutachten für meine Doktorarbeit zu erstellen. Durch seine bemerkenswert zügige Korrektur und äußerst hilfreichen Anmerkungen hat er einen raschen Abschluss meines Promotionsverfahrens innerhalb eines ehrgeizigen Zeitrahmens ermöglicht.

Prof. Dr. Klaus Schilling und Prof. Dr. Dietmar Seipel möchte ich für ihre Tätigkeit als Prüfer in der Disputation danken. Gerade trotz der terminlichen Engpässe zeigten sie außerordentliche Unterstützung durch ihre Bereitschaft, diese Aufgabe wahrzunehmen.

Die ausnehmend gute Atmosphäre am Lehrstuhl ist gekennzeichnet durch verlässliche Freundschaften, Respekt vor der Arbeit des jeweils anderen, das bereitwillige Teilen von Wissen und die keineswegs selbstverständliche Bereitschaft zur gegenseitigen Unterstützung. Besonders in Phasen akuten Zeitmangels habe ich hier große Hilfe erfahren. Die vielen wissenschaftlichen Diskussio-

nen, aber auch die gemeinsamen Unternehmungen in der Freizeit machten den Lehrstuhl zu mehr als einem Ort der Arbeit. Hierfür möchte ich meinen Kollegen Dr. Andreas Binzenhöfer, Michael Duelli, Matthias Hartmann, Robert Henjes, Tobias Hoßfeld, Frank Lehrieder, Dr. Andreas Mäder, Dr. Michael Menth, Dr. Jens Milbrandt, Simon Oechsner, Rastin Pries, Daniel Schlosser, Barbara Staehle, Dr. Dirk Staehle, Dr. Kurt Tutschku und Thomas Zinner meine Wertschätzung aussprechen.

Besonders eng durfte ich mit Dr. Michael Menth, Dr. Jens Milbrandt und Matthias Hartmann - zunächst als mein Diplomand und dann später als Kollege - in unserer Arbeitsgruppe zusammenarbeiten. Hierbei konnte ich viel lernen und interessante Ideen und Perspektiven gewinnen. Dr. Michael Menth hat die Arbeitsgruppe mit erstaunlichem und unermüdlichem Einsatz geleitet und zu - nach meiner Meinung - beachtlichen wissenschaftlichen Erfolgen motiviert. Mit Dr. Andreas Binzenhöfer und Dr. Andreas Mäder durfte ich nicht nur das komplette Studium vom Grundstudium bis zum Abschluss der Promotion bestreiten, sondern habe auch viele interessante und aufschlussreiche Gespräche über Grenzen und Möglichkeiten der Wissenschaft sowie mögliche Berufsoptionen geführt. Auch wenn ich nicht allen Kollegen hier persönlich meinen Dank für ihre individuellen Talente aussprechen kann, möchte ich aber noch einmal meine Wertschätzung betonen.

Im Laufe der Arbeit konnte ich interessante Erfahrungen im Umgang mit Studenten während meiner Tätigkeit als Fachstudienberater aber auch in Lehrveranstaltungen von der Vorlesung bis zur Diplomarbeit machen. Besonders danken möchte ich hier Matthias Hartmann und Michael Hemmkeppler für die fruchtbare Zusammenarbeit im Laufe ihrer Diplomarbeiten. Zudem gewährte mir meine Zugehörigkeit zu unterschiedlichen Universitätsausschüssen spannende Einblicke in die interne Arbeitsweise einer Hochschule, wofür der Fachschaft Dank gebührt.

In Kooperationen mit Wissenschaft und Industrie ist man auf eine gute Zusammenarbeit mit den Projektpartnern angewiesen. Hierbei habe ich stets sehr gute Erfahrungen gesammelt und möchte besonders Herrn Dr. Joachim Charzinski

danken. Der Anstoß zur Beschäftigung mit dem in dieser Arbeit dargestellten und in der Folge sehr erfolgreichen Vergleich von Zugangskontrolle und Überdimensionierung in Kernnetzen ist ihm geschuldet.

Schließlich möchte ich noch den zahlreichen Studenten und den Mitarbeitern danken, die ich in der Katholischen Hochschulgemeinde (KHG) Würzburg kennenlernen durfte. Die KHG Würzburg ist ein außergewöhnlicher Ort, ein Platz für ganz wichtige Bildung über das Fachliche hinaus. Dort fand ich viele Freunde, wurde reich beschenkt und unschätzbar unterstützt - vor allem und nicht zuletzt durch ihren Leiter P. Johann Spermann SJ. In der KHG wurde mir auch immer wieder deutlich, dass Wissenschaft nicht alleine um der Technik willen, sondern stets für den Menschen da sein muss! In enger Verbindung zur KHG steht auch der Monteverdichor Würzburg, wo ich über viele Jahre künstlerischen und musikalischen Ausgleich auf hohem Niveau fand und viele Freundschaften schließen durfte. Aus diesem Chor entstand eine kleine Gruppe, mit der ich mich regelmäßig in der Mensa zum Mittagessen getroffen habe und bei spannenden Gesprächen unterschiedlichsten Inhalts Energie und Unterstützung für die Arbeit bekommen, wofür ich diesen Freunden herzlich danken möchte.

Mein abschließender Dank richtet sich an meine Familie, nämlich meine Eltern Josef und Margarete Martin, meine Geschwister Uwe und Myriam und an meine Großeltern Andreas und Margarete Martin. In meiner Familie habe ich immer große Unterstützung, wichtigen Rückhalt und intensive Förderung in jeder Hinsicht erfahren.

Acknowledgements

The PhD thesis at hand is the result of the work of many years. During those years I received manifold support from different sides and, thus, would like to express my gratitude.

I owe thanks to my supervisor Prof. Dr.-Ing. Phuoc Tran-Gia who laid excellent foundations for the creative work on current and relevant research questions at his department. The valuable environment is distinguished by a sound combination of scientifically interesting industry co-operations, projects in international context, and the contact to the worldwide research community.

I also owe special thanks to Prof. Dr. Peder J. Emstad for serving as my second supervisor. His remarkably fast review and his very helpful remarks made the timely completion of my graduation within an ambitious time schedule possible.

I would like to thank Prof. Dr. Klaus Schilling and Prof. Dr. Dietmar Seipel for acting as examiners in my disputation besides my supervisor. Even in spite of a high workload they showed great support by their willingness to perform this important task.

Reliable friendship, respect for the work of others, unhesitating sharing of knowledge and willingness for mutual support - that is not at all a matter of course - characterize the excellent atmosphere at the department. Especially when the time available is short, I received great help. Many scientific discussions, but also joint activities after work turned this department into more than just a place for work. That is why I would like to express my appreciation to my colleagues Dr. Andreas Binzenhöfer, Michael Duelli, Matthias Hartmann, Robert Henjes, Tobias Hoßfeld, Frank Lehrieder, Dr. Andreas Mäder, Dr. Michael Menth, Dr.

Jens Milbrandt, Simon Oechsner, Rastin Pries, Daniel Schlosser, Barbara Staehle, Dr. Dirk Staehle, Dr. Kurt Tutschku and Thomas Zinner.

I worked especially closely with Dr. Michael Menth, Dr. Jens Milbrandt, and Matthias Hartmann - first as my diploma student and later as one of my colleagues - in our working group. Here, I learned a lot and obtained interesting ideas and perspectives. Dr. Michael Menth led this working group with amazing and sedulous dedication and motivated it to - in my opinion - remarkable scientific success. Together with Dr. Andreas Binzenhöfer and Dr. Andreas Mäder I made my way through the university for more than ten years beginning from undergraduate to doctoral studies and we also had many interesting and revealing conversations about limits and possibilities of science as well as options for a future career. Even though I cannot thank all my colleagues personally for their individual gifts here, I still would like to emphasize my gratitude to them.

During this work I made many interesting experiences concerning the interaction with students while acting as student advisor but also during courses from lectures to diploma theses. In particular, on this occasion I would like to thank Matthias Hartmann and Michael Hemmkeppler for the fruitful co-operation during their diploma theses. Besides, I got many interesting insights into the internal functioning of a university as a member of various university commissions for which I would like to thank the student representatives.

In co-operation with partners from science and industry one must rely on active teamwork where I only made good experiences. In this context I would like to thank especially Dr. Joachim Charzinski. The impulse to look at the comparison of admission control and over-provisioning, which is presented in this thesis and later became scientifically very successful, came from him.

Finally I would like to thank all those students and the professional staff from the catholic student community Würzburg (KHG). The KHG is an extraordinary place, a place for a very important form of education that exceeds the pure technical aspects. There I have found many friends, received so many values and insights, and obtained priceless support - especially and last but not least from its supervisor F. Johann Spermann SJ. There it became also clear to me that science

should not exist purely for the sake of technology, but its purpose must always be to serve humans! In close connection with the KHG, there is the Monteverdi Choir Würzburg, where I found artistic and musical balance on a high level and made many friends. From this choir, there developed a small group which I have been meeting regularly for lunch in the food court where I received energy and support for my work during thrilling conversations of various content. I would like to say thank you to those friends.

My final thank-you is directed to my family, namely my parents Josef and Margarete Martin, my siblings Uwe and Myriam, and my grandparents Andreas and Margarete Martin. In my family I always received great support, important backing, and intense encouragement in all respects.

Contents

1 Introduction

The Internet sees an ongoing transformation process from a single best-effort service network into a multi-service network. In addition to traditional applications like e-mail, WWW traffic, or file transfer, future generation networks (FGNs) will carry services with real-time constraints and stringent availability and reliability requirements like Voice over IP (VoIP), video conferencing, virtual private networks (VPNs) for finance, other real-time business applications, tele-medicine, or tele-robotics. Hence, quality of service (QoS) guarantees and resilience to failures are crucial characteristics of an FGN architecture. At the same time, network operations must be efficient. This necessitates sophisticated mechanisms for the provisioning and the control of future communication infrastructures. In this work we investigate such mechanisms for resilient FGNs.

There are many aspects of the provisioning and control of resilient FGNs such as traffic matrix estimation, traffic characterization, traffic forecasting, mechanisms for QoS enforcement also during failure cases, resilient routing, or scalability concerns for future routing and addressing mechanisms. In this work we focus on three important aspects for which performance analysis can deliver substantial insights: load balancing for multipath Internet routing, fast resilience concepts, and advanced dimensioning techniques for resilient networks.

1.1 Aspects of Resilience, Provisioning, and Control under Study

Routing in modern communication networks is often based on multipath structures, e.g., equal-cost multipath routing (ECMP) in IP networks, to facilitate traffic engineering and resiliency. When multipath routing is applied, load balancing algorithms distribute the traffic over available paths towards the destination according to pre-configured distribution values. State-of-the-art load balancing algorithms operate either on the packet or the flow level. Packet level mechanisms achieve highly accurate traffic distributions, but are known to have negative effects on the performance of transport protocols and should not be applied. Flow level mechanisms avoid performance degradations, but at the expense of reduced accuracy. These inaccuracies may have unpredictable effects on link capacity requirements and complicate resource management. Thus, it is important to exactly understand the accuracy and dynamics of load balancing algorithms in order to be able to exercise better network control. Knowing about their weaknesses, it is also important to look for alternatives and to assess their applicability in different networking scenarios. This is the first aspect of this work.

Component failures are inevitable during the operation of communication networks and lead to routing disruptions if no special precautions are taken. In case of a failure, the robust shortest-path routing of the Internet reconverges after some time to a state where all nodes are again reachable – provided physical connectivity still exists. But stringent availability and reliability criteria of new services make a fast reaction to failures obligatory for resilient FGNs. This led to the development of fast reroute (FRR) concepts for MPLS and IP routing. The operations of MPLS-FRR have already been standardized. Still, the standards leave some degrees of freedom for the resilient path layout and it is important to understand the tradeoffs between different options for the path layout to efficiently provision resilient FGNs. In contrast, the standardization for IP-FRR is an ongoing process. The applicability and possible combinations of different concepts

still are open issues. IP-FRR also facilitates a comprehensive resilience framework for IP routing covering all steps of the failure recovery cycle. These points constitute another aspect of this work.

Finally, communication networks are usually over-provisioned, i.e., they have much more capacity installed than actually required during normal operation. This is a precaution for various challenges such as network element failures. An alternative to this capacity overprovisioning (CO) approach is admission control (AC). AC blocks new flows in case of imminent overload due to unanticipated events to protect the QoS for already admitted flows. On the one hand, CO is generally viewed as a simple mechanism, AC as a more complex mechanism that complicates the network control plane and raises interoperability issues. On the other hand, AC appears more cost-efficient than CO. To obtain advanced provisioning methods for resilient FGNs, it is important to find suitable models for irregular events, such as failures and different sources of overload, and to incorporate them into capacity dimensioning methods. This allows for a fair comparison between CO and AC in various situations and yields a better understanding of the strengths and weaknesses of both concepts. Such an advanced capacity dimensioning method for resilient FGNs represents the third aspect of this work.

1.2 Outline

This monograph is organized as follows. Chapter 2 describes the basic principles of the provisioning and control mechanisms for resilient FGNs covered in this work and lays the foundation for the following chapters. In this chapter we also give a short overview of various quality concepts since high quality communication is the main goal of provisioning and control for resilient FGNs. Chapters 3, 4, and 5 then study the three aspects load balancing for multipath Internet routing, fast resilience concepts, and provisioning of resilient networks in detail. At the end of each chapter, we briefly summarize the main findings for the respective topic. Finally, Chapter 6 concludes this work.

2 Resilient Network Provisioning and Control

In this chapter we describe the basic principles needed in the course of our work. For this purpose, we first briefly explain basic quality concepts. The possibility to provide high quality services is one of the main design goals for future generation networks (FGNs). This is the background for our work and clarifies the definitions we have in mind when referring to the abstract term “quality”.

Thereafter, we introduce the fundamentals of the mechanisms for resilient network provisioning and control under study: load balancing in Sections 2.2 and 2.3, fast resilience concepts in Section 2.4, and provisioning of resilient networks in Section 2.5. At the end of each section, we also briefly indicate our contribution to the respective topic.

2.1 Basic Quality Concepts

Any success in business largely depends on the ability of the engaged companies to deliver an attractive degree of quality to their customers at competitive costs. If customers are dissatisfied with the received services, they often consider defection to competitors. They further tell relatives and friends about their bad experiences leading to a chain reaction. At the same time, assessment of customer satisfaction is a difficult task. Not all unhappy customers call the company support or complaint division to verbalize their dissatisfaction, some customers simply leave. Thus, it is important to provision the offered services appropriately,

to constantly assess their quality, and to take appropriate action in time if problems arise. For this purpose, definitions and concepts are required that describe what the abstract term “quality” means in the given context and how to achieve it.

The International Organization for Standardization (ISO) issued two basal definitions of quality in their standards 8402 and 9000. ISO 8402 [25] defines quality as “the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs”. This definition was replaced in ISO 9000 [26] by the “degree to which a set of inherent characteristics fulfils requirements”.

In this work we focus on mechanisms that achieve reliable services and thereby quality in communication networks. Due to the possibilities of modern broadband connections to offer real-time and other critical services this also became an issue for the initially best-effort Internet. In the context of communication networks, there are several views on quality that evolved over time. In the following we briefly outline some important concepts.

Grade of Service (GoS)

In the public switched telephone network (PSTN), the grade of service (GoS) is the probability that a call is blocked or delayed at its establishment or release for a longer period than a given interval due to limited system resources [27,28]. This value is usually assessed during the busy hour which is the time of the heaviest traffic intensity in the network. GoS is also often seen as measurable parameters pertaining to the traffic performance of a telecommunication network. Thus, GoS standards are required to achieve quality of service (QoS, see below), but GoS is not necessarily a subset of QoS [29]. According to [29], GoS takes the network point of view while QoS takes the user point of view.

Several recommendations of the international telecommunication union (ITU) telecommunication standardization sector (ITU-T) cover aspects of GoS. Recommendation E.543 (1988) [30], e.g., specifies an internal loss probability of 0.002 during normal and of 0.01 during high load for international telephone exchanges.

Quality of Service (QoS)

The traffic engineering term “Quality of Service” (QoS) has its roots in the telecommunication world [31]. Today it is widely used with varying meaning in different technological fields. QoS is a notion that evolved over time and is therefore hard to grasp. It is often not defined at all, defined only implicitly, or even misused.

In packet-switched communication networks, QoS in a narrow sense often refers to network parameters such as delay, jitter, packet loss, and throughput. But depending on the context, it may as well denote the perceived quality level or the collection of networking technologies and techniques such as resource reservation control mechanisms that provide guarantees on the network behavior. Further, QoS is also used to express a degree of excellence in a comparative sense relative to other technologies for technical evaluations.

Several standardization bodies developed QoS frameworks describing their interpretation of QoS, e.g the ATM Forum (now incorporated into the IP/MPLS Forum) for the asynchronous transfer mode (ATM) [32], the 3rd Generation Partnership Project (3GPP) for the universal mobile telecommunication system (UMTS) [33], the Internet Engineering Task Force (IETF) for the Internet [34, 35], and the ITU-T for communication networks [31].

We now briefly describe QoS as seen by the IETF since their QoS architectures were specifically developed for the Internet. After that we also give an overview of QoS as seen by the ITU-T since this is the most comprehensive QoS definition.

QoS within the IETF In RFC 2216 [36] QoS is defined as the quality referring “to the nature of the packet delivery service provided, as described by parameters such as achieved bandwidths, packet delay, and packet loss rates”. Hence, QoS is mainly a question of routing packets through the network.

In the context of the IETF, QoS is often associated with the IETF QoS architectures. The Integrated Services (IntServ) architecture [37, 38] gives guarantees for individual flows based on the distinction between real-time and elastic flows.

For each flow a path is reserved through the network using, e.g., the resource reservation protocol (RSVP) [39]. In [1] we presented a performance evaluation of different reservation protocols. Since reserving a path for each flow requires every node to maintain flow states, scalability issues arise. Differentiated services (DiffServ) [35], on the contrary, follows a fundamentally different stateless core approach. Packets of different flows are aggregated into service classes that obtain differentiated treatment, i.e., treatment better or worse relative to other classes, at the network nodes based on the per-hop behavior (PHB) of their service class specification.

		Service quality criteria						
		Speed 1	Accuracy 2	Availability 3	Reliability 4	Security 5	Simplicity 6	Flexiblity 7
Service function								
Service management	1 Sales and pre-contract activities							
	2 Provision							
	3 Alteration							
	4 Service support							
	5 Repair							
	6 Cessation							
Connection quality	7 Connection establishment							
	8 Information transfer							
	9 Connection release							
10 Billing								
11 Network/Service management by customer								

Figure 2.1: *Matrix for the identification of QoS criteria of different service functions of a telecommunication service according to [40]*

QoS within the ITU-T ITU-T Rec. E.800 [41] defines QoS as "the collective effect of service performance which determines the degree of satisfaction of a user of the service".

Based on this QoS definition and the framework for QoS implementation in E.800, ITU-T Rec. G.1000 [40] presents an application oriented QoS framework that sees quality from multiple viewpoints: the customer's and the service provider's viewpoints.

Classical QoS approaches that see QoS only in terms of measurable network performance parameters such as delay, packet loss, jitter, and throughput follow a bottom-up approach. They specify limits for the network parameters that must not be exceeded for different services in order to meet the customers' expectations. The ITU-T framework is a top-down approach. Based on the QoS definition from above, it breaks down the users' expectations and the resulting quality criteria of a service into different functional requirements. These functional requirements, then, are used by the service provider for the provisioning of the service and some of the functional requirements must be mapped onto appropriate network performance parameters. Figure 2.1 shows the matrix from G.1000 [40] that is used to identify the quality requirements for different functions of a telecommunication service. The service functions 7 "connection establishment" and 9 "connection release" partly relate to GoS, function 8 "information transfer" to the classical definition of QoS corresponding to measurable network performance parameters.

The introduction and the operation of a quality service necessitate constant monitoring whether the viewpoints of the customers and the provider match. The QoS promised by a provider may differ from the achieved QoS. And the achieved QoS may be differently perceived by the customers and not match their requirements. This is reflected by the four viewpoints on QoS within the ITU-T framework shown in Figure 2.2.

This QoS definition incorporates next to the objective technical aspects of communication also subjective expectations of customers. In this context the ITU-T further specified, e.g., a model called perceptual evaluation of speech quality (PESQ) [42] to predict the customers' perceived mean opinion score (MOS)

for voice quality. This is an approach to make subjective perceptions objectively measurable. A broad overview over QoS and network performance can be found in [31].

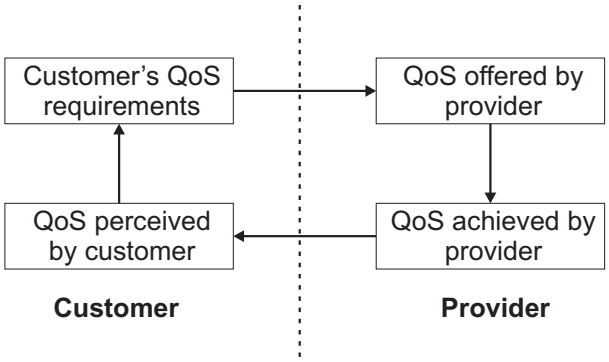


Figure 2.2: *The four viewpoints on QoS according to [40]*

Quality of Resilience (QoR)

The authors of [43] introduced the concept Quality of Resilience (QoR). Resilience is the ability of a network to provide and maintain an acceptable level of service in the face of various challenges to normal operations such as link or router failures. QoR measures the availability of a network with respect to a given service. In a small application-dependent time interval Δt, a service is either available or not. QoR measures the downtime distribution to asses the availability of the network.

Quality of Experience (QoE)

Quality of end-user experience or simply quality of experience (QoE) is a subjective quality measure. Similar to QoS, many different definitions exist. However, it is mainly used to describe the perception of end-users on how usable services are [44].

Since most QoS definitions cover the purely technical aspects of quality, QoE emphasizes the user's view. In the context of the comprehensive ITU-T QoS framework described above (cf. Figure 2.2), it can be seen as the viewpoint "Qos perceived by customers", but it also includes additional aspects such as satisfaction with provided content or user equipment usability. QoE is expressed in human feelings and therefore hard to contract.

In the context of QoE, QoS is often seen as the measurable network performance parameters only that contribute towards the user satisfaction.

2.2 Load Balancing for Multipath Internet Routing

Traffic splitting across multiple paths is an important functionality in modern communication networks. Many commercial router vendors, such as Cisco and Juniper, provide basic support for this feature in their products [45–47]. Load balancing algorithms distribute the traffic over multiple paths towards its destination according to pre-configured distribution values. In IP networks, multipath routing is typically implemented by the equal-cost multipath (ECMP) option for the most widely used interior gateway protocols (IGPs) Open Shortest Path First (OSPF) [48] and Intermediate System to Intermediate System Protocol (IS-IS) [49, 50]. Some proprietary router implementations also offer ECMP-capable versions of the Routing Information Protocol (RIP) [51]. With multiprotocol label switching (MPLS) technology, the ingress router may forward data over disjoint label switched paths (LSPs).

Multipath routing is used for traffic engineering (TE) purposes in general. Specifically, it makes data forwarding more robust against network failures [52] and helps to minimize backup capacities if capacity sharing is allowed [18]. Another potential application includes adaptive multihoming, which allows a stub domain to adaptively split its traffic across multiple access links connected to different ISPs to optimize performance and costs [53, 54].

In case multiple paths exist to reach the destination – so called path diversity – and the paths are used in parallel, packet reordering may occur. Packet reordering is a phenomenon generally known in the Internet that occurs often but not only due to traffic splitting over multiple paths. Packet reordering is generally considered to be caused by transient conditions, pathological behavior, and erroneous implementations. Oscillations or "route flaps" among routes with different round-trip times (RTTs) are common causes for packets delivered out-of-order [55, 56]. However, the authors of [57] find also other non-pathological sources for packet reordering due to increased parallelism in modern Internet equipment.

While packet reordering in routers is actually not explicitly disallowed in the Internet [58], it has a detrimental effect mainly on the performance of TCP since TCP interprets reordering as a sign of congestion [59]. Therefore, much effort has been put into making TCP more robust to retransmissions [56, 60–62]. However, this is not state of the art in current TCP implementations and some UDP-based applications such as VoIP are sensitive to packet reordering as well [59].

Hence, for multipath Internet routing, load balancing algorithms must be used that keep packet reordering low or avoid it completely. This contradicts the design goal of distributing the load as accurately as possible over the available paths according to the desired distribution values to make, e.g., traffic engineering most effective. The following sections discuss design principles for load balancing algorithms.

2.2.1 Definition: Load Balancing

In general, load balancing refers to the distribution of service requests to multiple service entities. The service entities are all equivalent with respect to the offered service but they may have different service capacity. The fraction assigned to each of them is given by a load distribution or load balancing function.

2.2.2 Load Balancing Paradigms

There are different paradigms for load balancing algorithms: traffic splitting on the packet level, on the flow level, and with flowlet switching. This directly influences the granularity at which traffic is split over the paths. Hence, each paradigm exhibits different properties concerning accuracy and potential packet reordering.

Load Balancing on the Packet Level

Packet-based load balancing offers the finest granularity. It is the most intuitive and simplest way to balance load. On the packet level, the arriving packets are distributed packet-by-packet over the alternative outgoing interfaces in a round-robin fashion. Since the available paths may have different capacities and the packets vary in size, algorithms like Deficit Round-Robin (DRR) [63] are used to achieve the desired traffic split. This packet-based solution is a standard implementation in many state-of-the-art routers. Its accuracy is very high [64]. However, the disadvantage is obvious: varying link and buffer delays on different paths lead to heavy packet reordering. Since packet reordering severely degrades the throughput of transport layer protocols such as TCP [55, 56, 59–62], this is not an option for TCP/IP networks (cf. Section 2.2.3).

Load Balancing on the Flow Level

To avoid packet reordering, all packets forming a flow should follow the same path which requires load distribution on the flow level. An intuitive algorithm to achieve this is recording the identifier (ID) of a flow together with its outgoing interface in a lookup table. The flow ID consists of invariant header fields such as source and destination address, possibly including the protocol number as well as source and destination port numbers. When the first packet of a flow arrives, an interface is selected and the information is inserted into a lookup table, which allows to forward succeeding packets to the same interface. However, the memory requirements of such a table are very expensive for a large number of flows

and the lookup in a large table is time-consuming. Since the number of concurrent flows can be in the order of tens of thousands [65], a solution that requires per-flow state is not viable for scalability reasons. Therefore, Cisco introduced a limited-size cache [66] and calls it “fast switching”. Whenever the cache is full at the arrival of a new flow, the oldest flow entry of the lookup table is replaced. This possibly leads to packet reordering if this flow is still active. So other approaches that avoid the problem of large lookup tables are required.

Hash-based Load Balancing The problem of large lookup tables can be avoided by hash-based algorithms. A hash function provides a mapping from the large space of flow IDs to a smaller space of, e.g., integral numbers. Another operation maps the hash value to outgoing interfaces. By the application of this concept, no per-flow states are kept since the extended hash function derives the outgoing interface from the flow ID. Depending on the actual implementation of the extended hash function, only a small lookup table of limited size is necessary to store the mapping between hash values and outgoing interfaces. Therefore, hash-based load balancing scales well with an increasing number of flows. Different hash functions are analyzed in [67]. The authors conclude that the 16-bit cyclic redundancy check (CRC) function [68–70] achieves good load balancing performance among the examined functions for static hashing. A further modulo operation maps the obtained hash values to the outgoing interfaces. As a simple alternative, the exclusive “OR” of source and destination IP addresses yields also good results.

Prefix-based Load Balancing Similar to hash-based load balancing schemes, prefix-based methods [64] require a small, limited-size lookup table only. The table stores a mapping between destination prefixes and outgoing interfaces. Initially, the table is empty. An incoming packet creates a new entry if the table is not full and no exact-match against its destination IP address exists. If the table is full, the algorithm examines the longest prefix match between the destination IP and each entry in the table. An entry is selected and determines

the outgoing interface if the match between itself and the destination IP is longer than the longest prefix match between every pair of entries in the table. Otherwise, the match between the destination IP and all entries is shorter than for at least one pair of entries in the table. Then, the algorithm merges two entries with the longest prefix match and creates a new entry for the packet's destination IP. New and merged entries are mapped to the path with the lightest load. The authors of [64] find that prefix-based load balancing algorithms have problems with popular routes and exhibit generally less potential than hash-based algorithms since they consider destination IP addresses only.

Static and Dynamic Load Balancing Load balancing algorithms on the packet level are intrinsically dynamic since a new decision which outgoing interface to use is made for every packet arrival. Load balancing algorithms on the flow level can be distinguished into static and dynamic mechanisms. If the mapping between flows and their outgoing interface is never changed, the algorithms are referred to as static. A static mapping makes it hard or even impossible to react to load imbalances. Load imbalances arise due to the stochastic nature of the flows. Both the flow rate variability — flows differ widely in their sizes and rates — and the number of simultaneous flows influence the load balancing accuracy. Dynamic load balancing, i.e. flow reassignment to other interfaces, helps to redistribute the traffic load. They periodically recompute the mapping between flows and their outgoing interfaces to account for the non-uniformity of flows. With lookup tables, new flows can be assigned intentionally to underloaded links. In case of hash-based load balancing, the assignment function from the space of the hash values to the outgoing interfaces is modified. The authors of [71] developed a dynamic algorithm that periodically reassigns flows from the most overloaded link to the most underloaded link.

Load Balancing Using Flowlet Switching

The idea of flowlet-based load balancing was introduced first in [72] and further elaborated in [73]. It exploits the following observation. Two successive packets can follow different paths without risk of packet reordering if their inter-arrival time is larger than the maximum delay difference between the paths. This leads to the definition of flowlets. Flowlets are packet bursts of one TCP flow spaced by a minimum interval δ. If the parameter δ is larger than the maximum delay difference between the possible paths towards the destination, two successive flowlets can follow different paths without packet reordering. Hence, flowlet switching operates at a coarser granularity than load balancing on the packet level where a new decision is made for every packet arrival, but on a finer granularity than load balancing on the flow level. The authors of [72, 73] suggest a load balancing algorithm called flowlet aware routing engine (FLARE) that implements this concept. FLARE measures the delay on the multipath and adjusts the parameter δ accordingly to their maximum delay difference. Flowlet switching is applicable to traffic that exhibits bursty behavior.

2.2.3 Applications and Problems

Multipath forwarding may be applied whenever packets can be sent over alternative paths. It can be implemented by different algorithms that exhibit algorithm-dependent difficulties.

Multipath Forwarding Applications

There are various technical solutions incorporating load balancing for multipath forwarding. An overview of different multiplath structures and their applicability can be found in [20].

Equal-Cost Multipath Routing Multipath routing is useful for traffic engineering purposes. In IP networks, it is implemented by the equal-cost mul-

tipath (ECMP) routing option which forwards packets from a certain location to their destination over any path with a shortest distance according to the link costs in the network. Multiple paths towards a destination can be obtained by the choice of suitable link costs. ECMP is a standard option of the OSPF [48] and the IS-IS [49, 50] routing protocols. Some proprietary router implementations also allow ECMP with RIP and other routing protocols [51]. Usually, traffic is forwarded equally over any interface leading to the destination over a shortest path. In contrast, dynamic traffic engineering mechanisms like the adaptive multipath routing (AMP) [74] — based on relaxed ECMP multipath forwarding structures — and REPLEX [75] — applicable for general multipath structures — dynamically signal the load distribution functions.

Resilient Multipath Routing Resilient multipath routing offers alternative paths such that there is still a working path in case of a failure. This property of multipath routing is deliberately exploited in [76] which is different from the standard IP routing. As long as at least two forwarding alternatives exist, the traffic is distributed in each node according to a given load balancing function.

Self-Protecting Multipath The self-protecting multipath (SPM) consists of disjoint label switched paths (LSPs) and provides at the source several alternatives to forward the traffic to the destination. If one of the paths fails, the traffic is transmitted over the working paths. The traffic distribution over the disjoint path follows an optimized load balancing function which minimizes the required backup capacity.

Problems of Load Balancing for Multipath Forwarding

New problems arise in networking due to the use of load balancing per se or due to the inaccuracy of load balancing.

Problems due to the Use of Load Balancing Different paths between a pair of nodes may have different maximum transfer units (MTUs) [51], leading to problems when multiple paths are used. Furthermore, popular debugging utilities like ping and traceroute may become unreliable for two reasons: either succeeding probes may follow different paths or the diagnosed path does not coincide with the data path. The authors of [77, 78] therefore created a tool that measures and characterizes load-balanced paths.

However, the main problem is that different queuing, transmission, and propagation latencies along different path may lead to packet reordering. Reordered packets have a detrimental effect on the throughput of transport layer protocols like TCP [55, 56, 59–61] and also affect some UDP-based applications such as VoIP [59]. Therefore, all packets of a single flow should be forwarded along the same path in order to avoid packet reordering. This demand for load balancing on the flow level has a significant influence on the design of load balancing algorithms.

Problems due to Load Balancing Inaccuracy The resource management entity of a network may configure the load balancing function of a network to optimize the network operation [79]. Then, overload may occur on some links if the realized load balancing proportions in the network deviate significantly from the corresponding configured values. This is problematic if the QoS of real-time traffic is protected by admission control but an unexpected traffic distribution corrupts the planned traffic load on the links [80]. Similarly, backup capacities may not suffice for the SPM or the above mentioned resilient multipath structures if the real traffic distribution in the network deviates from the pre-configured values.

We will see in Chapter 3 that both the reordering probability and the load balancing accuracy depend on the applied distribution algorithms.

2.2.4 Our Contribution towards Load Balancing

Most state-of-the-art routers implement either load balancing on the packet level or hash-based mechanisms. Since load balancing on the packet level suffers from packet reordering, we concentrate on the performance of hash-based mechanisms in this work. Even though an extensive survey of the load balancing qualities of different hash functions has been presented in [67], there is only little literature about dynamic load balancing for multipath forwarding. The load balancing accuracy in [71] was estimated based on long-term traffic distributions which lead to the conclusion that the load balancing accuracy is fairly good. This is an intuitive result provided that the hash functions spread large sets of flow IDs evenly over their codomain. Studies of the load balancing accuracy distribution over time are still missing. However, they are required to decide whether forwarding inaccuracies due to load balancing must be considered by the resource management of a network.

Hence, in Chapter 3, we present a new classification of hash-based algorithms that includes existing and new ones. Further, we compare their load balancing accuracy and their dynamics in terms of their flow reassignment rates, i.e., their behavior over time. For this purpose, we develop a performance evaluation framework for load balancing algorithms.

2.3 Load Balancing Scenarios in Communication Systems

The term load balancing is used in various application scenarios in communication systems. Its meaning and the problems and solutions involved differ depending on the scenario under study. In this section we characterize other important application examples for load balancing to distinguish them from load balancing for multipath forwarding. Since the term load balancing has such a broad meaning, we limit the discussion to application scenarios directly related to packet

forwarding and, in addition, give an interesting example for a load balancing algorithm for multiple servers with minimal disruption in case of reconfiguration.

2.3.1 Load Balancing for Inverse Multiplexing

A single point-to-point link on the network layer may be provided by bundling multiple parallel links on the link layer (cf. Figure 2.3). The packets of a traffic aggregate are distributed over these parallel links for transmission. This approach is called inverse multiplexing [81,82] because multiplexing normally means putting multiple small flows onto a large trunk. Various inverse multiplexing schemes for packet data networks have been proposed and implemented, including incremental bandwidth on demand (BONDING) [83] and the multilink point-to-point protocol (MP) [84]. Typical implementations use packet- or byte-based round-robin scheduling [85], which achieves a well balanced load on the separate links.

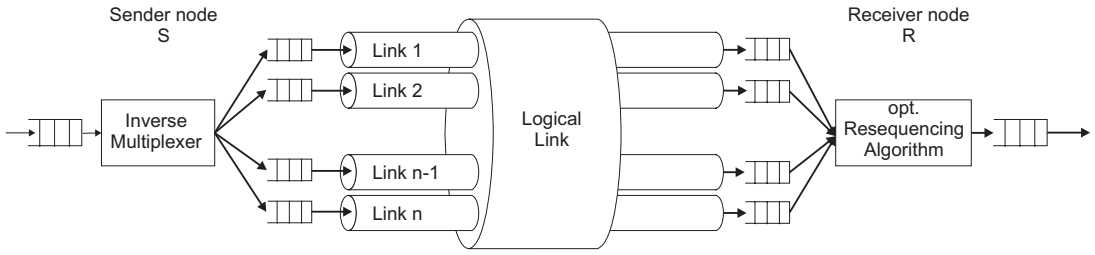


Figure 2.3: *Inverse Multiplexing bundles multiple parallel links to a single point-to-point link. Depending on the algorithm, an optional resequencing algorithm prevents packet reordering.*

The delay of the individual physical links varies due to different link capacities, due to different packet sizes, or due to buffering on the actual physical link. Thus, similar to multipath forwarding (c.f. Section 2.2.3), packet reordering with all its implications on, e.g., TCP throughput degradation, is also an issue. However, the delay variations are significantly smaller in contrast to load balancing on the IP or MPLS layer. In addition, an intelligent packet scheduling at the source allows for efficient packet resequencing at the sink for point-to-point links. For

example, the strIPe protocol [85] does scheduling and resequencing for this purpose by using Surplus Round-Robin (SRR) on both sides of the physical link. The sender transmits periodic synchronization packets to survive packet losses that cause sender and receiver to go out of packet order synchronization. Since multipaths in IP networks may be significantly more complex than multiple parallel links, this solution cannot be adopted for the multipath forwarding problem in Section 2.2.3.

Another implementation approach for inverse multiplexing renounces on packet resequencing. It avoids packet reordering within flows by a hash-based mapping between flows and physical links [86]. The scheme monitors buffer occupancies. To prevent packet loss, it reacts to load imbalances by moving flows from links with high buffer occupancy to those with low values. The load balancing objective for multipath routing, however, is not the prevention of overload on the next link. It aims at spreading the traffic for a certain destination over several links according to a given load balancing function. There is no direct connection between buffer state of the next link and the state of an entire path. Therefore, buffer occupancy is not a good indicator for unbalanced load in case of load balancing for multipath routing. Instead, rate measurements are required to detect imbalances between the individual paths.

In the context of ATM networks, the inverse multiplexing for ATM (IMA) [87] was standardized by the ATM Forum. Here, the point-to-point cell streams are dispatched in a synchronous and cyclic order among the physical links and the cells carry sequence numbers allowing the use of straightforward re-assembly methods at the remote end [88]. Inverse multiplexing for ATM has further been generalized to switching paths within an ATM switch referred to as switched connection inverse multiplexing for ATM (SCIMA) [89]. SCIMA instantiates an asynchronous cell ordering and re-assembly protocol together with a load-balancing cell dispatch algorithm.

2.3.2 Load Balanced Switching Architectures

Switches forward packets arriving from N input ports to M (usually $M = N$) output ports. To avoid collisions when packets from different input ports compete for the same output port, switching architectures rely on buffers and can be classified into input queued (IQ), output queued (OQ), and combined input/output queued (CIOQ) architectures accordingly.

When assessing switching performance, throughput and delay induced by the switching fabric are important measures. A switch is said to guarantee throughput θ if it will switch a fraction θ of the traffic for any input-output flow for any admissible arrival traffic, where arrival traffic is admissible if the rates of the traffic of all input ports destined for the individual output ports do not exceed the output port capacities [90].

OQ-switches are known to have the best performance in terms of QoS provisioning, but to avoid output contention, the output buffers must operate N times faster than the input line speed in an $N \times N$ switch. Thus, this architecture does not scale with increasing line rates.

IQ-switches overcome the buffer speedup problem, but simple implementations suffer from lower throughput and higher packet delay due to a phenomenon called head of the line (HOL) blocking. With HOL blocking, a packet in the front of a buffer blocks other packets destined to free output ports since its own destination is currently busy. The throughput can be limited to a value as low as 58.6% with simple FIFO queues [91]. Virtual output queueing (VOQ) and complex scheduling algorithms alleviate these problems [92], but with increasing line rates the time slot available for generating the scheduling decision decreases, which leads to scalability problems for the scheduling algorithms.

CIOQ-switches try to find a suitable tradeoff between OQ- and IQ-switches. The authors of [93] have shown that a speedup of two is necessary and sufficient to exactly emulate an OQ switch with any monotonic, work conserving service discipline on a CIOQ-switch.

In this context, [94] introduced load balanced Birkhoff-von Neumann switches

(cf. Figure 2.4). Birkhoff-von Neumann switches are single stage IQ-switches that were proven to achieve 100% throughput without internal speedup [95]. However, the original architecture has scalability problems due to its scheduling algorithm. The load balanced architecture is a two stage switch that introduces a load balancing stage in front of the original Birkhoff-von Neumann switch to completely replace the scheduler. The intuition behind it is that the first stage periodically connects each input to the VOQs of each intermediate input and thus transforms non-uniform traffic into uniform traffic that can be served by the sequence of periodic switching stages $\pi(t)$ in the second stage without a scheduler. [94] shows that under the assumption of weak mixing traffic, i.e. for almost all practical applications, a load balanced Birkhoff-von Neumann switch achieves 100% throughput. This assumption is only a problem for pathological periodic traffic patterns, but can be fixed as shown in [96].

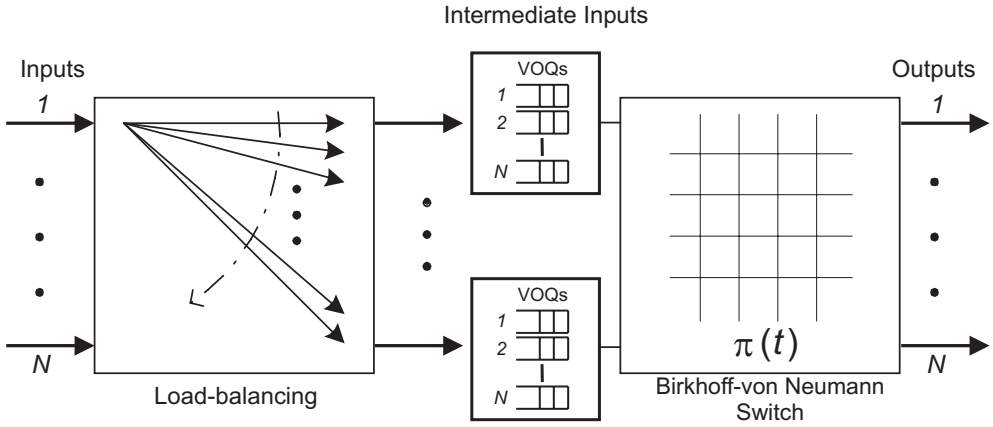


Figure 2.4: *The general architecture of a load balanced Birkhoff-von Neumann switch.*

Another problem of this architecture, typical for load balancing, is that packets can be mis-sequenced. Therefore, [97] proposed in a sequel to [94] to use the earliest deadline first (EDF) policy for the VOQs at the intermediate inputs together with resequencing buffers in a third stage to remedy this problem. The EDF policy bounds the amount of mis-sequencing such that the resequencing

buffers remain small. The authors of [98] follow another approach and use a policy called full frames first (FFF) that prevents mis-sequencing entirely without the need for an additional resequencing stage.

The load balanced Birkhoff-von Neumann switch is an interesting switching architecture since it is relatively easy to implement and scalable. [96] showed that it can be used to build a 100 Tb/s router where the switching fabric is implemented using passive optics. For this purpose, the load balanced architecture had to be extended in [99] for the case that not all ports, i.e. linecards, are present or working.

Finally, [90] gives a theoretical comparison of the full mesh interconnect used in load-balanced Birkhoff-von Neumann switches to alternative interconnects like a ring, a torus, and a hypercube. They find that the mesh interconnect is close to the optimal interconnect for loadbalancing in the sense that it achieves the highest throughput for a given capacity of the interconnect.

Since the purpose of the load balancing stage in a load balanced switch is the transformation of non-uniform traffic into uniform traffic over the input queues of the second stage, we conclude that the still very active topic of load balanced switches is only loosely related to load balancing for multipath forwarding.

2.3.3 Load Balancing for Parallel Network Processors on Highspeed Links

Network processors are special purpose hardware customized and optimized for packet processing. They execute functions such as pattern matching for address lookup, data bit field manipulation, and queue management.

Today, a single network processor alone is not able to serve highspeed links due to the large bandwidth of modern link technology. Parallel network processors are used to operate a highspeed link at full capacity as depicted in Figure 2.5. Thus, this is basically analogous to inverse multiplexing. The traffic is now distributed to different processing units instead of different links. The problem is related to our work because all packets of a flow should be forwarded to the

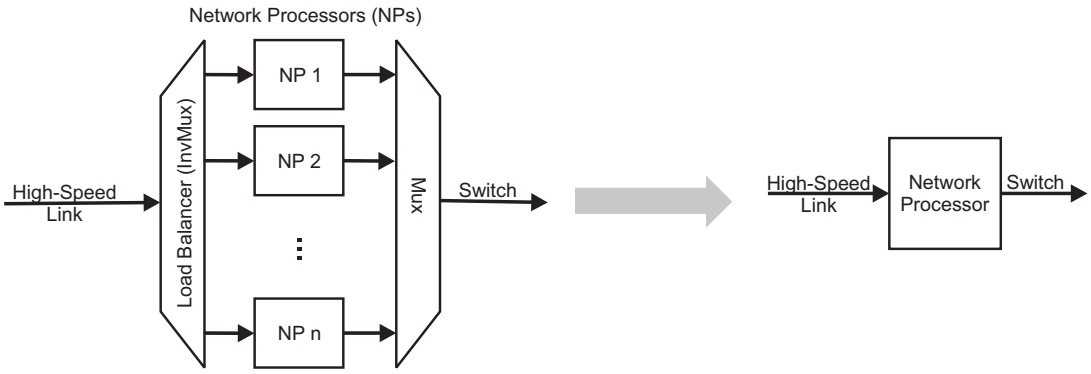


Figure 2.5: *Parallel load balanced network processors serve a single high-speed link. Packet reordering must be avoided to make the parallel processors appear like a single high-speed processor to the outside.*

same network processor to avoid packet reordering. Besides, scattering packets from the same flow to different network processors leaves copies of identical data in the processor caches. This impedes efficient caching and leads to unnecessary communication overhead due to continuous context updates between different forwarding engines [100]. Underloaded network processors lead to underutilized bandwidth and overloaded network processors lead to packet drops.

Like above, hash functions are suggested to map flows with the same hash value to so called flow bundles [101]. A lookup table entry directly assigns these flow bundles to the individual network processors. Unbalanced load is detected by monitoring the queue lengths of the network processors. If the buffer overflow probability of a network processor queue is high, flow bundles are reassigned. The time passed since the last packet arrival for a specific flow bundle determines whether the flow bundle may be reassigned to another network processor. If this time exceeds a specified timeout value, the reassignment is admissible. Timeout values larger than the packet forwarding latency through the network processor avoid packet reordering. This idea is similar to the idea behind load balancing for multipaths routing based on flowlet switching described in Section 2.2.2 where the burstiness of TCP traffic is exploited. This approach is also further applied for parallel network processors in [102]. A burst distributor assigns new packet

bursts to the currently least-loaded processor. It should be noted, however, that path latencies can be substantially longer than the latency of a network processor.

Moving only large flows reduces the number of flow reassignments that are required to achieve a balanced workload, it minimizes the packet reordering probability and the communication overhead for updating the network processor caches. Therefore, flows may be classified into high- and low-rate flows and the mapping for the few high-rate flows may be reassigned selectively [100, 103].

[104] proposes a scheduling algorithm for parallel network processors that is based on the "highest random weight (HRW)" algorithm (cf. Section 2.3.5). HRW was originally developed for systems like WWW caches to achieve minimal disruption in case of reconfigurations. The authors [104] modify it as an adaptive load balancing algorithm for parallel network processors.

[105] gives a performance analysis of the above mentioned schemes for parallel network processors and suggests a new approach built on the key ideas.

2.3.4 Multihoming

Enterprises may install multiple access links to achieve fault tolerance or to satisfy their bandwidth requirements. When the access links are subscribed from different Internet Service Providers (ISPs), this approach is called multihoming whereas multiconnecting or multi-attaching refers to obtaining simultaneous IP connectiviy from the same ISP [106] (cf. Figure 2.6). The term inverse multiplexing in contrast is only used for multiple parallel links.

Multihoming can be applied at different layers of the protocol stack such as the link, network, or transport layer [107]. Since we look at load balancing for multipath forwarding at the network layer, we limit this short description of multihoming to the network layer.

In the context of multihoming, the access links may be statically configured as primary and backup links for failover. In a more complex scenario, a multihoming load balancing system distributes the traffic over all access links. The load distribution preferably reacts to the current link load situation to turn the bundle

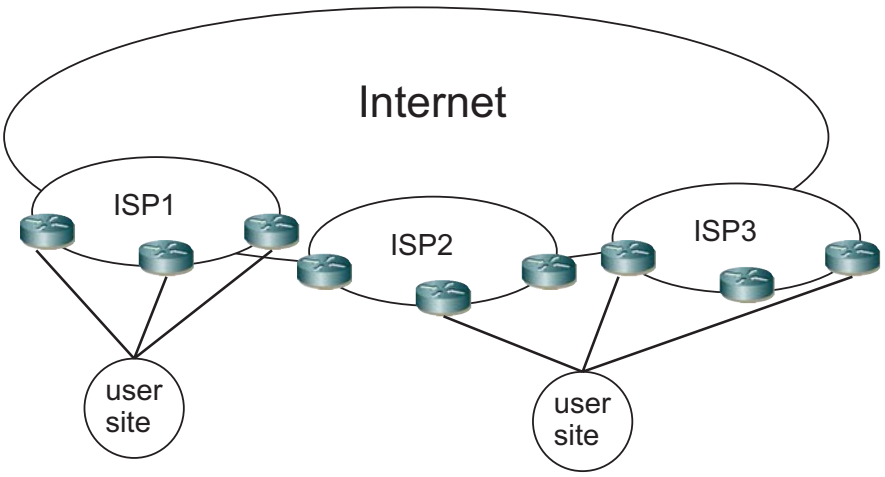


Figure 2.6: *Multi-attached and multihomed user sites.*

of different links into a reliable access connection with high performance.

There are basically two options to connect a load balanced multihomed user site. If the user site possesses its own range of IP addresses (provider-independent (PI) or provider-assigned (PA)) identified by a at least 24-bit address prefix and an autonomic system (AS) number, multihoming load balancing can be achieved through BGP peering [106, 108] using standard routing protocol functionality. However, the load distribution across multiple access links mainly depends on the static address assignment within the user site and on BGP routing policies. This results in inflexible load sharing and high configuration complexity.

If the user site does not possess an IP address range of its own, it receives different address blocks from each of its ISPs. Even though assigning multiple addresses to a single host is technically feasible, other techniques such as Network Address Translation (NAT) [109] should then be used to achieve multihoming load balancing [107]. The NAT-based approach is capable of balancing load at flow level granularity.

Since a NAT entity can control only the addresses and thereby the access links of the flows initiated from within the user site, additional load balancing techniques may be used for traffic initiated externally. This is mainly traffic destined

for servers hosted at the user site and therefore DNS-based techniques may be used.

The assignment of flows to the respective IP addresses representing the access links can be determined by a (static) hash function on the connection or session ID in a stateless fashion or by means of a stateful lookup-table [110]. These mechanisms are similar to load balancing for multipath Internet forwarding examined in this work. However, here it is impossible to dynamically reassign flows to other links since this also involves changing the corresponding IP address which destroys the session context of the transport protocols. Thus, while hash-based link assignment can be quickly reconfigured to react to link failures, only the lookup-table approach is capable of reacting also to the current load situation by assigning new flows to the currently best access link, which makes it the most flexible approach. Still, the authors of [110] do not find strong advantages of lookup-tables over hash assignment.

To summarize, only the methodology that assigns flows to links is in principle similar to load balancing for multipath Internet forwarding. In contrast, multihoming load balancing involves additional problems intrinsic to the IP address management. For instance, individual application protocols require multiple connections with the same IP address such as the control and data connection of an FTP transfer. This requires techniques to detect the connections within an application session [110]. Another issue is the assessment of the load situation on the access links and, beyond that, the e2e performance in terms of latency, throughput, and path availability [54, 111, 112]. Finally, multihoming raises scalability concerns for the current IP architecture (non-aggregation problem) leading to a significant increase of BGP routing table sizes [106], partly intensified due to the fact that multihomed networks have already surpassed singlehomed networks in number [113]. [107]gives a survey of multihoming technology for IPv4, [114] for IPv6.

2.3.5 Load Balancing for WWW Caches

WWW caches, also known as proxy caches, are used in a variety of ways. Among those, forward proxy caches and reverse proxy servers may be implemented on multiple load balanced machines.

Forward proxy caches are used in networks to reduce the number of outgoing WWW requests and, consequently, to reduce the outgoing traffic volume and the response time perceived by the users. In such a scenario, http requests of a browser are first forwarded to a proxy that looks for the desired content in a local cache. If the content is locally available, the request is served from the cache, otherwise it is forwarded to the web server providing the requested resource.

A reverse proxy is a proxy server that is typically used in front of web servers. It caches static content or often requested dynamic content with limited but still valid lifetime to offload the central web servers and reduce the response time perceived by the users. Only if the content is not available on the proxy server or outdated, the request is forwarded to the actual web servers. Besides, reverse proxy servers often handle encryption and compression tasks, act as load balancers for the web servers, and are popular for security reasons since they provide an additional layer of defense.

If forward or reverse caches are distributed over several machines for scalability reasons, special load balancing techniques are necessary. To avoid asking every cache individually for the requested content, the proxy hashes the request string to a value that points to the cache which is responsible for the request. The focus of this kind of load balancing is not primarily an even distribution of the load. It is intended to reduce the search time and to increase the hit rate of the caches since each request item is stored only once. In addition the disruption should be kept low in case one of the caches fails or additional hardware is added.

The latter can be done in an elegant way by the “highest random weight (HRW)” algorithm [115]. Here, a random weight is calculated for each cache by a hash function based on the request string and the cache ID. The cache with the highest random weight is responsible for the request. If a cache fails, the re-

quest points automatically to the cache with the next highest weight. Thus, the entries on the still working caches remain untouched. If an additional cache is added, the responsibility for most entries remains with the original caches, only those entries where the new cache has the highest weight are moved. Hence, all caches are offloaded and move part of their responsibilities to the new hardware. This approach has been extended to heterogeneous server systems in [116]. The idea is to assign multipliers to the individual servers according to their capacity to scale the return values of HRW. A recursive algorithm calculates the multipliers such that the object requests are divided among the servers according to a pre-defined fraction list.

2.3.6 Other Load Balancing Applications

Next to the load balancing applications from above that are directly correlated to the packet forwarding process and the example of minimum disruption load balancing mechanisms for WWW caches, there are various other load balancing applications. Load balancing is required for server clusters and web server farms for scalability and performance [117–121], in wireless ad-hoc networks for efficient energy usage [122–124], in peer-to-peer (p2p) systems to equally distribute allocated resources and responsibilities among the individual peers [125–128], and in mobile cellular systems for a high spectrum efficiency [129, 130].

Since these applications are only loosely related to the subject of load balancing for multipath forwarding in this work, we do not present their basic mechanisms here.

Finally, load balancing can also be seen as distributing the load evenly over one instance, for example, a network. This is the goal of traffic engineering. Load balancing for multipath Internet forwarding is one means to achieve this.

2.4 Fast Resilience Concepts

Given the growing size and complexity of modern communication networks, the presence of component failures is a fact of their daily operation [131] and requires special precautions. For this purpose, resilience mechanisms maintain connectivity in case of outages where possible, and the network resource management must provide sufficient capacity resources to transport the protected traffic through the network also during failure cases without service degradation.

Resilience mechanisms can be divided into two schemes, restoration and protection. Restoration sets up a new path after a failure while protection switching pre-establishes backup paths in advance. Due to their different design principles, restoration is slow while protection schemes react much faster. IP re-routing is the most widely used restoration mechanism. Careful tuning of timeout parameters reduces its recovery time to values in the order of one second [132–134], but this time cannot be reduced arbitrarily without jeopardizing the network stability [134].

Networks usually have a layered structure. For instance, an IP network may consist of an IP layer operating directly above a dense wavelength-division multiplexing (DWDM) optical infrastructure with SONET framing. There also might be an MPLS layer. Resilience mechanisms are applied at all layers to protect against different failure sources. Protection switching directly at the optical layer provides the fastest recovery times, but it cannot protect against the failure of an IP or MPLS node.

Currently, we see new emerging services such as Voice over IP (VoIP), virtual private networks (VPNs) for finance, and other real-time business applications. They require stringent service availability and reliability and, thus, a fast reaction to failures [135]. Further, the majority of link failures in a network are short-lived failures, 50% last less than a minute [131, 136]. This calls for resilience strategies that repair the failure locally. They react fast and can suppress network-wide failure notification for short-lived failures to avoid the involved problems during routing changes.

The demand for fast and local failure reaction led to the development of fast reroute (FRR) techniques. These techniques prepare alternate paths at each intermediate node of a path that are immediately available in case of a failure. For MPLS technology, two different FRR approaches have already been standardized [137]. However, pure IP networks also need fast resilience. Therefore, current IETF drafts and other publications propose various methods for IP-FRR [6, 138–142]. IP-FRR is especially attractive for many network providers since it relies on simple and plain IP routing.

In the following sections, we describe the basic design principles of FRR mechanisms for both MPLS and IP networks. For IP-FRR we present a comprehensive resilience framework that covers all steps from routing in the failure-free case to routing in the failure topology. It incorporates fast local reactions to failures and subsequent ordered re-convergence to the failure topology.

In Chapter 4 we discuss the layout of the backup paths for MPLS-FRR and suggest simple heuristics that already yield a significant reduction of the required backup capacity. For IP-FRR we study the combination of two IP-FRR mechanisms loop-free alternates (LFAs) [139] and not-via addresses [140] suggested by the IETF and the benefits thereof.

2.4.1 Failure Causes

In communication networks there are various reasons for failures like maintenance, accidental fiber cuts, and misconfigurations. An overview and characterization of network failures is given in [143, 144]. Failures can be categorized into planned and unplanned outages. Planned outages are intentional, e.g. maintenance operations, and operators can take appropriate precaution in advance. Unplanned outages are a serious risk for network operators since they may lead to disastrous network conditions. Unplanned outages can be further subdivided into failures with internal causes (e.g. software bugs, component defects, etc.) and those with external causes (e.g. digging works, natural disasters, terror attacks, etc.).

The authors of [131, 136] analyze and characterize failures in an operational IP backbone based on the Sprint IP network. [131] examines the frequency and duration of link failure events and shows that they are inevitable during the daily operation. Only 10% of failures last longer than 20 minutes. About 40% last between one minute and 20 minutes. In particular, 50% of all failures are short-lived and last less than a minute. Based on the work in [131], [136] further characterizes the causes for failures in an IP backbone. The results show that 20% of all failures are due to planned maintenance. Among the unplanned failures, almost 30% can be attributed to router-related and optical equipment-related problems, while the remaining 70% affect only a single link at a time.

These results confirm that resilience concepts on the network layer should protect against all single link and single node failures and possibly shared risk groups (SRGs). This protection is sufficient in most cases [145]. In addition, the protection against uncorrelated multi-failures remains a desirable feature. In any case, the possible performance degradation during multi-failures should be as low as possible. At the same time, the large fraction of short-lived failures leads to a growing demand for failure resilient routing protocols that ensure high service availability and reliability despite transient link failures. We further elaborate on this for IP-FRR later in this work.

2.4.2 Classification of Resilience Mechanisms

We give a brief overview on resilience mechanisms so as to classify the FRR concepts under study. A broader and more complete overview can be found, for instance, in [143, 144]. In case of a network failure, resilience mechanisms redirect the affected traffic around the failure location. They can be distinguished into restoration and protection switching mechanisms. Protection switching establishes backup paths in advance while restoration finds a new path only if a failure occurs. Therefore, protection switching reacts faster than restoration.

IP Restoration

Restoration is typically applied by IP re-routing. IP networks have the self-healing property, i.e., their routing re-converges after a network failure by exchanging link state advertisements (LSAs) such that all but the failed nodes can be reached after a while if a physical connection still exists. This is robust [145, 146], but slow: There have been several proposals [132–134] that accelerate the convergence of link state routing protocols by reducing the interval length for exchanging the LSA updates. However, timers cannot be reduced arbitrarily and the reductions run the risk of introducing instability in the network, in particular in the face of frequent transient link failures [134]. Besides, the computation of the shortest paths that are needed to construct the routing tables based on the new LSAs requires a substantial amount of time. Another example for restoration besides IP routing are backup paths in MPLS that are set up after a network failure.

Protection Switching Mechanisms

Protection switching addresses the problem of the slow re-convergence speed of restoration mechanisms. It is often implemented by MPLS technology due to its ability to pre-establish explicitly routed backup paths. Depending on the place where the reaction to failures is executed, protection switching mechanisms can be distinguished into end-to-end (e2e) and local protection.

End-to-End Protection Switching In case of e2e protection switching, the reaction to a failure along a path is executed at the path ingress router, i.e., at the first router of a path. Backup paths are set up simultaneously with the primary paths. The classical e2e protection scheme establishes one backup path for each primary path. In case of a failure, the path ingress router of a broken primary path simply shifts the traffic to the corresponding backup path. More sophisticated mechanisms use several backup paths per primary path to achieve a better traffic distribution in the network during failures.

The self-protecting multipath (SPM) [9, 147], e.g., consists of disjoint label switched paths (LSPs) between path ingress and path egress routers. It does not distinguish between primary and backup paths, but distributes the traffic over all paths of the multipath according to a traffic distribution function (see Figure 2.7). If one of the paths fails, the traffic is carried over the still working paths according to another precomputed traffic distribution function. Thus, traffic distribution functions can be optimized a priori to minimize the required backup capacity in the network [18]. The SPM reduces the required backup capacity relative to the classical e2e protection scheme significantly and is sufficiently robust to multi-failures [14]. The traffic distribution over multiple paths introduces additional problems due to the often unpredictable accuracy and dynamics of load balancing algorithms (cf. Chapter 3). Hence, [17] suggests the integer SPM (iSPM) and simplifies the traffic distribution function to a path-failure specific path selection function without sacrificing much of the efficiency in terms of capacity requirements.

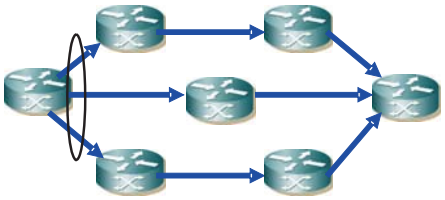


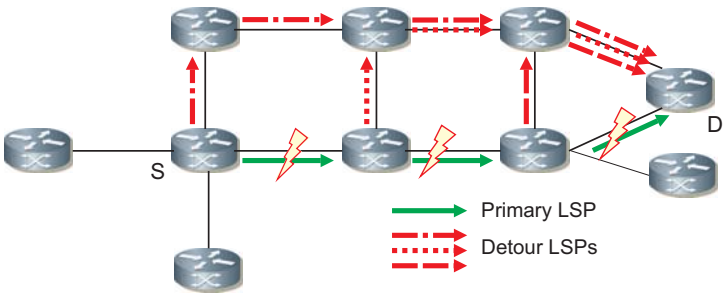
Figure 2.7: *The SPM performs load balancing over disjoint paths according to a traffic distribution function which depends on the working paths.*

E2e protection switching is faster than restoration methods, but the signaling of a failure to the path ingress router takes time within which traffic is lost.

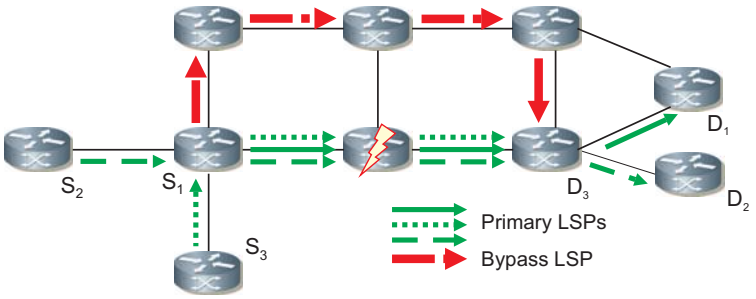
Local Protection Switching Local protection schemes tackle the problem of lost traffic during failure signaling from the outage location to the ingress router that is necessary for e2e protection. Backup paths towards the destination are set up not only at the ingress router of the primary path but at almost every node of the path. Then, a backup path is immediately available if the path

breaks at some location. Both MPLS- and IP-FRR are local protection switching mechanisms since they pre-establish local backup paths. A broad overview about different MPLS- and IP-FRR schemes can be found in [148, 149].

2.4.3 MPLS Fast Reroute



(a) One-to-one backup.



(b) Facility backup.

Figure 2.8: *Illustration of the one-to-one and facility backup options for MPLS-FRR.*

The operations for MPLS-FRR mechanisms have been standardized by the IETF in [137, 150–152]. Commercial router vendors already support these mechanisms in their MPLS-capable routers [153, 154]. In case of a network element failure, MPLS-FRR deviates the traffic at the router closest to the failure location (cf. Figure 2.8), the point of local repair (PLR). The standard describes two basically different options: one-to-one and facility backup. The one-to-one backup

deviates the traffic directly from the outage location to its destination (cf. Figure 2.8(a)) while the facility backup bypasses the traffic around the outage location (cf. Figure 2.8(b)) to repair the original primary path. The facility backup concept deviates several label switched paths (LSPs) over a single bypass around the failure location while the one-to-one concept needs a separate backup path for each LSP. Thus, the facility backup leads to a lower configuration overhead, but it introduces other configuration problems.

The protection of MPLS-FRR has been extended in [155] to point-to-multipoint (p2mp) LSPs suitable for multicast traffic. The facility backup option for p2mp LSPs uses point-to-point (p2p) bypass tunnels for the protection against the failure of the next hop. If the failed next hop has n children as shown in Figure 2.9, this requires the use of n p2p bypass tunnels and leads to n times the traffic on some links. This obviously wastes capacity resources. A current IETF Internet draft [156] therefore suggest the use of p2mp bypass tunnels instead. However, in this work we focus on MPLS-FRR for unicast traffic only.

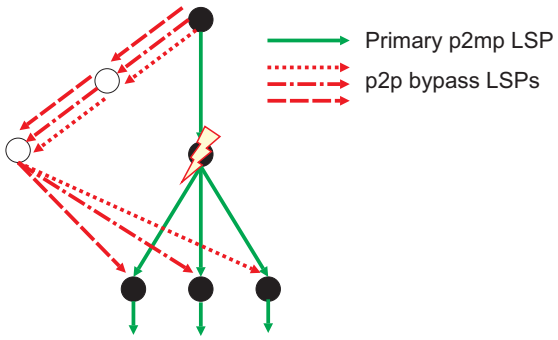


Figure 2.9: *The use of p2p bypass tunnels for the repair of node failures of p2mp LSPs possibly leads to the multiplication of traffic and therefore increased resource requirements.*

Path Layout Algorithms

The MPLS-FRR standards provide only the protocol mechanisms for the implementation of a detour or a bypass, but the path layout is not determined. Thus,

operators have many degrees of freedom for the set up of backup paths. Usually, the default layout for the backup paths follows the shortest paths that avoid the outage location [143].

Besides, there are different approaches that try to find an optimized path layout. We shortly describe their main aspects in the following. There are also various algorithms that deal with optimal path layout for local protection in optical and ATM networks. These solutions can be partly translated to the examined problem. However, since we focus on simple heuristics based on shortest paths algorithms in the remainder of this work, we restrict the following short discussion to work that is directly related to MPLS.

The various path layout schemes can be categorized into online and offline algorithms. Online algorithms are designed for the case when LSP setup requests arrive one-by-one with no a priori knowledge of future arrivals and, thus, LSPs are set up and torn down on demand. Offline algorithms configure a fixed set of LSPs for planned end-to-end demands, e.g., a fully meshed LSP overlay.

Online Algorithms The authors of [157] suggest a mixed integer linear program (MILP) formulation to find optimum backup paths for the one-to-one backup mechanism. However, the solution of MILPs is complex and it may be difficult and very time consuming for medium-size or large networks.

Besides, using an MILP requires a central server for the computations and the result is not very robust since the global optimum is very sensitive to minor changes in the set of requests [158]. The authors of [158] suggest a heuristic algorithm based on Dijkstra's shortest path algorithm that may be implemented both in a centralized and a distributed manner. The scheme is designed to improve the resource sharing between backup paths protecting against independent failures. When a new LSP request arrives, the primary path is reserved along the shortest path subject to some metric such as delay constraints. Afterwards, bypasses that protect against node failures are computed for one node of the primary LSP after the other, beginning from the egress node of the primary LSP. To compute such a bypass, Dijkstra's algorithm is run on the network with spe-

cific link weights. Links that cannot be used due to the considered failure receive infinite weight, links where additional capacity must be reserved for the bypass receive a weight equivalent to the required capacity, and links where no additional capacity is necessary due to capacity sharing receive a very small weight. This shortest path computation prefers paths that do not increase the additional bandwidth. For implementation in a distributed manner, some aggregated information about reserved, used, and available bandwidth on the links in failure cases must be signaled through the network. Even though this fast reroute approach uses bypasses, it is different from the facility backup option specified in the MPLS-FRR standard since it applies LSP-specific bypasses. The algorithm is based on the algorithms in [159, 160].

The authors of [161] present another distributed online algorithm for the one-to-one backup path layout. It aims at optimized backup capacity sharing depending on the current network state.

Offline Algorithms The offline problem to find a suitable path layout for given end-to-end demands can be considered for network configuration and network dimensioning. For network configuration, the link capacities are given and the task is to divide the link capacities into working and backup capacities such that the working capacities fit the given demands as good as possible while the backup capacities are minimized. For network dimensioning, the necessary link capacities are still to be assigned and the primary and backup paths must be routed through the network such that the resulting required overall capacity is minimized.

The problem to guarantee fast restoration in a network with given link capacities while minimizing the amount of network capacity dedicated to protection is NP-complete as shown in [162]. The authors present two 2-approximation algorithms to solve this problem. However, the task solved here is not the accommodation of given traffic demands in the network. Instead, the given link capacities are divided into working and backup capacities such that there is a bypass for each link and this bypass has sufficient backup capacity to carry the working ca-

pacity of the link in case of a failure. The goal is the minimization of the sum of backup capacities. Consequently, the problem solution does not consider the amount of working capacities required for given traffic demands. The resulting working capacities might not be adequate on all links to accommodate given traffic demands even though there is a solution for the given demands that requires only slightly more backup capacity.

While there are related theoretical results for the complexity of the offline problem to find a suitable path layout for planned end-to-end demands, simple mechanisms that determine such a resilient path layout are still required. We study suitable mechanisms for network dimensioning in Chapter 4.

Related Algorithms Besides these approaches to obtain a suitable path layout for the backup paths in single failure cases, [163] extends MPLS-FRR to deal with multi-failure scenarios and [164] presents an FRR algorithm for p2mp protection.

[165] suggests a segment-based protection scheme, i.e., not every single link and node failure is protected by a backup LSP, but only overlapping segments consisting of a small number of hops. Similarly [166] describes a family of algorithms that allow for backtracking. The failure detecting node sends traffic back for a maximum number of D hops until it reaches another node that has a local repair path installed.

2.4.4 Our contribution towards MPLS-FRR

Only little has been written in the MPLS-FRR literature about offline algorithms for the planning of a resilient path layout for given end-to-end demands. In particular, a systematic analysis of the standard path layout for both MPLS-FRR options that considers important network characteristics and outage scenarios is still missing. Such an analysis is important for the understanding of the tradeoffs between the different options and also for comparison to other resilience mechanisms. In Chapter 4, we discuss the standard path layout for the one-to-one and

facility backup options. Based on a network dimensioning approach, we analyze the required capacity for the standard MPLS-FRR one-to-one and facility backup concepts considering various networks and outage scenarios. The insights gained lead to the suggestion of a simple enhancement of the path layout that efficiently reduces the backup capacity requirements. Finally, we compare the backup capacity requirements to other protection methods.

2.4.5 IP Fast Reroute

IP routing as implemented by the most widely used interior gateway protocols (IGPs) OSPF [48] and IS-IS [49, 50] relies on the exchange of link state advertisements (LSAs). Naturally, this imposes limitations on the convergence time in case of a network element failure [134]. Hence, additional mechanisms are required to provide a faster reaction. These mechanisms are referred to as IP-FRR techniques.

IP-FRR is still under development and discussion in the research community. In this section, we briefly describe the main ideas behind several competing mechanisms. [138] divides IP-FRR repair paths into three basic categories with increasing complexity and failure coverage: equal-cost multipaths (ECMPs), loop-free alternates (LFAs), and multi-hop repair paths.

ECMPs are paths of equal length according to the link cost metric. If they exist between point of local repair (PLR) and destination, one or more of the equal-cost paths that do not traverse the failed element can be trivially used to repair the outage. LFAs [139] are direct neighbors of the PLR that still have a shortest path towards the destination. In contrast to ECMPs, LFAs are longer than the shortest path from the PLR to the destination and they must fulfill certain requirements to avoid routing loops. Multi-hop repair paths, finally, are paths from routers to the destination that are more than one hop away from the PLR. Special mechanisms are necessary to force the packets on their way to the router offering the multi-hop repair path before they can be forwarded to the destination using normal IP routing. Multi-hop repair paths can be further classified and sub-divided accord-

ing to mechanisms used to precompute several alternate forwarding information bases (FIBs) such as failure inferencing based re-routing (FIFR) [142, 167] or multiple routing configurations (MRC) [168], mechanisms equivalent to a loose source route such as tunnels [169], and mechanisms that require special addresses such as not-via addresses [140] that instruct all routers not to use certain network elements.

In general, IP-FRR mechanisms should cover most failures, e.g., all single link or node failures, and should not create problems, e.g., unpredictable severe routing loops, in case of unanticipated multi-failures.

Mechanisms for IP Fast Reroute

In the following we limit our description to LFAs, not-via addresses, FIFR, and MRC. LFAs and not-via addresses are the most favored mechanisms within the IETF routing working group (RTGWG). LFAs are simple, but they cannot cover all single link and node failures. Not-via addresses cover all single failures, but they are more complex and require IP tunnels. FIFR and MRC are two mechanisms proposed outside the IETF.

Loop-free alternates A loop free alternate (LFA) is a local alternative path from a node S towards a destination D in the event of a failure [139]. If S cannot reach its primary next hop P towards D anymore, it simply sends the traffic to another neighbor N that still can forward the traffic to D avoiding both the failed element and S and thus does not create routing loops. Figure 2.10 gives an example. LFAs are pre-computed and installed in the FIB of a router for each destination. An Internet draft [139] specifies criteria for LFAs with different properties. A detailed description of LFAs including a classification of LFAs with respect to their ability is given in Section 4.3.1.

Not-via addresses The intention of not-via addresses [140] is to protect the failure of a node P or of its adjacent links by deviating the affected traffic

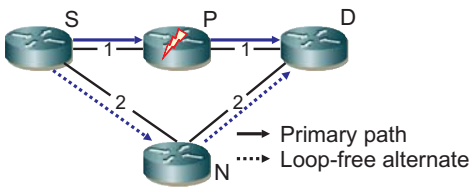


Figure 2.10: *In case node P fails, neighbor N provides a loop-free alternate towards D.*

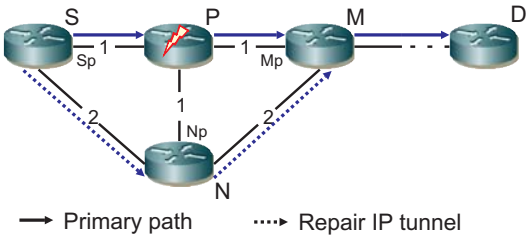


Figure 2.11: *Not-via addresses use IP-in-IP tunnels. Packets with special address Mp are carried to M not via the failed node P.*

around P to the next-next-hop (NNHOP) M using IP-in-IP tunneling. This is not achievable with normal IP forwarding and requires a special "not-via address" Mp. Figure 2.11 gives an example. In case of the failure of node P, S encapsulates the traffic destined to D into packets towards Mp. All routers in the network know this special address and forward the packets on their shortest path to M that does not contain P. M decapsulates the packet and sends it to D using normal IP routing. This approach is similar to MPLS-FRR facility backup (cf. Section 4.1). A more detailed description of not-via addresses is given in Section 4.3.2.

Failure Inferencing based Fast Rerouting Failure inferencing based fast re-routing (FIFR) exploits the fact that packets arrive at routers through other interfaces during network element failures if re-routing is applied. It computes interface-specific forwarding tables where the next hop of a packet does not only depend on its destination but also on the incoming interface. It has been proposed to handle transient link [142, 170, 171] and node [167] failures.

Figure 2.12 demonstrates the basic idea of FIFR. Node S usually forwards packets destined for D via P. Due to the link weight settings, node C sends all packets destined for D to node S. Hence, in the failure-free scenario it never sees a packet destined for D arriving from interface $S - C$. In case node P fails, S forwards the packets for D to C instead and C infers from this unusual behavior

that a failure occurred along the regular path and now forwards the packet over interface $C - E$.

FIFR achieves 100% failure coverage for single link and node failures [167]. However, when node protection is applied, the protection of the last hop towards a destination causes problems for the inferencing mechanism for the following reason. If the router directly preceeding the destination cannot reach the destination anymore, it is reasonable to assume link failure. Since it is impossible for the routers along the infered repair path to differentiate in general which router actually deviated the traffic and, hence, whether the deviating router assumed link or node failure, an additional mechanism like tunneling is required.

The original mechanism had problems with asymmetric link weights, which have been fixed in [172]. There, extensions to handle inter-AS failures have also been developed. When multiple failures occur and during the re-convergence to the new topology after a failure, FIFR still has difficulties. Then, major instabilities resulting in routing loops may occur caused by the FRR mechanism. [173] suggested a modification called blacklist-based interface-specific forwarding (BISF) that avoids routing loops also in case of multiple failures. Even though it can repair a subset of double link and double node failures, its coverage for single node failures already drops below 100%.

Multiple Routing Configurations Multiple routing configurations (MRC) described in [141] and as a similar concept in [6, 174] are a small set of backup routing configurations for use in failure cases to provide local recovery. The routing configurations complement each other in the sense that at least one valid route remains in a single link or node failure scenario for each pair of nodes in at least one configuration.

The normal configuration C_0 consists of the basic network topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ – where $\mathcal{V}$ is the set of n nodes and $\mathcal{E}$ is the set of m links in the network – and an assignment of finite link weights $w_0(l) \in \{1, \ldots, w_{max}\}$ for all links $l \in \mathcal{E}$. In the failure-free scenario, normal IP routing is performed on configuration C_0. The backup configurations C_p all have a different and individual

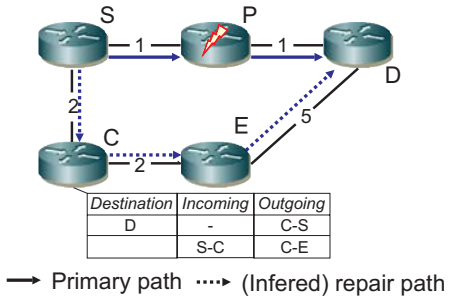


Figure 2.12: *If a packet for D arrives via $S - C$, node C infers a failure along the regular path and forwards via $C - E$ instead.*

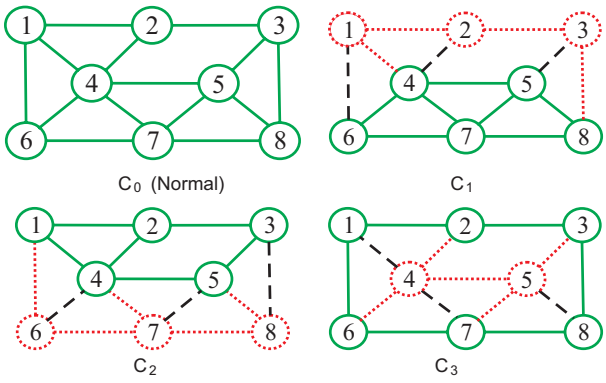


Figure 2.13: *Example network topology with backup configurations for MRC. In the normal configuration, all links have finite weights. Isolated nodes and links are depicted dotted, while restricted links are dashed.*

weight set. A link l is called isolated in a configuration if $w_p(l) = \infty$ and restricted if $w_p(l) = |\mathcal{E}| \cdot w_{max}$. A node v with restricted or isolated adjacent links only – among them at least one restricted link – is called isolated.

The intuition behind this terminology is that isolated links and nodes never carry traffic in a configuration. Besides, due to their high but finite weight, restricted links are only used to access isolated nodes in case of link failures. Each link and node must be isolated in at least one backup configuration C_p. During failure events this guarantees that a valid route exists for each node pair in the corresponding configuration C_p that does not use the failed element. Figure 2.13 shows an example topology with three backup configurations.

The router detecting a network element failure, the PLR, locally selects an appropriate backup configuration C_p and marks the packets accordingly. The routers in the network know all backup configurations and forward incoming packets in the corresponding configuration. The MRC concept can be implemented using the multi-topology extensions for OSPF and IS-IS [175–177].

MRC achieves 100% failure coverage for single link and router failures and

does not create routing loops in case of unanticipated failures since the packets may not be switched from one backup topology into another. [178] proposes an extension 2DMRC for handling two concurrent failures. [22] presented a new, enhanced MRC scheme called relaxed MRC (rMRC) that simplifies the topology construction and increases the routing flexibility in each topology while retaining the failure coverage.

IP-FRR and Loop-Free Convergence

According to [131, 136], most failures are short-lived and last less than a minute. Hence, for those failures, IP-FRR local failure recovery can suppress network-wide failure notification that triggers global convergence to the failure topology. This avoids unnecessary problems involved during convergence to the failure topology and back. In case of long-lived failures, a predetermined timer indicates that the current outage is likely to last longer and network-wide failure notification and convergence to the failure topology is unavoidable. This is particularly necessary to survive additional failures during the outage period leading to unplanned multi-failures. Most failures are single failures, but with increasing outage duration double failures become more likely.

Since the IP-FRR mechanism in place protects the traffic hit by the current failure, there is enough time for loop-free convergence. Mechanisms for loop-free convergence guide the convergence process, e.g., by means of a given order in which routers are allowed to apply their updated FIBs. Routing loops have a detrimental effect on network performance. They impair the traffic for looping packets, but also for other packets that encounter increased link utilizations due to looping packets [179]. The authors of [180] suggest a framework for loop-free convergence. Among their suggestions are ordered FIB updates [181] initially proposed in [182]. Routers closer to the outage location must revise their routes before routers further away. The order is based on the router's position in the reverse spanning tree and timer values. Once the network converged to the failure topology, the traffic follows the converged routing. After failure recovery, a loop-

free convergence algorithm transforms the routing back into its original state.

This leads to the proposal of the comprehensive resilience framework depicted in Figure 2.14. It consists of five states:

I Routing in the failure-free case.

II Fast local reaction to failures until failure recovery for short-lived failures or until timer expiration classifying the failure as medium- to long-lived.

III Loop-free convergence to failure topology.

IV Routing in failure topology until failure recovery.

V Loop-free convergence to failure-free topology and routing in the failure-free case thereafter (I).

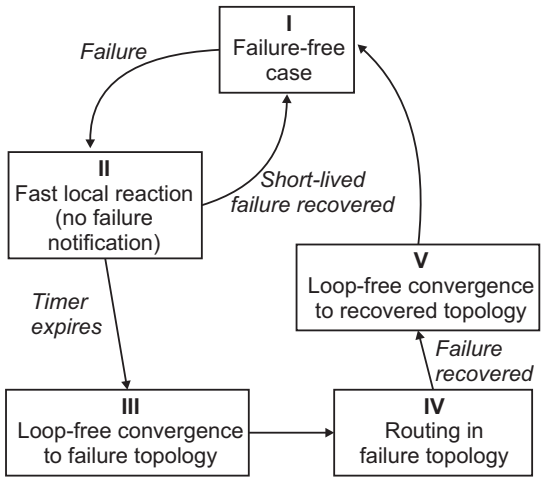


Figure 2.14: *Resilience framework consisting of five states describing the failure recovery cycle.*

We refer to these five states as the failure recovery cycle. From a network provisioning perspective, the network must be sufficiently dimensioned to accommodate the protected traffic during all states for all considered failure scenarios. This framework raises several interesting questions. What is the prize for short time failure coverage through IP-FRR (states I and II)? What is the prize for medium- and long-lived failure coverage (states I and IV)? What is the prize for full cover-

age (all states)? In this work our analysis of IP-FRR mechanisms contributes to the first question.

Distinguishing the appropriate reaction to short-lived and long-lived failures is also beneficial for exterior gateway protocol (EGP) routing. When the egress point selection from a set of possible egress points is decided by comparing IGP costs, it is commonly called hot potato routing. The border gateway protocol (BGP) selects the intra-domain route associated with the closest egress point based upon intra-domain path costs. Consequently, intra-domain routing changes can impact inter-domain routing and cause abrupt changes of external routes with detrimental effects [183, 184].

Supplementary Literature

Proposals for mitigating the impact of link failures on network performance were presented in [16, 145, 146, 185]. These approaches are based on IP link weight optimizations. They find appropriate link weight settings such that overload on links is reduced for a set of considered failure scenarios. Thus, they prepare the network for failures in terms of link load but they are not concerned with increasing availability such as IP-FRR. They still rely on IP re-convergence. However, both approaches can and should be combined to achieve cost-efficient resilience.

The idea to use ECMPs for local repair is very common. The authors of [186] proposed to introduce a limited number of MPLS tunnels in the network such that there are always two equal-cost paths towards a destination at every node for fast reaction. Highly meshed backbone networks often have multiple equal-length shortest paths between every pair of nodes [187].

[188] developed a method for routing loop detection based on the work in [179]. They associate BGP and IS-IS routing events with loops detected in traffic traces of the Sprint backbone network. Even though there were network element failures, none of the detected routing loops was correlated with such a failure event. However, this is due to the extensive use of ECMPs in this specific network.

In the context of inter-domain routing, [133] developed a method that achieves fast recovery of BGP peering link failures.

2.4.6 Our contribution towards IP-FRR

IP-FRR is a new resilience concept that is currently under development and still has many open research issues. Our work deals with the combination of the relatively simple LFAs and the relatively complex not-via addresses to obtain further insights whether such a combination is beneficial. For this purpose, the contribution of our work in Chapter 4 is twofold. Firstly, we provide a classification of different LFAs with respect to their ability and establish a new taxonomy. This yields suggestions for their combination with not-via addresses regarding different resilience requirements. Secondly, we study the effect of combining both mechanisms to achieve 100% coverage. We discuss pros and cons of both mechanisms and analyze their applicability for different resilience requirements following our suggestions for their combination. We also study the backup path lengths and the amount of tunnelled traffic.

An analysis of the coverage of IP-FRR mechanisms can be found in [189–191]. So far only average values over all nodes in the network [189, 190] or cumulative distribution functions for the number of alternate nodes offering a specific repair mechanism [191] were given. [190] presented graphs for backup path lengths, but not with respect to the combined usage of LFAs and not-vias. Hence, the detailed analysis of the applicability of LFAs and not-vias presented for the individual nodes and in particular the combination options for different resilience requirements and the analysis thereof in Chapter 4 is novel. The analysis of the performance measure amount of decapsulated traffic has not been done before either.

2.5 Dimensioning of Resilient Networks

Network dimensioning is the task of providing a network with sufficient resources. Link capacities sufficient for the traffic in the network are crucial to fulfill the QoS level promised to customers in service level agreements (SLAs). If the traffic currently in the network depletes the network capacity, congestion arises and leads to service degradation. Therefore, network operators rely on estimates of the traffic expected in their network and translate them into capacity requirements for the provisioning of their networks.

Most CO studies use both a flow and a packet level model. The first models the number of active flows of a traffic aggregate in the network whereas the second produces the required extra bandwidth above the mean data rate of the traffic.

Traffic demands are usually given in a point-to-point traffic matrix that contains the traffic between any origin-destination pair in the network. Traffic matrices fluctuate over time in a 24 h and 7 day period and the busy hour traffic matrix is required for the purpose of network dimensioning. Traffic matrix estimation is a difficult problem since the information available from network measurements is usually limited [192–199].

Provided the traffic demand matrix is known, traffic models are used to calculate the link capacities required in the network to guarantee a given quality level. The concept of effective bandwidths [200], for instance, translates statistical traffic characteristics into the required bandwidths, i.e., the effective bandwidths, to meet particular QoS targets. It has been studied for various traffic models [201–205].

Traffic can be modeled on two different levels, on the flow and on the packet level. The flow level describes the arrival rate of new requests to the network, their duration, and the distribution of the average flow rates. The packet level describes the bit rate of the flows or of a traffic aggregate. This may comprise the packet arrival rate within a flow or an aggregate of flows and the packet size distribution. Hence, the flow level characterizes the number of flows in the network and the mean data rate of the traffic, whereas the packet level models the extra bandwidth

above the mean data rate that is necessary to safely accommodate the traffic in the network. Efficient network dimensioning requires accurate traffic models and traffic models may be significantly different for individual applications.

Network element failures and unexpected user behavior complicate the task of network dimensioning. Resilience mechanisms react to failures and deviate the traffic around the outage location. This leads to increased capacity requirements in other parts of the network that are higher than during failure-free operation. Resilience mechanisms maintain the mere connectivity, the resource management must provision sufficient capacity. Unexpected user behavior may be due to singular events. For a short period, a single spot in the network, a so called hot spot, may attract more traffic than usual since it currently offers extremely popular content. This skews the traffic matrix, may lead to overload in the network, and also must be taken care of.

It is impossible to foresee all contingencies. Mechanisms that avoid overload in the network and thus enforce QoS must be applied. There are two main approaches for this objective: capacity overprovisioning (CO) and admission control (AC) [206]. CO adds a security margin to the expected traffic demand under normal conditions and provides so much capacity on the links that overload is unlikely while AC limits the number of flows over a specific network element, e.g. a link, to avoid overload. CO is applied in today's core networks and it is currently favored by many Internet providers (ISPs) and researchers as the preferred mechanism [207]. Reports from various tier-1 ISPs suggest that IP backbone networks are usually over-provisioned to the point where the utilization of backbone links is less than 50% of their total capacity [208]. Other analyses see even less utilization in data networks [209] and do not anticipate this fact to change.

It is generally perceived that CO keeps the networks simple while AC is complex and requires a significant amount of interoperability. In return, AC is often presumed to be more suitable for guaranteed QoS which can be difficult and costly with CO. As a consequence, the discussion between both parties regarding the question which approach should be taken in an quality-enabled Internet resembles an almost religious war [210, 211].

In this context, in Chapter 5 we present a capacity dimensioning method for networks with resilience requirements and changing traffic matrices, investigate the impact of various sources of overload on the required capacity for CO in networks with and without resilience requirements, and compare this required capacity with the one for AC.

In the following sections, we briefly describe the fundamentals of resilient networks, CO, and AC. We further review existing literature on CO, AC, and the comparison of both. Finally we present capacity dimensioning approaches for AC and CO on a single link.

2.5.1 Sources of Overload in Networks

If the amount of traffic in the network is too large, this leads to overflow of the transmission buffers of individual links and, thus, to congestion. Hence, overload must be avoided. There are various sources of overload in a network. Overload may be caused, e.g., by fluctuations of the bit rate of the traffic aggregates on a link due to normal stochastic behavior (a), by traffic shifts within the network due to popular content, so called hot spots, (b), or by redirected traffic due to network failures (c). Overload Capacity dimensioning methods for CO need to take into account all potential sources of overload (a), (b), and (c). In contrast, AC can block excess traffic if overload is caused by (a) and (b). However, AC cannot prevent overload due to network failures (c). Flows once admitted to the network must be treated with the assured quality level also in case of network failures. Since redirected traffic due to failures is the most frequent reason for overload in core networks [136], resilient AC admits traffic only if it can be carried without QoS violation together with the redirected traffic of potential failure scenarios [147]. Thus, capacity dimensioning methods for resilient AC need to take overload due to (c) into account.

2.5.2 Capacity (Over-)Provisioning

Capacity overprovisioning (CO) provides sufficient bandwidth so that overload in networks becomes unlikely and achieves thereby the desired QoS. The link capacities are chosen in such a way that the predicted traffic exceeds them only very rarely. To model and predict the characteristics of the traffic in the network, traffic measurements are required and must be analyzed.

Traffic Characteristics and Measurements

Bandwidth provisioning in highly aggregated core networks is usually based on the characteristics of traffic aggregates, i.e. on the combined traffic stream of multiple flows. The aggregated number of bytes over time exhibits statistical properties like long-range dependence, self-similar scaling properties and non-stationarity [212]. These properties complicate traffic measurements and modeling.

The amount of network traffic increases constantly and follows diurnal and weekly patterns, which makes the underlying process non-stationary over long time intervals. However, the authors of [212] also found non-stationarity at multi-second time scales possibly caused by the superposition of the stationary, high-variance packet inter-arrival time distributions of the single sources. This requires change-point tests for the identification of change free regions where the traffic can be considered stationary. Traffic models must incorporate this piecewise stationarity and adjust the parameters accordingly to the individual regions.

The phenomenon of long-range dependence has first been discovered in local area networks [213] but then also in wide-area networks [214] and for general WWW traffic [215]. It refers to large correlations in the aggregate number of bytes in time and makes the observation of the corresponding process difficult since observations tend to be higher or lower than the mean of the process for long durations of time. Thus, an accurate measurement of the mean traffic rate requires long observation intervals, which possibly causes problems due to only piecewise stationarity.

Self-similar processes behave the same on different scales in time. Self-similarity implies long-range dependence, but not all long-range dependent processes are self-similar. Self-similarity is problematic for network traffic since it may lead to significant packet loss with finite buffers [216]. However, for moderate utilization values, a superposition of sufficiently many long range dependent traffic sources may lead to the same buffer overflow probability in the limit as a Poisson process [217].

For high speed links, the measurement results in [218] show that traffic fluctuations at small time scales tend to be rather uncorrelated. On a scale of 1 s and above, however, they reveal a self-similar structure. Similarly, [219, 220] finds empirical evidence that on links with a high level of aggregation the amount of traffic arriving in small intervals is modeled well by the Gaussian distribution. The minimum time scale for the Gaussian assumption to hold on links with average traffic rate greater than 50 Mb/s is between 1 - 8 ms. This is in contrast to earlier studies where the distribution for the amount of traffic seen within intervals of less than several hundred milliseconds was quite complex. Hence, the high level of aggregation facilitates bandwidth provisioning in core networks since the time scales relevant to the buffering behavior of high speed links allow for Gaussian modeling [219].

Generally, it is problematic that rate measurements are often obtained by SNMP on a time scale of 5 min. Those measurements reveal substantially smaller variations than traffic on a small time scale like 10 ms. The difference may be 100% or more [221]. This makes it difficult to derive suitable parameters for small time scales that are required for bandwidth provisioning using – for instance – a Gaussian model.

Bandwidth Provisioning

Bandwidth provisioning procedures differ fundamentally from access to core networks since the lower number of users in the access inherently limits the aggregation level [220].

The network dimensioning approach in [219] supports latency sensitive traffic. Accordingly, the QoS measure considered for network dimensioning is the probability that the queue length Q of a router exceeds a certain value x: $\Pr(Q > x)$. To satisfy end-to-end delay requirements as low as 3 ms requires only 15% extra bandwidth above the average data bit of the traffic in the highly aggregated Sprint network.

The work in [222] focuses on the probability that the amount of traffic $A(T)$ generated on a link within a specified time interval T exceeds the capacity C of the link: $\Pr(A(T) \geq C \cdot T)$. The authors argue that applications can cope with lack of bandwidth within an application-dependent small interval T if this occurs sufficiently rarely. They develop an interpolation formula that predicts the bandwidth requirement on a relatively short time scale in the order of 1 s by relying on coarse traffic measurements. So-called 'user-oriented' and 'black box' traffic models are used to characterize measurement results. They are evaluated in [223] with regard to their accuracy for link provisioning. It turned out that black box models are easier to apply and yield slightly conservative capacity estimates, which makes them reliable provisioning guidelines.

Traffic Forecasting

Another problem closely related to the network dimensioning problem is forecasting of Internet traffic. An approach for long-term forecasting can be found in [224]. The authors of [225] combine traffic forecasting and network dimensioning to yield an adaptive bandwidth provisioning algorithm. Based on measurements, the required capacity is predicted and adjusted on relatively small time scales between 4 s and 2 min. The Maximum Variance Asymptotic (MVA) [226] approach for the tail probability of a buffer fed by an input Gaussian process is used to make the QoS requirement $\Pr(delay > D) < \epsilon$ explicit.

2.5.3 Admission Control

Admission control (AC), proposed for the Internet in [206], limits the number of flows in the network. It denies access to new flows if the network risks to be overloaded. Admission control mechanisms have two objectives. On a single link they perform link AC (LAC) and decide whether the admission of a new flow compromises the QoS in terms of packet loss and delay on that link. In a network they perform network AC (NAC) and decide whether the admission of a new flow violates the QoS on any link of its path. Numerous methods and protocols have been proposed to solve both aspects. Implementations always have to solve both issues, even if one of them is implemented in a trivial way. An extensive overview on AC can be found in [147].

Link Admission Control

Link AC (LAC) concentrates on a single resource and primarily on the packet level. The methods can be roughly subdivided into descriptor based, measurement based, and hybrid LAC mechanisms.

Descriptor Based LAC Connection requests carry a flow descriptor [227] that typically characterizes the rate and the variability of a flow on different time scales. This may be done by a single or dual token bucket which includes mean and peak rates. To enforce the conformance of the flow characterization on input and output interfaces of a router, policers and spacers may be used. The admission decision is based on the flow descriptor and the amount of already admitted traffic.

Measurement Based LAC and Hybrid Methods Measurement based AC uses measurements to determine the bandwidth requirements of individual admitted flows [228] or of the entire admitted traffic aggregate [229]. From these values it derives the available bandwidth for additional flows. [230] suggests a hybrid approach that uses descriptors and additionally determines a

feasible degree of overbooking which is obtained through measurement experience from the past.

Network Admission Control

Network AC (NAC) prevents overload on multiple resources within a network. This is a non-trivial task if the decision should be made solely at the network border without cooperation of interior nodes.

Link-by-Link NAC The most intuitive NAC implementation is certainly the application of LAC for each link along the path of a flow. The reservation for the flow is admitted if and only if all AC decisions succeed. This requires interior nodes of a network to keep per flow states which is difficult to handle, in particular when network failures occur.

Feedback Based NAC Several protocols rely on feedback from the network. They perform stateless core admission control and avoid explicit per-flow signaling messages. A recent feedback based NAC approach is based on pre-congestion notification (PCN) [231]. Each link is monitored by a PCN router and all its packets are "admission-stop" marked if the current traffic rate on the link exceeds a pre-configured threshold. The egress routers Z measure the rate of admission-stop marks for all ingress routers Y separately. If the fraction of marked packets exceeds a certain threshold, the egress Z signals admission-stop to the ingress Y. If the fraction drops again below this threshold, Y may continue to admit new flows.

B2B Budget (BBB) Based NAC The border-to-border (b2b) budget (BBB) based NAC defines capacity budgets for each b2b relationship (v, w) within the network and assigns them a capacity portion. A new flow originating at ingress router v and destined for egress border router w asks for admission only at its ingress router v. This ingress router performs AC based on the a priori

dedicated capacity budget $BBB(v, w)$ like on a single resource. This concept is implemented, e.g., by label switched paths (LSPs) in MPLS.

Resilient NAC Resilient NAC reserves backup capacities in advance to protect redirected traffic during failure cases and to avoid heavy reservation signaling in such a critical situation. According to [147], the simplest and most efficient resilient NAC implementation is the enhancement of the BBB NAC. The virtual capacity budgets $BBB(v, w)$ are just set low enough such that the redirection of admitted traffic cannot cause overload on any link when a failure occurs. The configuration of the budgets for resilient BBB NAC is well feasible and leads to reduced but still acceptable resource utilization. Since we assume network resilience as a mandatory requirement for carrier grade networks, in the following AC refers to the non-resilient and resilient version of BBB NAC based on our results in [4, 10]. We performed a similar analysis for the link-by-link (LBL) NAC in [5].

2.5.4 Comparisons of AC and CO

We briefly address other comparisons of AC and CO to distinguish them from our work.

The work in [210] considers utility functions for different applications and different flow level models including the Poisson model. They are used to compare the additional capacity above the mean rate that is required for networks with reservations and for networks with a best effort service. In case of the Poisson model, they find only marginal benefits for AC versus CO while for other flow level models AC reveals clear benefits. The study regards only a single link such that questions like the impact of overload due to traffic shifts and redirected traffic are out of scope.

A comparison of AC and CO in access network dimensioning is the topic of [232]. They consider the aggregation link in a hierarchically structured access network and find a clear benefit of AC. Depending on blocking probability, packet

loss probability, and user activity, the number of subscribers for a given access network capacity can be substantially higher when AC is used. In contrast, our work focuses on the dimensioning of an entire network and considers potential traffic shifts and redirected traffic.

2.5.5 Our Contribution towards Resilient Network Provisioning

The discussion whether AC or CO is the more suitable concept for efficient dimensioning of quality-enabled networks is mainly based on assumptions and beliefs. In Chapter 5, we provide insights into this discussion by quantifying and comparing the required capacity for CO and AC under potential overload and resilience requirements. We do not concentrate on methods for obtaining reliable traffic matrix estimates or specific packet level traffic models, we establish a more general basis for the comparison. In particular, the contributions in Chapter 5 of this work are (1) the presentation of a capacity dimensioning method for networks with resilience requirements and changing traffic matrices, (2) the investigation of the impact of the mentioned sources of overload (a-c) on the required capacity for CO in networks with and without resilience requirements, and (3) a comparison of this required capacity with the one for AC.

3 Load Balancing for Multipath Internet Routing

In this chapter, we evaluate load balancing for multipath Internet routing. First we present a new classification of hash-based algorithms that includes existing and new ones in Section 3.1. Then we present our evaluation method for assessing the load balancing accuracy and dynamics over time in Section 3.2. Based on this and the notion of single-stage and multi-stage load balancing introduced in Section 3.3, the actual performance evaluation of load balancing algorithms can be found in Sections 3.4 and 3.5. Section 3.6 summarizes this chapter.

This chapter is based on basic principles described in Chapter 2, mainly in Sections 2.2 and 2.3

3.1 An Overview of Hash-Based Load Balancing Algorithms

We introduce our notation for the formalization of the problem of load balancing for multipath forwarding. In this chapter, we refer to load balancing for multipath forwarding simply by load balancing since we concentrate on this particular application.

3.1.1 A Formal Notation

The set of outgoing links (interfaces) $\mathcal{L}(r,d)$ at router r to destination d can be derived from the routing table and corresponds to the paths used from r to d. All flows at a certain router r with destination d are denoted by the flow set $\mathcal{F}(r,d)$. This is not the set of currently active flows, but the set of all possible flows. The destination d actually represents the set of destinations subsumed by one entry in the routing table. Hence, the flows in $\mathcal{F}(r,d)$ are all spread over the same interfaces. The target load fraction $tLF(r,d,l)$ for a specific outgoing link $l \in \mathcal{L}(r,d)$ describes the desired load balancing objective as a percentage of the total traffic forwarded at any time instance at router r towards destination d over link l. Thus, the condition $\sum_{l \in \mathcal{L}(r,d)} tLF(r,d,l) = 1$ must be fulfilled. For instance, if router r uses two outgoing links l_0 and l_1 to spread the traffic towards d equally, then $\mathcal{L}(r,d) = \{l_0, l_1\}$ and $tLF(r,d,l_0) = tLF(r,d,l_1) = 50\%$.

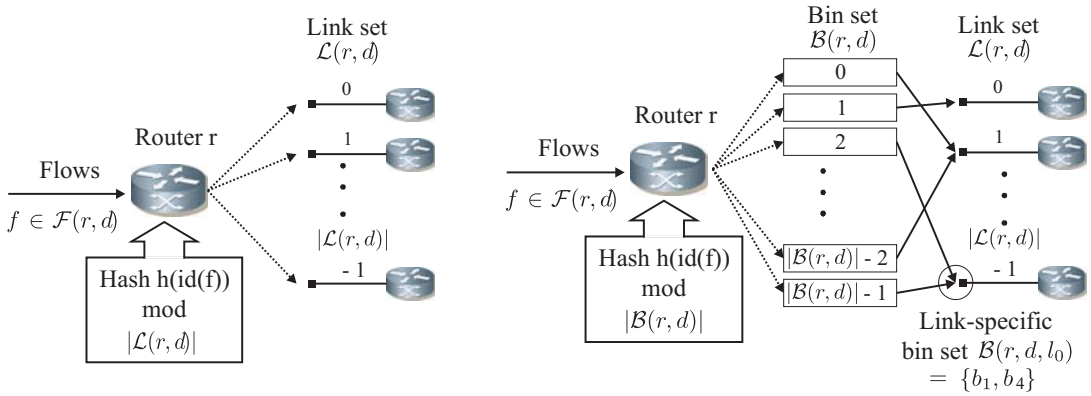
Hash-based load balancing algorithms use a hash function $h(\cdot)$ to compute a hash value $h(id(f))$ from the characteristic flow ID $id(f)$ of every packet destined to d. A link selector function $s_{r,d}(h(id(f)))$ then yields the outgoing interface $l \in \mathcal{L}(r,d)$ from the respective set of outgoing links based on the hash value. This functional approach avoids the need to store the corresponding outgoing interface for every flow separately. We use the 16-bit cyclic redundancy check (CRC) in our experiments as recommended in the analysis [67] of different hash functions for this purpose. The flow ID $id(f)$ consists mostly of the five-tuple source and destination IP address, source and destination port number, as well as protocol id, or a subset thereof, which are part of the invariant header field of each packet. Thus, hash-based algorithms differ with respect to the applied hash function h and link selector functions $s_{r,d}$.

We assume that the current traffic rate $cTR(r,d,l)$ at router r over a specific link $l \in \mathcal{L}(r,d)$ to destination d can be obtained by some means, e.g. by online measurements [2]. It allows for calculating the current load fraction $cLF(r,d,l) = \frac{cTR(r,d,l)}{\sum_{l' \in \mathcal{L}(r,d)} cTR(r,d,l')}$. If it differs substantially from the target load fraction $tLF(r,d,l)$ due to stochastic effects, a change of the link selec-

tor function $s_{r,d}$ is required. For instance, if currently $cLF(r,d,l_0) = 40\% < 50\% = tLF(r,d,l_0)$ and $cLF(r,d,l_1) = 60\% > 50\% = tLF(r,d,l_1)$ for the example from above, flows should be relocated from l_1 to l_0 to abolish this imbalance.

3.1.2 Static and Dynamic Load Balancing Algorithms

Static load balancing algorithms do not allow such a change of the link selector function $s_{r,d}$ while dynamic algorithms automatically adapt their link selector function to achieve a new balanced traffic distribution.



(a) Direct link selector function. (b) Table-based link selector function.

Figure 3.1: *Data structures of direct and table-based link selector functions.*

Static Hashing

Link selector functions perform either a direct mapping between hash values and links or an indirect, table-based mapping using intermediate data structures.

Direct Hashing Direct link selector functions may be implemented by a simple modulo operation, i.e., $\mod (h(id(f)), |\mathcal{L}(r,d)|)$ determines the num-

ber of the outgoing interface within the link set. This leads to an even objective distribution of the traffic aggregate $\mathcal{F}(r,d)$ over the links in $\mathcal{L}(r,d)$: $tLF(r,d,l_i) = tLF(r,d,l_j)\ \forall l_i,l_j \in \mathcal{L}(r,d)$. The data structure of such a direct link selector function is illustrated in Figure 3.1(a).

Table-Based Hashing Target load fractions other than even load distribution can be obtained by table-based link selector functions. They perform an indirect mapping from the hash value $h(id(f))$ to an outgoing interface $l \in \mathcal{L}(r,d)$ via so-called intermediate bins. The bins have pointers to the outgoing interfaces. The entire bin set is denoted by $\mathcal{B}(r,d)$ and the bins are numbered 0, ..., $(|\mathcal{B}(r,d)|-1)$. Now, the table-based link selector function consists of a bin selector function (e.g. $mod(h(id(f)),|\mathcal{B}(r,d)|)$) that maps a hash value to a specific bin, and the pointer of the bin that further directs the flow f to an interface. The data structure of such a table-based link selector function is illustrated in Figure Figure 3.1(b). The link specific bin set $\mathcal{B}(r,d,l)$ contains all bins of $\mathcal{B}(r,d)$ with pointers to l.

Dynamic Hashing

For static link selector functions, the assignment between bins and links is fixed. Dynamic algorithms adapt their link selector functions to the current load conditions during runtime. Increasing the link specific bin set $\mathcal{B}(r,d,l)$ of a link l increases also the current load fraction of l. This is achieved by redirecting pointers to l from bins with pointers to other links. The reduction of the current load fraction of a link l works analogously. Dynamic algorithms check the current load difference

$$cLD(r,d,l) = cLF(r,d,l) - tLF(r,d,l) \tag{3.1}$$

for any link $l \in \mathcal{L}(r,d)$ from time to time, e.g. in periodic intervals of length $t_r = 1$ s, and reassign the pointers of the bins if needed. Links with a positive $cLD(r,d,l)$ are called overloaded and those with a negative $cLD(r,d,l)$ are called underloaded. In the example from above, link l_0 is underloaded with a

current load difference $cLD(r,d,l_0) = cLF(r,d,l_0) - tLF(r,d,l_0) = 40\% - 50\% = -10\%$. Link l_1 is overloaded with $cLD(r,d,l_1) = 10\%$. A link l may be overloaded with regard to some flow set $\mathcal{F}(r,d)$ and, simultaneously, it may be underloaded with regard to some other flow set towards other destinations.

3.1.3 Hash-Based Load Balancing Algorithms under Study

The reassignment of dynamic load balancing algorithms can be decomposed into a bin disconnection and a bin reconnection step. Here we introduce a modular composition of load balancing algorithms based on algorithms from literature and new ones, propose algorithms consisting of a combination of different disconnection and reconnection strategies, and evaluate their performance. Some of the algorithms are simple, others are rather complex – depending on the number of reassigned bins. All algorithms are greedy. They are only heuristics and do not achieve the optimal accuracy. However, [104] demonstrated that finding the optimal solution to the load balancing problem with minimal flow re-mapping is NP-hard even if we knew the exact packet sequence in advance. Since this is of course not true for a realistic load balancer, simplicity and fast execution counts more than optimality.

In the following, the size of a bin $b \in \mathcal{B}(r,d)$ is determined by its current traffic rate $cTR(r,d,b)$. It is the overall rate of the currently active flows $f \in \mathcal{F}(r,d)$ whose IDs $id(f)$ are mapped to b via the hash and the modulo function. The current traffic load fraction of a bin is defined by $cLF(r,d,b) = \frac{cTR(r,d,b)}{\sum_{b' \in \mathcal{B}(r,d)} cTR(r,d,b')}$. This definition is analogous to the definitions for links.

Bin Disconnection Strategies

Bin disconnection strategies differ with regard to the number of simultaneously disconnected links, i.e., they disconnect either only a single bin or multiple bins in the disconnection step. Furthermore, disconnection strategies may be conser-

vative (+), i.e., they try to avoid to bring overloaded links into underload; or they may be progressive (-), i.e., they are allowed to bring overloaded links into underload.

Single Bin Disconnection (SBD) Both single bin disconnection strategies ($SBD^{+/-}$) are illustrated in Figure 3.2.

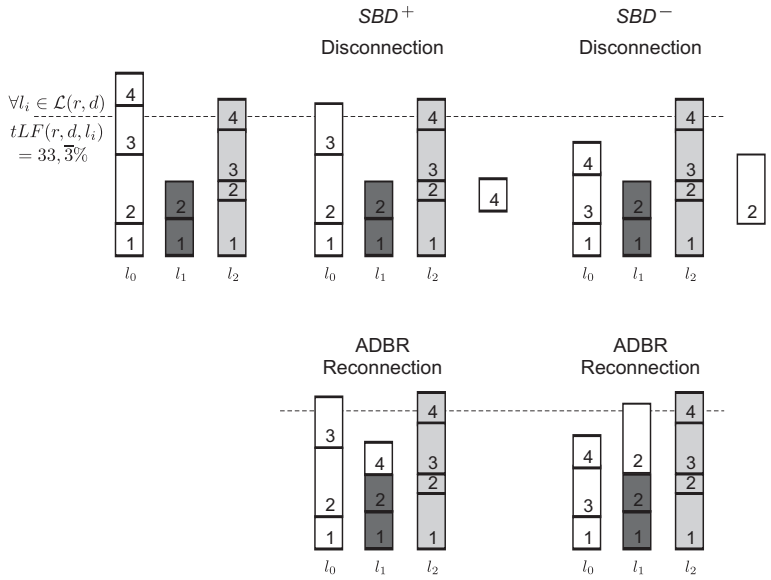


Figure 3.2: *The single bin disconnection $SBD^{+/-}$ strategies relocate only one bin in each step from the three links l_0, l_1, l_2.*

Conservative Single Bin Disconnection (SBD^{+}) The conservative single bin disconnection strategy (SBD^{+}) disconnects from the link-specific bin set $\mathcal{B}(r, d, l)$ of the link with the largest overload the largest bin b that does not turn the link into underload. SBD^{+} avoids to bring any link into underload and is therefore called conservative (+). This avoids heavy oscillations when big bins that turn links into underload are moved back and forth between a few links at successive reassignment steps. SBD^{+} does not disconnect any bin if the disconnection of the smallest bin from the heaviest loaded link l turns the link l into underload.

Progressive Single Bin Disconnection (SBD^-) The dynamic load balancing algorithm in [71] proposed for best accuracy disconnects the largest bin from the link-specific bin set $\mathcal{B}(r,d,l)$ of the link l with the largest overload. It is irrelevant, whether the considered link l then is underloaded or not. Therefore, this strategy is progressive and we denote it by SBD^-.

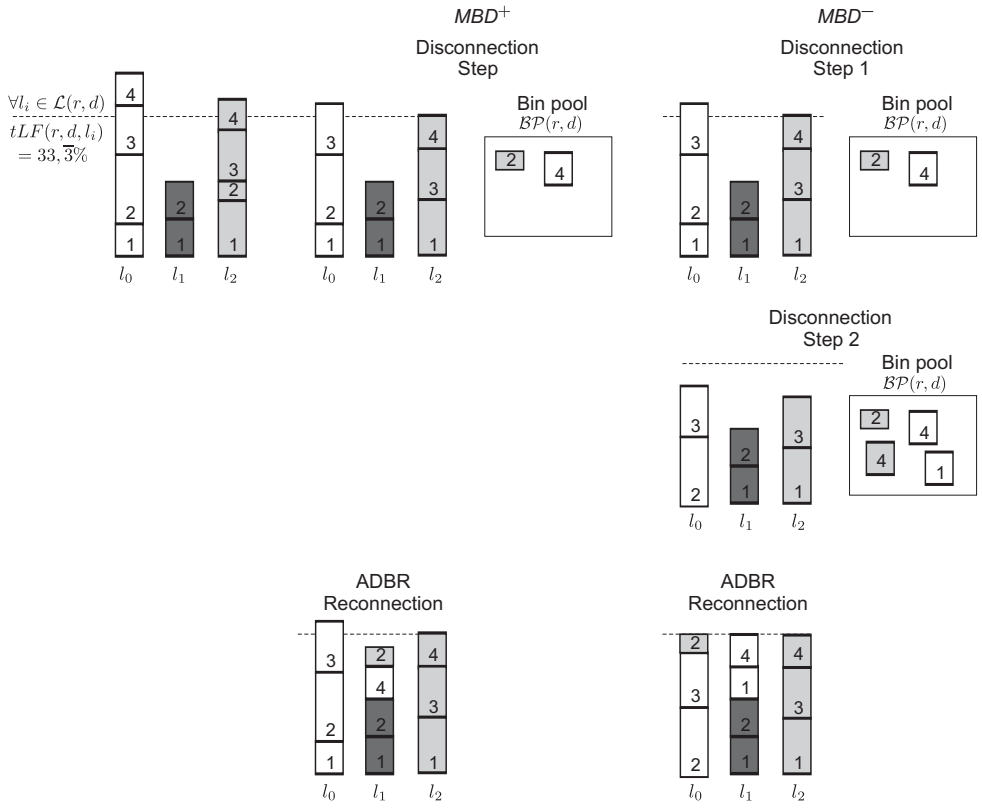


Figure 3.3: *The multiple bin disconnection $MBD^{+/-}$ strategies relocate several bins in each step from the three links l_0, l_1, l_2.*

Multiple Bin Disconnection (MBD) Both multiple bin disconnection strategies ($MBD^{+/-}$) are illustrated in Figure 3.3.

Conservative Multiple Bin Disconnection (MBD^+) In contrast to SBD^+, the conservative multiple bin disconnection strategy (MBD^+) dis-

connects from all overloaded links so many bins until any further removal turns them into underload. The bins within the link-specific sets $\mathcal{B}(r, d, l)$ are checked in the order of decreasing size for removal. The disconnected bins are collected in a so-called bin pool $\mathcal{BP}(r, d)$.

Progressive Multiple Bin Disconnection (MBD^-) The progressive multiple bin disconnection strategy MBD^- works like MBD^+ in the first step (step 1), but it eventually turns each overloaded link l intentionally into underload by removing its smallest bin from its link-specific bin set $\mathcal{B}(r, d, l)$ (step 2). Therefore, we call this strategy progressive (-).

Bin Reconnection Strategies

After single or multiple bin disconnection, the bins must be reconnected to new links in such a way that the target load fraction $tLF(r, d, l)$ of each link l is met. Usually, this objective can be achieved only approximately. The resulting load balancing inaccuracy on any link l when reconnecting bins may be measured by the current load difference $cLD(r, d, l)$. As an alternative, this difference may be viewed in a relative way by the relative current load difference $rCLD(r, d, l) = \frac{cLD(r,d,l)}{tLF(r,d,l)}$.

The exact reconnection optimized for one of these measures is difficult, given the time constraints in high speed routing. Therefore, we solve it again by simple greedy approaches. For MBD, we sort all bins collected in the bin pool $\mathcal{BP}(r, d)$ according to their size and select bins in the order of decreasing size for reconnection to the bin sets. For the purpose of reconnecting each bin — in case of SBD the only bin — we propose two simple strategies.

Absolute Difference Bin Reconnection (ADBR) We reconnect the bin to the link l with the lowest current load difference $cLD(r, d, l)$, i.e. with the largest underload. The bin reassignment strategy $ABDR$ is illustrated in Figures 3.2 and 3.3.

Relative Difference Bin Reconnection (RDBR) In a first step, we try to reconnect the bin to the link l with the largest underload like above, but only if the new bin b does not turn the link into overload. This can be achieved if the following holds: $\exists l \in \mathcal{L}(r,d) : cLF(r,d,l) + cLF(r,d,b) \leq tLF(r,d,l)$. If this fails, we reconnect the bin b in a second step to a link l that obtains the lowest relative overload among all links in $\mathcal{L}(r,d)$ if the bin b is assigned to its link set $\mathcal{B}(r,d,l)$. Such a link can be formally described by $argmin_{l \in \mathcal{L}(r,d)}(\frac{cLD(r,d,l)+cLF(r,d,b)}{tLF(r,d,l)})$. The intuition behind it is the following. If there are two links l_0 and l_1 with current load difference $cLD(r,d,l_0) \approx cLD(r,d,l_1)$ but target load fractions $tLF(r,d,l_0) \gg tLF(r,d,l_1)$, link l_0 suffers from less overload relative to its target value than link l_1 after accommodating the new bin. With $ABDR$ this does not matter. For equal target load fractions $tLF(r,d,l_i)$ for all links l_i, there is no difference between $ABDR$ and $RDBR$. Therefore, Figures 3.2 and 3.3 do not illustrate $RDBR$.

Composition of Bin Reassignment Algorithms

We proposed several methods for the disconnection and reconnection of bins. The generic bin reassignment Algorithm 1 may be instantiated by any of the presented options SBD^-, SBD^+, MBD^+, and MBD^- for $BinDisconnection$ and $ABDR$ or $RDBR$ for $BinReconnect$. Thus, we get 8 substantially different bin reassignment methods by this generic approach.

$\mathcal{BP}(r,d) = BinDisconnection(\{\mathcal{B}(r,d,l) :$
$\qquad l \in \mathcal{L}(r,d) \wedge cLD(r,d,l) > 0\})$
while $\mathcal{BP}(r,d) \neq \emptyset$ **do**
  $b_{max} = argmax_{b \in \mathcal{BP}(r,d)}(cLF(r,d,b))$
  $BinReconnect(\{\mathcal{B}(r,d,l) : l \in \mathcal{L}(r,d)\}, b)$
end while

Algorithm 1: Generic bin reassignment method.

In the following, we use a slash-notation to refer to the actual instances of the algorithms, e.g. $SBD^-/ADBR$. This is the algorithm proposed by [71], while

the other 7 combinations are new methods. Simplicity and fast execution of the algorithms was the design goal for the algorithms.

3.2 Evaluation Method for Hash-Based Load Balancing

In this section we provide an evaluation methodology for hash-based load balancing algorithms. We use a simulation methodology on the flow level in topologies suitable for the various problems concerning load balancing. Our simulation methodology uses synthetic flow IDs instead of packet traces and generates the flows according to a Poisson model. We motivate these assumptions in the following.

Flow Level Simulation

Many related studies perform a fully detailed network simulation on the packet level to measure the packet reordering probability. However, the obtained results depend significantly on the network topology and the routing, on the latency of different paths, and on the queueing delay caused by cross traffic. Thus, there are many other factors but load balancing that influence the packet reordering probability. Therefore, we rather focus on the flow reassignment rate λ_{FR}. It is affected only by the dynamic load balancing and influences the packet reordering probability proportionally. In addition, flow level simulations run much faster and allow us to produce very reliable results.

Synthetic Flow ID Generation

Often, real traffic traces are used to evaluate the quality of load balancing mechanisms and to point out that the results are realistic. This is important for assessing

the quality of hash functions for a certain application. In our study, we use the 16-bit CRC function because the authors of [67] have shown that "hashing using a 16-bit CRC over the five-tuple gives excellent load balancing performance", i.e., it spreads the flow IDs almost uniformly over the codomain of the hash function. We are interested in the general potential of static and different dynamic load balancing algorithms and not in the quality of different hash functions. Therefore, we use synthetically generated flow IDs with uniform distribution to avoid any correlation effects within a specific trace.

Traffic Model

The interarrival time of flows on Internet links are exponentially distributed with rate λ_{IAT} [214, 233, 234]. Therefore, the Poisson model is well applicable on the flow level. The call holding times are identically and independently distributed with a typical mean value of $E[B] = 90$ s. Thus, the offered load can be calculated by $a = \lambda_{IAT} \cdot E[B]$ measured in Erlang (Erl) and can be viewed as the average number of simultaneous flows. The variation of the bit rates of different flows has a significant impact on the quality of the load balancing mechanisms [103]. In fact, there are a few large flows (elephants) producing fifty to sixty percent of the total traffic while the rest is due to many small flows (mice) [235, 236]. As a consequence, our traffic model is multi-rate to capture this effect. We use $n_r = 3$ different flow types $r_i, 0 \leq i < n_r$ with flow rates $c(r_i) \in \{64, 256, 2048\}$ kbit/s. Details can be found in [147]. We also use this traffic model in Chapter 5.

Performance Measures and Simulation Credibility

We consider two important performance aspects for load balancing algorithms: load balancing accuracy and dynamics in terms of the flow reassignment rate.

For the load balancing accuracy, a time-weighted histogram captures the measurements of the current load fraction $cLF(l)$ for each link $l \in \mathcal{L}$. This reveals the load balancing accuracy over time. Since this information is hard to grasp for more than two links, we additionally define the absolute deviation of the load

fractions $cLF(l)$ from their target values $tLF(l)$ averaged over all links $l \in \mathcal{L}$

$$\begin{aligned} I(r,d) &= \frac{1}{|\mathcal{L}(r,d)|} \sum_{l \in \mathcal{L}(r,d)} |cLF(r,d,l) - tLF(r,d,l)| = \\ &= \frac{1}{|\mathcal{L}(r,d)|} \sum_{l \in \mathcal{L})(r,d} |cLD(r,d,l)| \end{aligned} \tag{3.2}$$

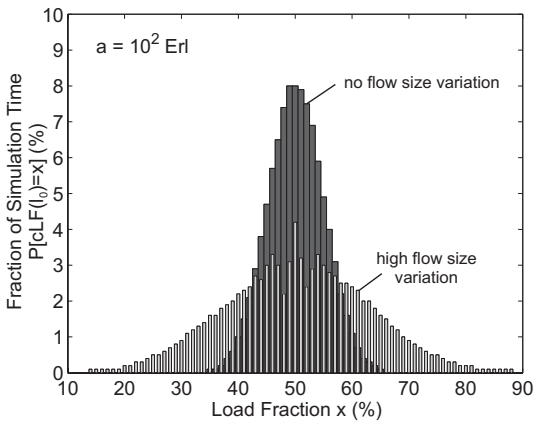
and use its mean $E[I(r,d)]$ to capture the inaccuracy by a single number.

For the load balancing dynamics, we capture the flow reassignment rate λ_{FR} as the fraction of all flow reassignments during the simulation and the overall lifetime of all simulated flows. Its reciprocal can be viewed as the average time between two reassignments of a flow.

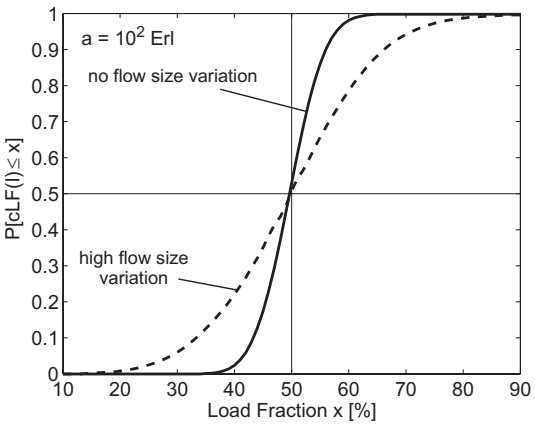
To obtain credible simulation results, we calculate confidence intervals for all performance metrics used in this work based on standard simulation techniques such as replicate-delete [237]. We simulate so long that the 99% confidence intervals deviate at most 1% from the respective mean values. Thus, they are very small which proves the statistical significance of our results. As they are hardly visible, we do not show them in the following figures.

Example: Impact of Flow Rate Variability

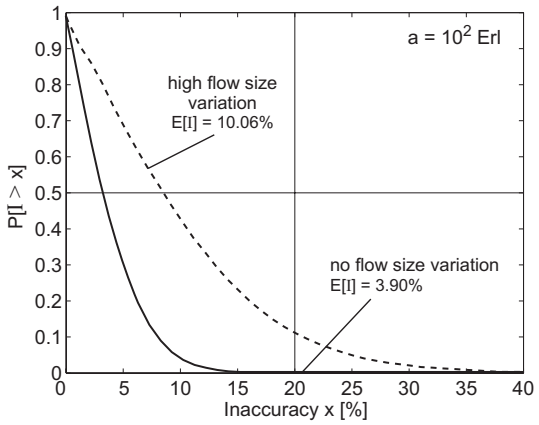
To give an example of our performance measures from above and to illustrate our visualization techniques, we consider the following simple experiment. A single router r is supposed to split the flows at r destined for d over two outgoing interfaces l_0 and l_1 with given equal target load fraction $tLF(r,d,l_0) = tLF(r,d,l_1) = 50\%$. The load balancing algorithm uses the simple static direct hashing. We assume an offered link load of 100 Erl and consider the influence of the flow rate variability. In the case of homogenous flows, all flows have a bit rate of 256 $kbit/s$ whereas in the case of heterogeneous flows, the flows have only 64 $kbit/s$ and 2048 $kbit/s$ but the same mean of 256 $kbit/s$, which yields a high coefficient of variation of 2.29.



(a) Time-weighted histogram of the current load fraction $cLF(l_0)$ on link l_0.



(b) Cumulative distribution function (CDF) for the current load fraction $cLF(l_0)$ on link l_0.



(c) Complementary cumulative distribution function (CCDF) for the inaccuracy I for both links l_0 and l_1 according to Equation 3.2.

Figure 3.4: *Impact of flow rate variability on the traffic distribution for static load balancing.*

Figure 3.4(a) presents the time-weighted histogram of the current load fraction $cLF(l)$ for both homogeneous and heterogeneous flows for link l_0. Since we consider only two links, the result for link l_1 is symmetric. The x-axis shows the load fraction on link l_0 in percent with a granularity of 1%. The y-axis shows the corresponding percentage with which the respective load fraction could be

observed in the simulation. The histogram for homogeneous flows illustrates that the measured load fraction varies from 0.35 to 0.65 in spite of a target load fraction of 0.5. The histogram follows exactly a binomial distribution according to $P(cLF(l) = i(\%)) = \binom{100}{i} 0.5^i (1-0.5)^{100-i} = \binom{100}{i} 0.5^{100}$. In case of heterogeneous flows, the broader histogram shows that the load balancing accuracy is significantly decreased.

The information in the histograms of Figure 3.4(a) is presented as cumulative distribution functions (CDFs) in Figure 3.4(b). The x-axis is like above but the y-axis shows the probability that the observed load fractions are smaller than or equal to a value x. The load balancing accuracy is high if the curve increases around the respective target load fraction $tLF(l)$ with a steep slope. Figure 3.4(a) is more intuitive but the curves in Figure 3.4(b) are easier to differentiate.

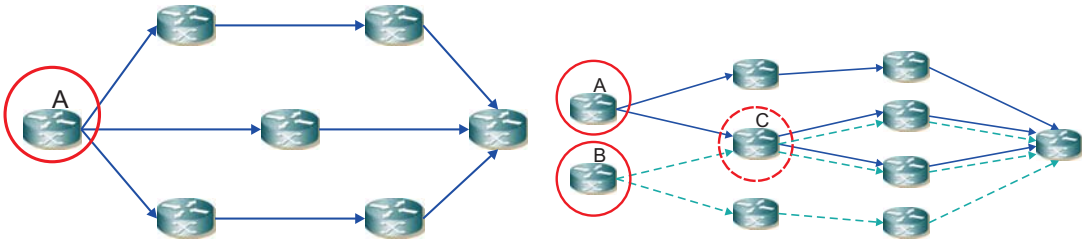
Figure 3.4(c) shows the load balancing inaccuracy I defined in Equation 3.2 as complementary cumulative distribution functions (CCDFs). The x-axis shows the load balancing inaccuracy in per-cent, the y-axis indicates the probability that the absolute value of the current load difference $|cLD(r, d, l)|$ averaged over both links l_0 and l_1 is larger than a value x. The load balancing inaccuracy is low if the curve approaches the x-axis fast. The average inaccuracy $E[I]$ increases from 3.90% to 10.06% for heterogeneous flows.

In the following we use the presentation from Figure 3.4(b) in simple experiments with two links only and the presentation from Figure 3.4(c) in more complex experiments with more than two links.

This example shows that the flow rate variability has a clear impact on the load balancing accuracy. If all flows have the same size, the task of load balancing reduces to the problem of distributing the active flows over the paths just according to their number and not to their rate. Heterogeneous flow rates complicate this task with an increasing variability. This finding is in line with the results in [103, 236]. Hence, we conduct all further studies with heterogeneous flows because this model is more realistic.

3.3 Single-Stage and Multi-Stage Load Balancing

Technologies like the Self-Protecting Multipath (SPM) transmit the traffic over several disjoint paths between source and destination according to a pre-configured load balancing function. Hence, in this application scenario the traffic is balanced only once (cf. Figure 3.5(a)). But load balancing is also required for equal cost multipath (ECMP) routing with OSPF [48] and IS-IS [49, 50]. This application scenario differs from the SPM by the fact that traffic undergoes load balancing possibly more than once and that the amount of input traffic for a load balancer depends on preceding load balancers, which is illustrated by router C in Figure 3.5(b).



(a) The SPM load balances the traffic only once.

(b) ECMP routing causes the traffic to undergo multiple load balancing steps.

Figure 3.5: *Single-stage and multi-stage load balancing applications.*

This leads to the notion of single-stage load balancing [12] if the traffic is load balanced only once on its way from source to destination and to the notion of multi-stage load balancing [19] if the traffic possibly undergoes multiple load balancing steps.

We evaluate both applications separately since single-stage load balancing reveals the general properties of the algorithms while multi-stage load balancing has additional inherent problems: (1) flows forwarded by an earlier hash-based load balancer over a specific interface are “polarized” such that a succeeding load balancer is potentially not able to spread this traffic aggregate anew [45]; (2) flow

reassignments by a preceding dynamic load balancer entails possibly further flow reassignments at succeeding load balancers since suddenly missing or new flows affect their traffic distribution.

3.4 Evaluation of Single-Stage Load Balancing

We first evaluate single-stage load balancing to examine the general potential of the algorithms under study and to extract important parameters that influence the load balancing result.

3.4.1 Simulation Topology

Since we are interested in the load balancing behavior for a flow set $\mathcal{F}(r, d)$ at router r and destined for destination prefix d, the simulation topology is very simple. We simulate only the traffic distribution to a given number of interfaces at a single node according to a given target load fraction $tLF(r, d, l)$.

In the following, we fix the parameters r and d and abandon them from our notation for easier readability. This is possible as we consider only a single router and a single destination prefix d.

3.4.2 Evaluation Results

3.4.3 Impact of Exogenous Parameters on the Accuracy of Static Load Balancing

The exemplary experiment from above already demonstrated the influence of the flow rate variability on the load balancing accuracy of static load balancing. Here, we further study the influence of the offered load as the second important exoge-

nous parameter in a load balancing setting. We assume only two outgoing links over which the traffic should be equally forwarded.

Figure 3.6 shows the load balancing accuracy for an offered load of $a = 10^{\{2,3,4\}}$ Erl. It is clearly visible that the load balancing accuracy increases with the number of simultaneous flows. An offered load of 10 Erl is definitely too small for load balancing since we observe almost any load fractions between 0% and 100% and, thus, is not shown here. For 10^3 Erl we get better values between 0.38 and 0.62 for static load balancing algorithms and an average inaccuracy of 3.13% instead of 10.06% as observed for $a = 10^2$ Erl. Very high volume traffic aggregates at $a = 10^4$ Erl lead to almost perfect load balancing with a very low mean inaccuracy of 0.93%. In the following experiments, we consider an offered load of 10^2 Erl because it is a moderate aggregation level and, thereby, more challenging for the load balancing accuracy.

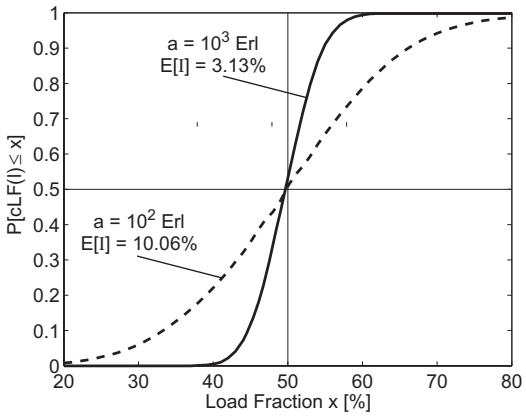


Figure 3.6: *Impact of the offered load on the traffic distribution for static load balancing.*

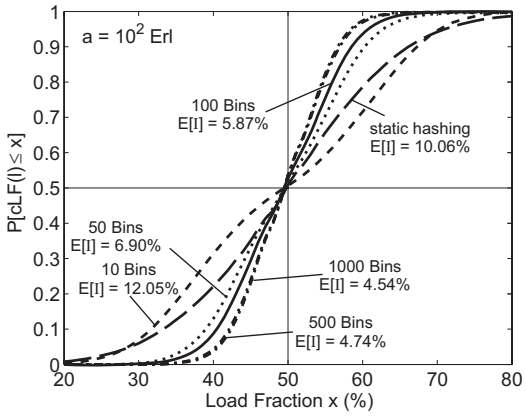


Figure 3.7: *Impact of the number of bins on the traffic distribution for $SBD^-/ADBR$ dynamic load balancing.*

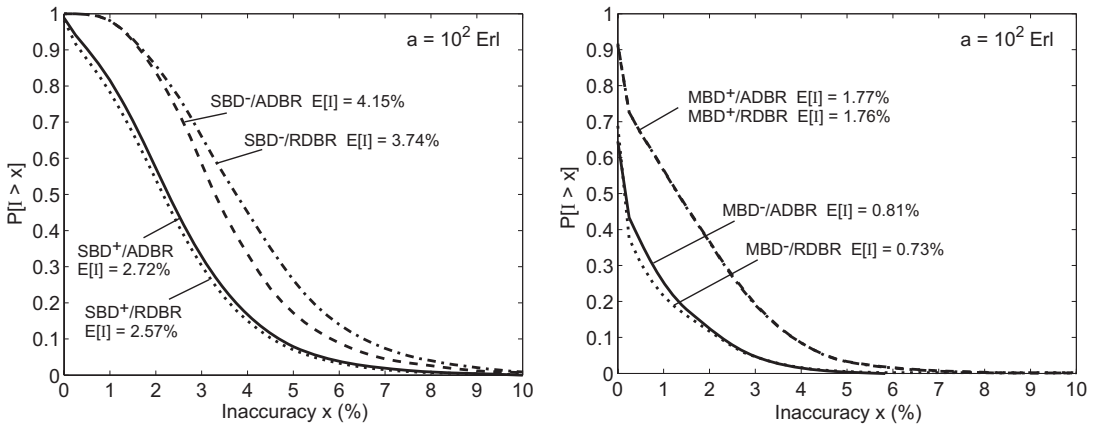
3.4.4 Accuracy Increase through Dynamic Load Balancing

Now we consider possible performance gains through dynamic load balancing algorithms and first analyze $SBD^{-}/ADBR$ as it has been proposed in [71]. We use a bin reassignment interval with a length of $t_r = 1$ s. The size of the bin set $\mathcal{B}$ is crucial for the accuracy of table-based load balancing. Figure 3.7 shows the distribution function of the load fraction for static load balancing and for dynamic load balancing with a different number of bins in the two-link experiment from above. With only 10 bins, the average load balancing inaccuracy $E[I] = 12.05\%$ is larger than for static load balancing ($E[I] = 10.06\%$). The small number of bins with dynamic adaptation is counterproductive. However, there is a significant improvement of the inaccuracy for 50 bins ($E[I] = 6.90\%$), 100 bins ($E[I] = 5.87\%$), and 500 bins ($E[I] = 4.74\%$). Another doubling of the number of bins does not lead to any clear performance gain ($E[I] = 4.54\%$).

The algorithms become more complex if the number of bins increases. Therefore, we work with 100 bins in the following since they lead to a sufficiently high accuracy and impose still moderate complexity, which is important for technical feasibility.

3.4.5 Comparison of the Accuracy of Different Dynamic Load Balancing Algorithms

In case of moderate aggregation level, static load balancing is not accurate enough, but dynamic load balancing has the potential to alleviate this problem as demonstrated above. We now compare different dynamic load balancing algorithms and use a more sophisticated experiment for that purpose. The traffic is distributed over four links with target load fractions of 10%, 20%, 30%, and 40%. We first study the inaccuracy of the single bin disconnection (SBD) and multiple bin disconnection (MBD) algorithm families. The inaccuracy I is a very intuitive measure, but it only should be used to compare the algorithms in the same



(a) The family of SBD-based algorithms. (b) The family of MBD-based algorithms.

Figure 3.8: *Complementary distribution function of the load balancing inaccuracy for various load balancing algorithms.*

scenario. Load balancing accuracy of scenarios with other target distribution values or even a different number of links cannot be compared by that approach. In addition, we show the details on each of the four links for SBD to motivate the observed performance. Finally, we contrast the detailed results for the best MBD algorithm to the SBD results to illustrate the potential of multiple bin disconnection.

Comparison of the Inaccuracy Distribution

Figure 3.8(a) illustrates the complementary distribution function of the inaccuracy I for the entire SBD algorithm family. The faster the curves decay, the better is the load balancing accuracy. The SBD^+-based algorithms ($E[I] = 2.57\%$ and 2.72%) are significantly more accurate than the SBD^--based algorithms ($E[I] = 3.74\%$ and 4.15%). For SBD^-, the version based on relative difference bin reassignment ($RBDR$) is significantly more inaccurate than the version based on absolute difference bin reassignment ($ABDR$) while there is hardly any difference between them for SBD^+.

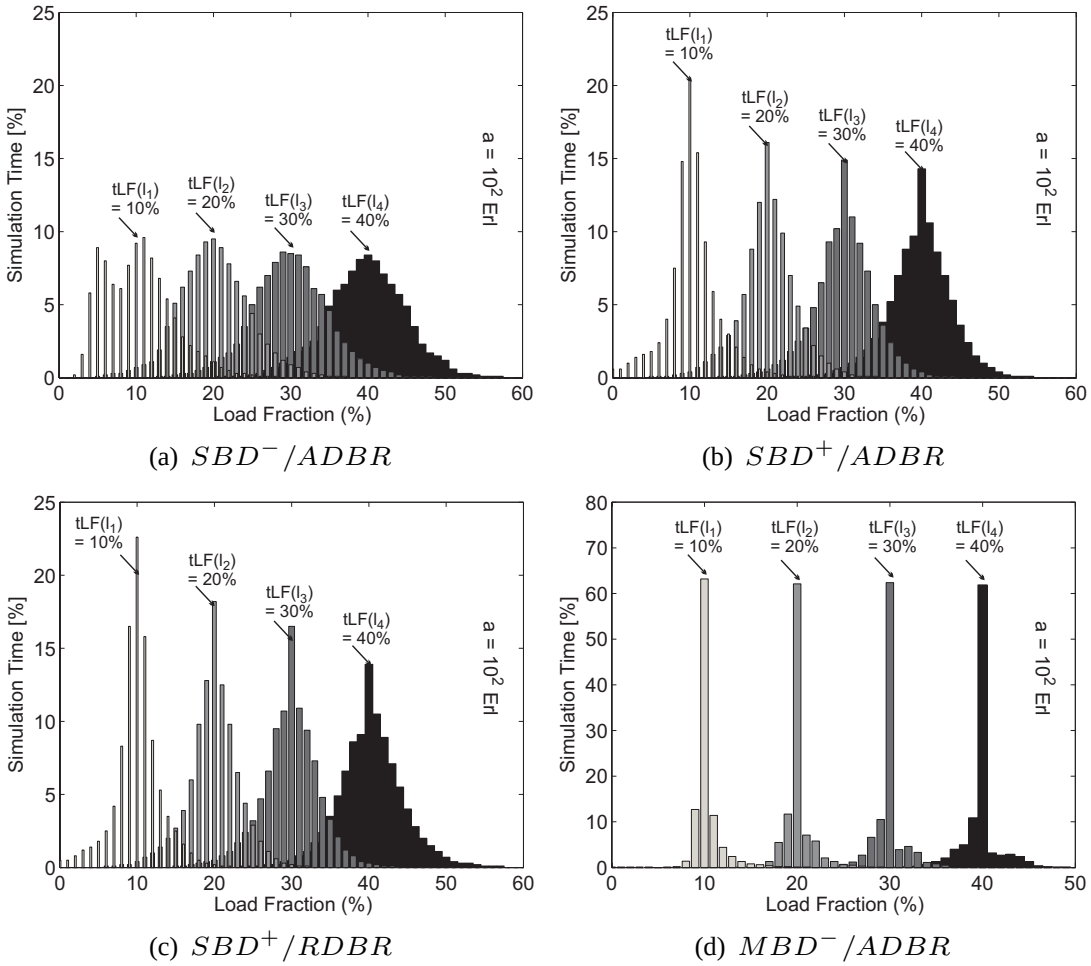
The MBD algorithm family outperforms the SBD family clearly as illustrated with the corresponding results for the MBD algorithms in Figure 3.8(b). The lines decay much faster. Here, the MBD^- versions ($E[I] = 0.73\%$ and 0.81%) are significantly more accurate than the MBD^+-based methods ($E[I] = 1.76\%$ and 1.77%). For MBD^-, the $RDBR$-based version is only little more inaccurate than the $ABDR$-based approach and for MBD^+ we cannot see any difference between them.

Comparison of SBD-Based Load Balancing Algorithms

Figures 3.9(a) – 3.9(c) show the histograms of the load fraction on each of the four links for various SBD-based algorithms to understand the above results in detail. For the $SBD^-/ADBR$ method in Figure 3.9(a), the deviations around the target load fraction is symmetric and similar for all links except for the one with the smallest target load fraction. This phenomenon is due to the flow size variation. Generally, the range of observed load fractions is still quite broad for $SBD^-/ABDR$. It removes always the largest bin from the link with the heaviest overload. This bin may be too large to balance the load and its disconnection causes significant underload on the considered link. In addition, this may cause oscillations if the same bin is exchanged back and forth between the same two links. As illustrated in Figure 3.9(b), the conservative algorithm $SBD^+/ABDR$ avoids this problem, leads to more accurate load balancing and clearly outperforms the progressive approach $SBD^-/ADBR$. It is interesting that links with a smaller load fraction have a larger peak around their target load fraction, which is a good feature. This effect is enforced by the $SBD^+/RDBR$ approach seen in Figure 3.9(c) as it tries to minimize the load deviation relative to the respective target value. However, the data reveal that the impact of the $RDBR$ strategy is quite weak. The improvement of the load balancing accuracy for links with a low target load fraction is reached at the expense of a slightly degraded load balancing accuracy for links with a high target load fraction. The same phenomenon can be observed with $SBD^-/ADBR$ and $SBD^-/RDBR$.

Potential of MBD-Based Load Balancing Algorithms

Figure 3.9(d) illustrates the load balancing accuracy for $MBD^{-}/ADBR$. It is significantly better compared to the SBD-based methods and to emphasize this, we draw particular attention to the differently scaled y-axis. This clearly shows the benefit of MBD opposed to SBD.

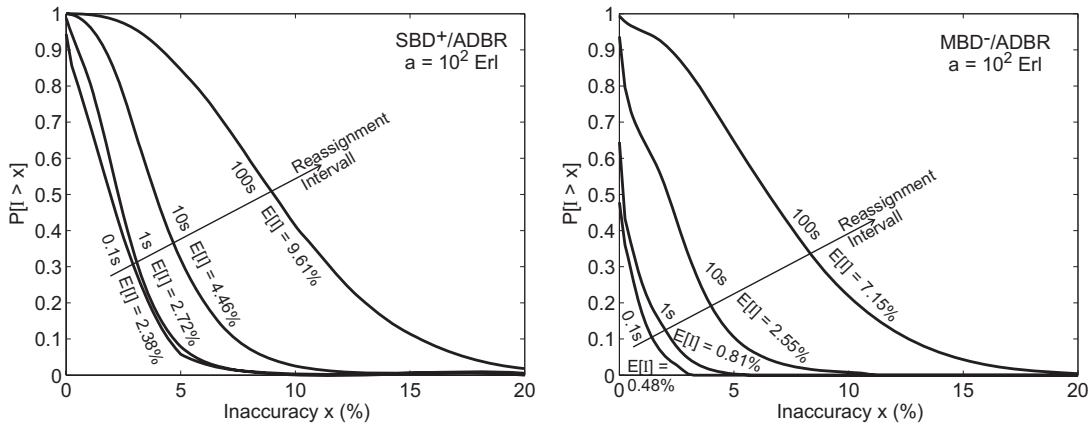


(a) $SBD^{-}/ADBR$ (b) $SBD^{+}/ADBR$

(c) $SBD^{+}/RDBR$ (d) $MBD^{-}/ADBR$

Figure 3.9: *Accuracy of load balancing over four links for various algorithms.*

The accuracy is quite similar for each link. However, links with small target load fractions rather tend to have positive load deviations while links with large target



(a) $SBD^{+}/ADBR$. (b) $MBD^{-}/ADBR$.

Figure 3.10: *Impact of the bin reassignment interval t_r on the load balancing accuracy.*

load fractions rather tend to have negative load deviations. The details for the other MBD versions are omitted here.

In the following we use $ADBR$-based algorithms because they are less complex and more accurate.

3.4.6 Impact of the Bin Reassignment Interval Length on the Accuracy and the Flow Reassignment Rate

The duration of the flows and the application frequency of dynamic reassignment steps have a significant impact on the load balanced results. In our simulations, the flow durations are exponentially distributed with a mean value of 90 s but we do not further elaborate on this issue since this is not a parameter under control. We rather investigate the load balancing accuracy depending on the reassignment interval length t_r.

Figure 3.10(a) shows the impact of t_r on the load balancing accuracy for

$SBD^+/ADBR$. The complementary distribution functions of the inaccuracy are similar for $t_r = 0.1$ s and $t_r = 1$ s with mean values of $E[I] = 2.38\%$ and 2.72%. The accuracy is clearly degraded for $t_r = 10$ s ($E[I] = 4.46\%$) and it is not acceptable for $t_r = 100$ s ($E[I] = 9.61\%$). We get similar results for $MBD^-/ADBR$ in Figure 3.10(b). The inaccuracy for $t_r = 100$ s is not acceptable ($E[I] = 7.15\%$) but the inaccuracy for $t_r = 10$ s ($E[I] = 2.55\%$) is comparable to the best accuracy of $SBD^+/ADBR$. The accuracy values for $t_r = 0.1$ s and $t_r = 1$ s are also similar, but with $E[I] = 0.48\%$ and 0.81% it is significantly better than for the corresponding values of $SBD^+/ADBR$.

The flow reassignment rate λ_{FR} introduced above is the average number of reassignments of a flow per second. If we multiply λ_{FR} with the lifetime of a given flow we get the number of reassignments this flow will perceive over its complete duration on average. The length of the bin reassignment interval t_r has a significant impact on the rate λ_{FR}. Figure 3.11 compiles the flow reassignment rates for $SBD^+/ADBR$ and $MBD^-/ADBR$. The flow reassignment rate increases for both algorithms by a factor of 10 if t_r decreases by a factor of 10 from 100 s to 10 s. We conclude that the same number of flows is reassigned whenever the load is balanced for $t_r \in \{10, 100\}$ s. A further reduction of t_r increases the reassignment rate significantly less. Hence, the number of reassigned flows within one step decreases.

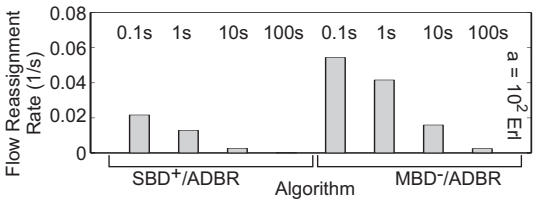


Figure 3.11: *Impact of the bin reassignment interval t_r on the flow reassignment rate.*

$SBD^+/ADBR$ and $MBD^-/ADBR$ achieve good load balancing results for $t_r = 0.1$ s and $t_r = 1$ s. However, for $t_r = 0.1$ s the flow reassignment rate is much higher. A similar accuracy can be obtained for $MBD^-/ADBR$

at $t_r = 10$ s and $SBD^+/ADBR$ at $t_r = 1$ s with about the same flow reassignment rate. After all, a bin reassignment interval length of 1 s should be chosen for both algorithms. Then, the flow reassignment rate is about 0.04 $\frac{1}{\text{s}}$ for $MBD^-/ADBR$ which means that a flow is reassigned every 25 s and that a flow with a duration of 90 s is reassigned 3.6 times on average. Note that packet reordering occurs less frequently because packets do not get necessarily out of order when flows are switched to other paths. The flow reassignment rate may be further reduced for MBD algorithms if the reconnection process tries to reconnect bins to their previous links if reasonable. This obviously already happens by chance but more intelligent algorithms can enforce this. Their complexity may be feasible for a bin reassignment interval length of $t_r = 1$ s such that this gives room for further research.

3.5 Multi-Stage Load Balancing

We now extend the single-stage performance evaluation to the multi-stage application in networks where polarization effects and interdependencies between decisions made at different stages occur. We first explain the polarization effect and its implications and then discuss the accuracy and dynamics of multi-stage load balancing. Since we use more complicated topologies for the different experiments here, we explain the topologies together with the experiments.

3.5.1 The Traffic Polarization Effect

In Figure 3.12 both router 11 and 21 use the same static load balancing algorithm without flow reassignments. Router 11 ideally splits the flows in half. Since the static load balancing depends only on the characteristic flow ID, the algorithms at both routers make the same decisions based on this ID. Every flow that is sent over the left interface by 11 is sent over the left interface by 21 as well since their IDs again produce the same hash values. Thus, the load balancing

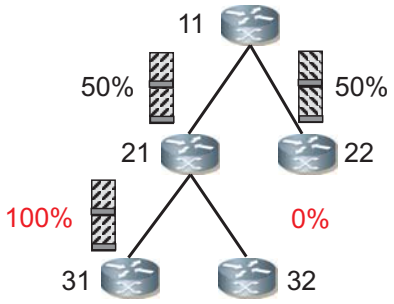


Figure 3.12: *Traffic polarization effect.*

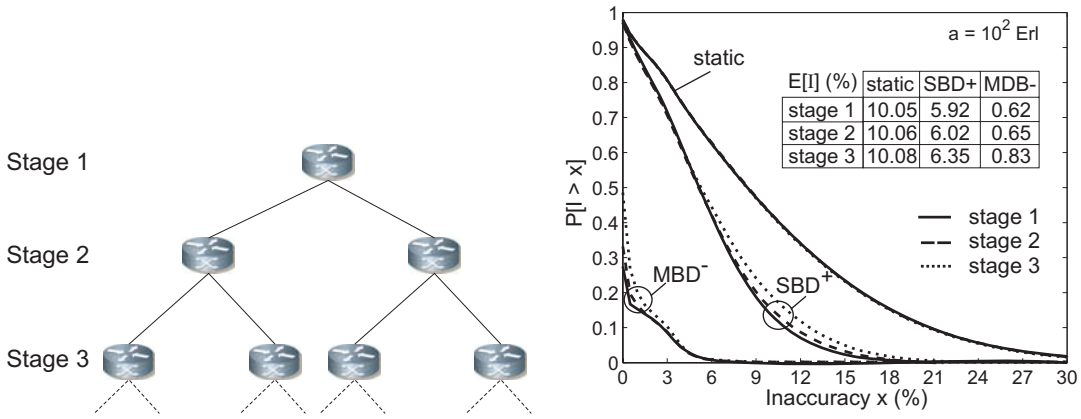
algorithm at router 21 is without effect. This phenomenon is called polarization effect similar to light passing through polarization filters [45]. Dynamic hashing alleviates this effect as it reassigns flows grouped in bins to other links. However, some bins remain empty and this leads to decreased load balancing granularity and to worse accuracy. To heal the polarization effect, a randomly generated ID can be assigned to every node in the network. Ideally, this ID is unique for every node and changes the output of the hash function such that the polarization effect vanishes completely. This modification of the input values to the hash function must be fast and retain the original potential of the load balancing mechanisms. We suggest a 32-bit random ID. There are many different possible operations to combine the random ID and the flow ID to a modified input value:

APP	Append random and flow ID
XOR	Combine last 32 bits of random and flow ID by bitwise-XOR
AND	Combine last 32 bits of random and flow ID by bitwise-AND
ADD	Perform integer addition between both IDs as binary numbers

So far anti-polarization mechanisms are proprietary and no information about influencing the hash function input values with the random ID are publicly available. In [45] Cisco suggests the use of an algorithmically generated ID which is not further specified.

3.5.2 Accuracy of Hash-Based Multi-Stage Load Balancing

We use the simple test scenario illustrated in Figure 3.13(a) to efficiently test the effect of the proposed modifications against polarization and to evaluate the accuracy of hash-based multi-stage load balancing. To assess the effectiveness of the modifications against polarization, we use it as a worst case scenario. All routers perform static hashing since it is most sensitive to traffic polarization. All routers at the lower stages obtain input from only one link with traffic that is possibly polarized. Finally, the link selector function simply decides to map even hash values to one link and odd hash values to the other link. Thus, there are no mechanisms for the compensation of polarization.



(a) Test topology. (b) Accuracy (target distribution: $50\%, 50\%$).

Figure 3.13: *Hash-based load balancing with anti-polarization mechanisms in networks.*

Ideally, the load is split in half at every router. As seen in Section 3.4.3, the offered load has a severe impact on the load balancing accuracy. For a fair comparison we require an offered load of $a = 10^2$ Erl at all stages where we observe the load balancing results. We achieve this by simulations where we feed the router at the first stage with 100, 200, or 400 Erl when we evaluate the load balancing accuracy on the first, second, or third stage.

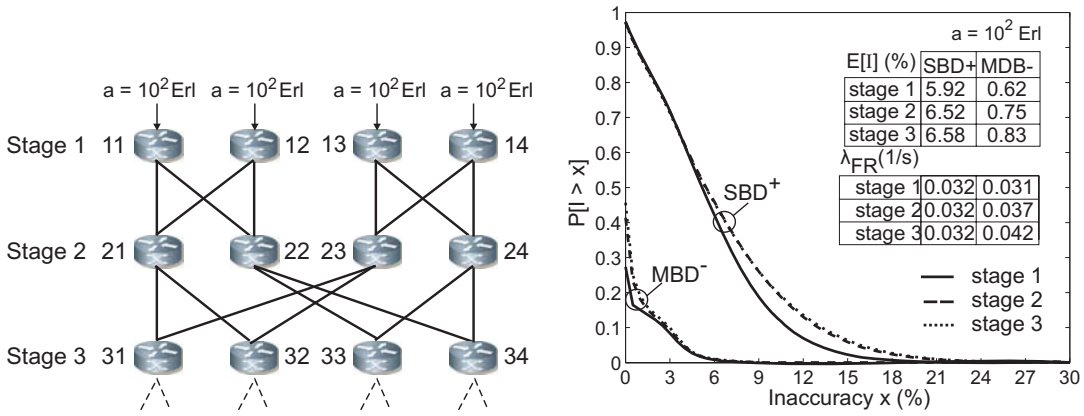
Figure 3.13(b) shows the complementary distribution function of the load balancing inaccuracy for the bitwise AND and the integer addition on the three different stages together with the mean inaccuracy $E[I] = 10\%$. We omit the results for appending the random ID (APP) and the XOR-operator as they have no effect against polarization. With both APP and XOR one link carries 100% of the traffic at stages 2 and 3. This can be explained by the mathematical properties of the used hash function CRC16. Basically, CRC16 interprets the flow ID as a polynomial over the field consisting of $\{0, 1\}$. The hash value is the residual of the polynomial division of the flow ID by a standardized generator polynomial. Thus, the hash is an element of the vector space of all polynomials of degree at most 16 over $\{0, 1\}$. It can be shown that both modifications are linear functions in this vector space and therefore have no effect on polarization.

The bitwise AND-operator and the integer addition, in contrast, cancel the polarization effect completely and retain the full load balancing potential of static hashing with $E[I] = 10\%$ at all stages as seen in Figure 3.13(b). These modifications can be interpreted as non-linear functions. Bitwise operations should be preferred as they can be easily computed in hardware. Thus, we choose the bitwise-AND operation to eliminate the polarization effect and use the modified input values in the following experiments if not mentioned otherwise.

Figure 3.13(b) also shows the inaccuracy at each stage if we use the dynamic algorithms SBD^+ and MBD^- instead. The load balancing inaccuracy for both algorithms increases slightly at each stage. Thus, even though the polarization vanishes completely as shown above, the dynamic algorithms suffer slightly from the reassignments made at other routers to which they can react after some delay only. However, the loss in accuracy is well acceptable.

3.5.3 Dynamics of Hash-Based Multi-Stage Load Balancing

To evaluate the dynamics of multi-stage load balancing in terms of flow reassignments, we use the more complex scenario shown in Figure 3.14(a). Flows



(a) Test topology.

(b) Dynamics (target distribution: $50\%, 50\%$).

Figure 3.14: *Dynamics of hash-based load balancing algorithms with anti-polarization mechanism in networks.*

arrive at the lower stages from two mutually disjoint paths. This models the dynamics caused by multiple independent load balancing entities as nodes in real networks receive traffic from multiple interfaces. At the same time, the symmetry of the scenario still keeps the complexity sufficiently low and we can observe the multi-stage dynamics without bothering with undesirable side effects. Besides, we configure the target load fraction $tLF(r, d, l) = 50\%$ for all routers r and their links $l \in \mathcal{L}(r, d)$. Hence, the routers are expected to receive an offered load of $a = 10^2$ Erl at all stages which does not require different simulation runs for the assessment of the load balancing accuracy at each stage as before. The flow reassignment rates $\lambda_{FR}(r, d)$ are measured locally for each router r. If – for instance – router 11 relocates a flow from the interface to node 21 to the interface to node 22, router 21 perceives this as the termination of the flow. If router 11 changes this assignment later and reroutes the flow to node 21, router 21 perceives this as the start of a new flow.

Figure 3.14(b) summarizes the results. The inaccuracy rises slightly from stage to stage for both dynamic algorithms. The gap between stage 1 and 2 is larger than in the previous experiment. This is due to the increased dynamics caused

by the input traffic from two independent dynamic load balancing entities. The reassignment rates for SBD^+ remain constant at $0.032\frac{1}{\text{s}}$ because the SBD^+ bin reassignment potential is limited since only one bin is relocated in each reassignment step. For MBD^- the rates increase slightly from stage 1 ($0.031\frac{1}{\text{s}}$) to stage 3 ($0.042\frac{1}{\text{s}}$) due to its larger potential to reassign bins. The increase is still well acceptable. However, for both concepts the overall end-to-end reassignment rate λ_{FR}^{e2e} for the flows routed over the three stages is the sum of the rates at the three stages. Thus, the end-to-end reassignment rate λ_{FR}^{e2e} increases linearly with the number of load balancing stages. Therefore, performing load balancing at too many stages is not recommended.

In addition to the results shown in Figure 3.14(b), we investigated the accuracy and dynamics of SBD^+ and MBD^- in the scenario of Figure 3.14(a) without anti-polarization mechanisms. The polarization effect leads to larger variations among the four different routers at the same stage than with anti-polarization mechanisms. For instance, the inaccuracy at stage 3 is in the range from $E[I] = 0.72\%$ to $E[I] = 0.94\%$ for the four different routers and the flow reassignment rate in the range from $\lambda_{FR} = 0.043\frac{1}{\text{s}}$ to $\lambda_{FR} = 0.050\frac{1}{\text{s}}$. Thus, polarization leads to performance degradation also in case of dynamic algorithms and the modifications against polarization should be used.

3.6 Summary: Accuracy and Dynamics of Hash-Based Load Balancing Algorithms

The term load balancing refers to a broad variety of application scenarios. In this chapter, we gave a brief overview of load balancing applications closely related to packet forwarding to identify similarities and differences to load balancing for multipath forwarding, the subject of our study.

Most state-of-the-art routers implement load balancing on the packet level or

on the flow level using hash-based mechanisms as alternatives. While load balancing mechanisms on the packet level in principle achieve the highest accuracy, they entail packet reordering and hence TCP throughput degradation. Consequently, our study focused on hash-based mechanisms, in particular the impact of dynamic load balancing with hash-based mechanisms.

We developed an evaluation methodology for hash-based load balancing algorithms that reveals the load balanced results over time. For our performance evaluation, we identified two fundamentally different scenarios: single-stage and multi-stage load balancing. The first one was used to demonstrate the basic properties of the algorithms under study, the second to examine interdependencies between individual nodes performing load balancing in a network.

In case of moderate aggregation level, static load balancing is not accurate enough. The deviation from the target value can be as high as 30%. Dynamic load balancing algorithms are needed. They use an intermediate data structure, so called bins, and change the assignment of the bins to outgoing links periodically to account for imbalances. However, the number of bins is an important parameter, that must be choosen high enough. 100 bins were sufficient and impose a still moderate implementation complexity. The distribution accuracy improves significantly if more than a single bin may be reassigned in a single load balancing step since this leads to more flexibility. Considering the dynamics caused by flow reassignments, we showed that a bin reassignment interval of 1 s is enough to achieve a good accuracy. In that case, flows are reassigned every 25 s to other paths which may cause packet reordering.

For multi-stage load balancing in networks, the simple application of the same load balancing algorithm in case of static load balancers cannot balance the traffic due to the polarization effect. We selected an efficient anti-polarization mechanism among some intuitive candidates and showed that suitable methods provide a general improvement of load balancing methods for their application in networks in terms of accuracy. Then, we investigated the flow reassignment rate in a complex multi-stage network architecture where load balanced traffic from different origins provides the input for the next load balancer. This does not degrade

the load balancing accuracy if anti-polarization mechanisms are used, but the overall flow reassignment rate increases approximately linearly with the number of load balancing steps.

After all, load balancing mechanisms should be carefully chosen to minimize the load balancing inaccuracy. Load balancing should also not be applied too often to the same set of flows since this increases the probability for route flaps and packet reordering.

4 Fast Resilience Concepts

In this chapter, we evaluate fast resilience concepts. First we discuss the standard path layout for both MPLS-FRR options in Section 4.1. Section 4.2 contains the MPLS-FRR performance study. There we analyze the required capacity for the standard path layout and suggest a simple enhancement that efficiently reduces the backup capacity requirements. Section 4.3 then describes the most favored IP-FRR mechanisms within the IETF routing working group (RTGWG) loop-free alternates (LFAs) and not-via addresses. This section provides a classification of different LFAs with respect to their ability and suggests options for their combination with not-via addresses regarding different resilience requirements. Based on this, Section 4.4 analyzes the effect of combining both IP-FRR mechanisms. Finally, Section 4.5 summarizes this chapter.

This chapter is based on basic principles described in Chapter 2, mainly in Section 2.4.

4.1 Mechanisms for MPLS Fast Reroute

MPLS fast reroute mechanisms protect primary LSPs by local repair methods. A primary LSP is said to be protected at a given hop if it has one or multiple associated backup tunnels originating at that hop. Here, we want to protect the primary LSP along all intermediate routers of its path. Thus, each intermediate router is a so-called point of local repair (PLR) that serves as head-end router for at least one backup path. In the following, we review both the one-to-one and the facility backup option of MPLS-FRR and explain the standard options for the layout of the backup path.

4.1.1 Local Repair Options in the MPLS Fast Reroute Framework

We briefly introduce the one-to-one backup and the facility backup together with mandatory conditions regarding the path layout for the protection of link or router failures.

One-to-One Backup Using Detour LSPs

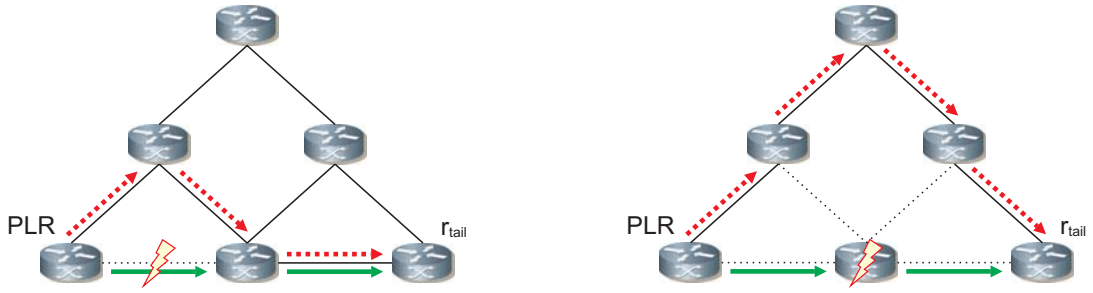
The one-to-one backup sets up a backup path from the PLR to the tail-end of the protected LSP. This backup path is called detour LSP. Each detour LSP protects exactly one primary LSP, but the primary LSP may be protected by several detour LSPs starting at different PLRs. If a detour LSP intersects its protected path further upstream, it may be merged with the primary path at a so-called detour merge point (DMP) to reduce the LSP states in the routers further downstream. However, we disregard this possibility in the following since we focus on the path layout and not on configuration details. Operational modes are defined in which detour LSPs may contain elements of the protected LSP and others are defined in which such elements are forbidden. In the following, we point out mandatory constraints to protect against link or router failures.

Link Detour To protect a primary path against a link failure, the router preceding the failed link acts as PLR by redirecting the traffic onto a detour LSP towards the tail-end router r_{tail} of the primary path. The backup path must not contain the failed link, but it may contain the adjacent routers of the failed link. We call this type of backup path $LinkDetour(PLR, r_{tail})$.

Router Detour To protect a primary path against a router failure, the router preceding the failed router acts as PLR by redirecting the traffic onto a detour LSP towards the tail-end router r_{tail} of the primary path. The backup path must not contain the failed router and all its adjacent links. We call this type of backup path

$RouterDetour(PLR, r_{tail})$. Note that the primary path cannot be protected against the failure of its head-end or tail-end label switched router (LSR).

The paths $LinkDetour(PLR, r_{tail})$ and $RouterDetour(PLR, r_{tail})$ from the same PLR within the same flow can take different shortest paths due to their specific requirements as shown in Figures 4.1(a) and 4.1(b).



(a) $LinkDetour(PLR, r_{tail})$. (b) $RouterDetour(PLR, r_{tail})$.

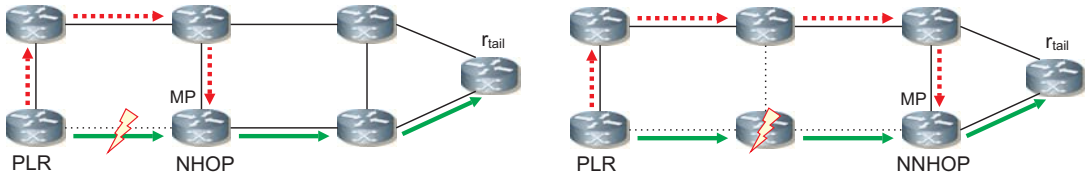
Figure 4.1: *One-to-one backup option using detours to protect link and router failures.*

Facility Backup Using Bypass LSPs

The facility backup sets up a backup path from the PLR to a downstream router of the protected LSP. This router is called merge point (MP) as it merges the backup path with the protected LSP. Since the backup path bypasses the failure location, it is called bypass LSP. Unlike detour LSPs, a bypass LSP can protect multiple primary LSPs that share the same PLR and MP. In the following, we describe the placement of the MP for the protection against link or router failures.

Link Bypass To protect a primary path against a link failure, the router preceding the failed link acts as PLR by redirecting the traffic onto a bypass LSP towards the next hop (NHOP) LSR of the PLR. Thus, the adjacent routers of the link are the head-end and the tail-end LSRs of the bypass

LSP which must not contain the failed link. We call this type of backup path $LinkBypass(PLR, NHOP)$.



(a) $LinkBypass(PLR, NHOP)$. (b) $RouterBypass(PLR, NNHOP)$.

Figure 4.2: *Facility backup using bypasses.*

Router Bypass To protect a primary path against a router failure, the router preceding the failed router acts as PLR by redirecting the traffic onto a bypass LSP towards the next-next hop (NNHOP) LSR of the PLR. Thus, the neighboring routers of the failed router within the primary path are the head-end and the tail-end LSRs of the bypass LSP which must not contain the failed router and all its adjacent links. We call this type of backup path $RouterBypass(PLR, NNHOP)$. Like above, the primary path cannot be protected against the failure of its head-end or tail-end LSR.

The $LinkBypass(PLR, NHOP)$ and $RouterBypass(PLR, NNHOP)$ in Figures 4.2(a) and 4.2(b) from the same PLR within the same flow take different paths due to their specific requirements.

4.1.2 Backup Path Configuration

An intuitive standard approach is characterized by setting up backup LSPs according to the shortest path principle [143]. Each potential PLR, i.e. each intermediate LSR of a primary LSP, needs a backup path for the protection against the failure of the next link and the next router, respectively. We now assess the number of required backup paths. We assume n routers and m bidirectional links in the network as well as a fully meshed LSP overlay, i.e., there are $n \cdot (n-1)$

protected LSPs. The length of a specific primary path p is given by $len(p)$ in terms of links and the average number of links per primary path is denoted by $\overline{len}$. The number of adjacent links of router r is given by its node degree $deg(r)$. The average node degree in a network is $deg_{avg} = \frac{2 \cdot m}{n}$

Number of Required Detour LSPs If the one-to-one backup concept uses separate backup paths for the protection against link and router failures within LSP p, it requires $len(p)$ link detour LSPs to protect against all link failures and $len(p)-1$ router detour LSPs to protect against all router failures of the primary path. Thus, $2 \cdot len(p) - 1$ detour LSPs are required altogether for the protection of p. As a consequence, $n \cdot (n-1) \cdot (2 \cdot \overline{len} - 1)$ detours are needed in the network. The authors of [143] suggest that a link failure can be protected by a $LinkDetour(PLR, r_{tail})$, but it can also be protected by the $RouterDetour(PLR, r_{tail})$. The latter simply has stricter requirements for the layout of its backup paths. Such backup paths exist for all links except the last one within the primary path. Thus, $len-1$ link failures can be protected by a $RouterDetour(PLR, r_{tail})$ and the failure of the last link must be protected by a $LinkDetour(PLR, r_{tail})$. This reduces the number of detours in the network to $n \cdot (n-1) \cdot \overline{len}$ and is the proposed standard path layout for the one-to-one backup concept.

Number of Required Bypass LSPs The network requires $2 \cdot m$ unidirectional link bypasses to protect against the failures of m different links since these backup LSPs can protect multiple primary paths. In addition, router bypass LSPs are needed for the protection against the failure of each of the n routers. We consider a specific router r with $d = deg(r)$ adjacent bidirectional links over which traffic can be received from and forwarded to its neighbors. If all combinations are possible, i.e., each neighbor serves both as PLR and NNHOP for a primary LSP carried over r, $d \cdot (d-1)$ different router bypasses are needed to

protect against the failure of this router. As a consequence, a rough guess for the number of required backup paths is

$$\begin{aligned} 2 \cdot m + n \cdot deg_{avg} \cdot (deg_{avg} - 1) &= 2 \cdot m + 2 \cdot m \cdot (deg_{avg} - 1) = \\ 2 \cdot m \cdot deg_{avg} &= n \cdot deg_{avg}^2. \end{aligned}$$

This expression proposes that considerably fewer bypasses than detours are required to protect the network against all single link and router failures.

4.2 MPLS-FRR Performance Study

In this section we investigate the performance of the above discussed options for MPLS-FRR by parametric studies regarding different network characteristics. First, we explain our evaluation method, then we study the required backup capacity and the number of backup paths per primary path before we compare their efficiency with other well known resilience mechanisms.

4.2.1 Evaluation Method

We explain the network dimensioning approach used to calculate the required backup capacity. We also describe the foundation of our parametric study which is based on artificially generated random networks.

Calculation of the Required Backup Capacity

The required backup capacity is the major performance measure in this study. We obtain it as follows for a given network topology, a given traffic matrix, and a given resilience mechanism. The network topology is given by a graph $\mathcal{N} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of routers and $\mathcal{E}$ is the set of links. We first compute the capacity $c(l)$ of all links $l \in \mathcal{E}$ in the network that is required to carry the traffic according to the shortest path principle. The sum of these capacities yields

the required network capacity $C_\emptyset = \sum_{l \in \mathcal{E}} c(l)$ for the failure-free scenario $\emptyset$. The network must be protected against the failures of a set of failure scenarios $\mathcal{S}$ that contains always the failure-free scenario $\emptyset$. Resilience mechanisms require sufficient backup capacity on the links to carry the traffic in each protected failure scenario. We first determine the link capacity $c(s,l)$ that is required to carry the traffic in each protected failure scenario $s \in \mathcal{S}$ according to the routing applied by the resilience mechanism during the failure. All backup paths follow the respective shortest path that does not contain the failed element. We use the link capacity $c(s,l)$ to calculate the required capacity for the resilient network by $C_\mathcal{S} = \sum_{l \in \mathcal{E}} max_{s \in \mathcal{S}}(c(s,l))$. Note that traffic aggregates are inactive in failure scenarios if their source or destination node fails. We express the required backup capacity relative to the capacity needed for shortest path routing by $B = \frac{C_\mathcal{S} - C_\emptyset}{C_\emptyset}$. This method can be viewed as a network dimensioning approach. It grants the capacity to the links where it is needed by the considered resilience mechanism. An alternative option is to calculate, e.g., blocking or QoS violation probabilities for networks with given link capacities.

Parametric Study

In our parametric study, we assume that every network node serves as border router with transit capabilities, i.e., all nodes are origin and destination of traffic and forward transit traffic. We assume a fully meshed overlay network and a homogenous traffic matrix. [238] showes that the heterogeneity of the traffic matrix has a significant impact on the required backup capacity, but an investigation of this issue in this context is beyond the scope of this work. We consider three different failure scenarios: all single router failures, all single bidirectional link failures, and all single router and bidirectional link failures (cf. Section 2.4.1).

We use sample networks in our study to examine a broad range of different network characteristics. The most important characteristics for resilient networks are the network size in terms of nodes $|\mathcal{V}| = n$ and in terms of links $|\mathcal{E}| = m$. They define the average node degree $deg_{avg} = \frac{2 \cdot m}{n}$ that indicates the average number

of adjacent links of the nodes and is thereby an indirect measure for the network connectivity. In addition, the minimum and the maximum node degree deg_{min} and deg_{max} are also important measures. Since today's well established topology generators cannot control deg_{min} and deg_{max}, we use our own topology generator which is described in [147] and which incorporates features of the well known Waxman model [239, 240]. It allows direct control over n, deg_{avg}, and the maximum deviation deg_{dev}^{max} of the individual node degrees from their predefined average value. It generates connected networks and avoids loops and parallels. We consider networks of size $n \in \{10, 15, 20, 25, 30, 35, 40\}$ nodes with an average node degree $deg_{avg} \in \{3, 4, 5, 6\}$ and a maximum deviation from the average node degree of $deg_{dev}^{max} \in \{1, 2, 3\}$. We generate 5 networks of each combination randomly. This sums to 420 sample networks. For each of them we calculate the required backup capacity for each of the following resilience mechanisms.

I The set of protected failure scenarios comprises all single link failures; $LinkDetour(PLR, r_{tail})$ and $LinkBypass(PLR, NHOP)$, respectively, are used to achieve link protection *(LP detour/bypass)*.

II The set of protected failure scenarios comprises all single router failures; $RouterDetour(PLR, r_{tail})$ and $RouterBypass(PLR, NNHOP)$, respectively, are used to provide router protection *(RP detour/bypass)*.

III The set of protected failure scenarios comprises both all single link and all single router failures; to achieve link and router protection, $LinkDetour(PLR, r_{tail})$ and $RouterDetour(PLR, r_{tail})$ or $LinkBypass(PLR, NHOP)$ and $RouterBypass(PLR, NNHOP)$ are used for the link and router failures, respectively *(LRP detour/bypass)*.

IV As an alternative to the previous scenario, we substitute the link backup paths through existing router backup paths wherever possible *(LRP-SL detour/bypass)*.

In the following these abbreviations indicate the protected failures and the applied method. Note that LRP-SL Detour and LRP Bypass are the standard approaches proposed in [143].

4.2.2 Backup Capacity Requirements

We compare the backup capacity requirements for the eight protection options defined above. The protection option also determines the set of protected failure scenarios the networks are dimensioned for. Each point in Figure 4.3 represents the average backup capacity for all 60 networks of a specific size and the respective considered failures scenarios. The chosen resilience mechanism has a significant impact on the required backup capacity.

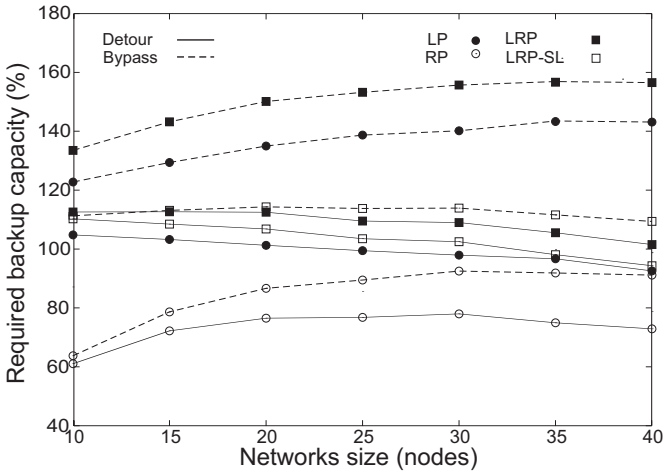
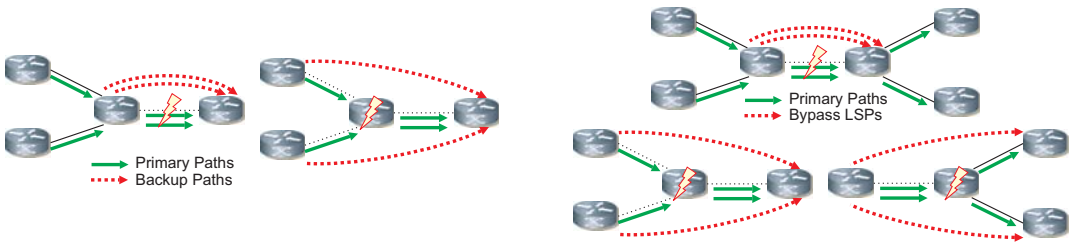


Figure 4.3: *Impact of the network size, the protected failures, and the resilience mechanism on the required backup capacity for MPLS-FRR.*

We first compare the capacity requirements for the four resilience mechanisms from above link protection (LP), router protection (RP), link and router protection where link and router failures are protected by separate backup paths (LRP), and link and router protection where router backup paths also protect link failures if applicable (LRP-SL) without differentiation between the one-to-one and the facility backup option where possible. This makes it easy to understand the reasons for the different capacity requirements of both concepts afterwards. For both the one-to-one and the facility backup, plain router protection (RP) requires the least resources and in particular less resources than plain link protection (LP). This is at first glance counterintuitive since router failures af-

fect also several adjacent links, but there are two reasons that explain the phenomenon. First, inactive aggregates whose source or destination failed decrease



(a) In case of router failures, the one-to-one backup concept deviates the traffic from different locations in the network.

(b) In case of router failures, the facility backup concept deviates the traffic from and also to different locations in the network.

Figure 4.4: *The Traffic distribution for the one-to-one and facility backup concept during link and router failures.*

the amount of traffic in the network. However, this affects only $\frac{2}{n}$ of the entire traffic and, therefore, this effect shrinks with an increasing network size and makes the curves for RP increase significantly from small to medium size networks in Figure 4.3. Another reason for the reduced backup capacity requirements of RP compared to LP is the improved traffic distribution around the outage location. This is illustrated in Figure 4.4(a). In case of the one-to-one backup concept, the $LinkDetour(PLR, r_{tail})$ backup paths for LP have a single point of local repair (PLR) while the $RouterDetour(PLR, r_{tail})$ backup paths have different PLRs. Thus, the traffic is deviated over a larger number of different links starting from different locations in the network. As a consequence, the required backup capacity is distributed over a larger number of different links in the network which increases the potential for backup capacity sharing for independent scenarios. In case of the facility backup concept, the $LinkBypass(PLR, NHOP)$ backup paths of LP have a single point of local repair (PLR) and a single merge point (MP) at the next hop (NHOP) while the $RouterBypass(PLR, NNHOP)$ backup paths have different PLRs or different MPs at the next next hop (NNHOP) as illustrated in Figure 4.4(b). This leads to an even stronger diversification of the

rerouted traffic and, therefore, the gap between RP Bypass and LP Bypass is larger than for the detour concept.

LRP uses the backup paths of both RP and LP and requires clearly more capacity than their maximum. Hence, LP allocates its capacity at different locations compared to RP. As a consequence, the substitution of the $LinkDetour(PLR, r_{tail})$ backup paths through suitable $RouterDetour(PLR, r_{tail})$ backup paths for the one-to-one concept and the substitution of the $LinkBypass(PLR, r_{tail})$ backup paths through suitable $RouterBypass(PLR, r_{tail})$ where possible (LRP-SL) leads to a notable reduction of required backup capacity in Figure 4.3. However, the last links of the primary paths cannot be protected by suitable router backup paths. This backup capacity reduction technique is very effective for the facility backup. Since one $LinkBypass(PLR, NHOP)$ is used by many primary LSPs, the subsitution of link bypasses through router bypasses increases the traffic spreading and reveals an enormous capacity savings potential.

In all considered investigation scenarios, the facility backup concept always requires more capacity than the one-to-one backup concept. This is clearly due to the reduced capacity sharing potential: it uses a single path to carry the traffic of many affected primary LSPs whose traffic is spread over many links by detour LSPs.

4.2.3 Configuration Overhead: Number of Backup Paths

As mentioned above, resilience mechanisms differ regarding their configuration overhead. The number of backup paths that must be configured contributes to the number of connection states in the network. Therefore, we compare this measure for the investigated scenarios.

Figure 4.5 shows the average number of backup LSPs per primary LSP depending on the network size. For the one-to-one backup concept (detour), the number of backup paths scales with the average path length in the network.

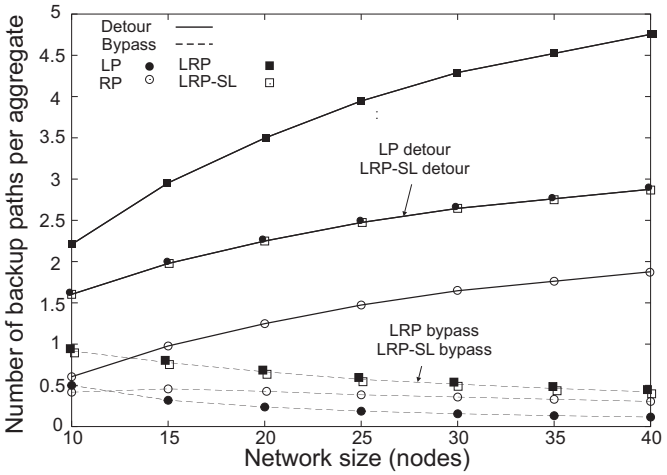


Figure 4.5: *Impact of the network size, the protected failures, and the protection method on the configuration overhead.*

The number of backup paths for LP is exactly the average path length (cf. Section 4.1.2). The number of intermediate routers along a path is smaller by one than the number of links and, thus, the number of backup paths for RP smaller by exactly one than for LP. LRP uses all link detours from LP as well as all router detours from RP and, therefore, its number of backup paths is their sum. LRP-SL uses all router detours from RP to substitute appropriate link detours in LP. This leads to a protection of link and node failures while keeping the number of backup paths as low as for LP.

For the facility backup concept (bypass), the number of backup paths for LP is exactly the number of links $2{\cdot}m$. This yields an average per aggregate of $\frac{2 \cdot m}{n \cdot (n-1)}$ and decreases with the network size. We approximated the number of backup paths for RP as $n \cdot deg_{avg}^2$. The results of our evaluation in Figure 4.5 reveal that this is only an upper bound. Not all routers serve as transit nodes and not all

combinations are actually used to transport transit traffic over the adjacent links of a node. This effect is extremely strong for small networks where many aggregates are direct connections between neighboring nodes. The number of backup paths for LRP is the sum of LP and RP as before. LRP-SL substitutes link bypasses with router bypasses for all links within a primary LSP except for the last one. Since each link is at least once the last link within the primary LSP that consists of exactly one link connecting neighboring nodes, this does not reduce the number of required backup paths.

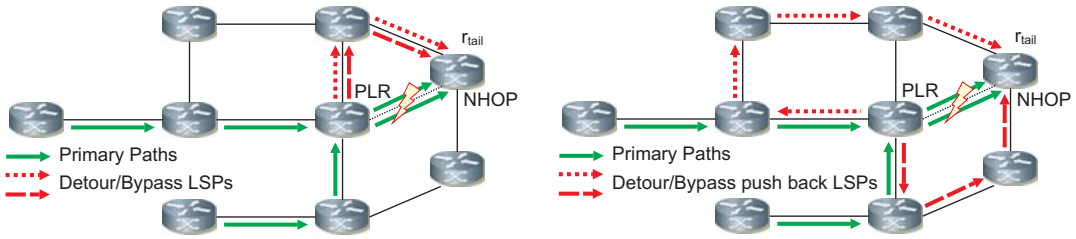
The facility backup clearly requires less backup paths than the one-to-one backup since one bypass can protect several primary LSPs. While LRP requires less than one bypass per primary LSP for the facility concept, it requires almost $2 - 5$ detours per primary LSP for the one-to-one concept.

4.2.4 A Simple Mechanism for Increasing the Traffic Spreading

We showed above that the substitution of link backups with suitable router backups where possible (LRP-SL) leads to a notable reduction of the required backup capacity. This is due to an improved traffic spreading and, hence, increased capacity sharing potential in the network. However, the failure of the last link of the primary path cannot be protected by suitable router backup paths. This leaves room for further improvement and motivates the following simple push back mechanism for the last link: the idea is to deviate the traffic one hop prior to the outage location.

In detail, the idea is based on the following observation. When a router fails, its neighbor routers act as PLRs and redirect the traffic onto different backup paths. Hence, the traffic can be spread well across the network. When a link fails, there are several LSPs for which this link is the last link. For all those LSPs, only a single PLR redirects the traffic over the same link backup path (cf. Figure 4.6(a)). Pushing the traffic back by one link to the previous router within the primary path and deviating it from there leads to the same situation like for router failures: the

traffic can be distributed from different locations (cf. Figure 4.6(b)). This leads to the following rules for the one-to-one and facility backup options.



(a) $LinkDetour(PLR, r_{tail})$/ $LinkBypass(PLR, NHOP)$

(b) $PushBackDetour(PLR, r_{tail})$/ $PushBackBypass(PLR, NHOP)$

Figure 4.6: *Application of the push back concept to detour and bypass LSPs. Since NHOP and r_{tail} are identical, detours and bypasses follow the same path.*

Push Back Mechanism for Detour and Bypass LSPs

When the last link fails and the primary path contains only a single link, a normal $LinkDetour(PLR, r_{tail})$ or $LinkBypass(PLR, NHOP)$, respectively, is the only option. Otherwise, we push the traffic back to the previous router within the primary path such that it can be deviated from different locations to the tail-end router r_{tail} or the next hop router NHOP. We call this new kind of backup paths $PushBackDetour(PLR, r_{tail})$ and $PushBackBypass(PLR, NHOP)$, respectively. Note that these backup paths start at the PLR and visit the previous router before heading to r_{tail} or NHOP. Since the NHOP router and the tail-end router r_{tail} are identical in case of the failure of the last link, both the detour and the bypass follow the same path.

Our newly proposed path structures $PushBackDetour(PLR, r_{tail})$ and $PushBackBypass(PLR, r_{tail})$ are shown in Figure 4.6(b). They can be applied for the protection against the failure of the last link of an LSP. The push back mechanism leads to the following additional resilience mechanism.

V The set of protected failure scenarios comprises both all single link and all single router failures; to achieve link and router protection, the following backup structures are applied:

- LSPs that consist of only one link are protected against the failure of their single link by $LinkDetour(PLR, r_{tail})$ or $LinkBypass(PLR, NHOP)$, respectively.
- The last link of LSPs longer than one link are protected against the failure of their last link by $PushBackDetour(PLR, r_{tail})$ or $PushBackBypass(PLR, NHOP)$, respectively.
- All other link and router failures of the primary LSPs are protected by their corresponding $RouterDetour(PLR, r_{tail})$ or $RouterBypass(PLR, NNHOP)$, respectively.

(LRP-PB detour/bypass)

Backup Capacity Requirements and Configuration Overhead

We evaluate the backup capacity requirements for the push back mechanism in Figure 4.7. The application of the push back mechanism for the protection of the last link (LRP-PB) reduces the required backup capacity additionally between 11 to 22% for the one-to-one backup option and between 16 to 22% for the facility backup option relative to LRP-SL. The improvment decreases with increasing network size for both backup concepts. This is all due to the improved traffic spreading of the push back paths. Remarkably, LRP-PB detour, LRP-PB bypass, and LRP-SL bypass require clearly less capacity than LP detour/bypass where only link failures are protected (cf. Figure 4.3).

Concerning the configuration overhead in terms of the number of backup paths, the push back mechanism does not increase this overhead for the one-to-one backup relative to LRP-SL. Since the backup paths are LSP-specific, our mechanism simply replaces the LSP-specific $LinkDetour(PLR, r_{tail})$ with another LSP-specific $PushBackDetour(PLR, r_{tail})$ where possible. This is different for the facility backup concept. Here, all bypasses protect a specific net-

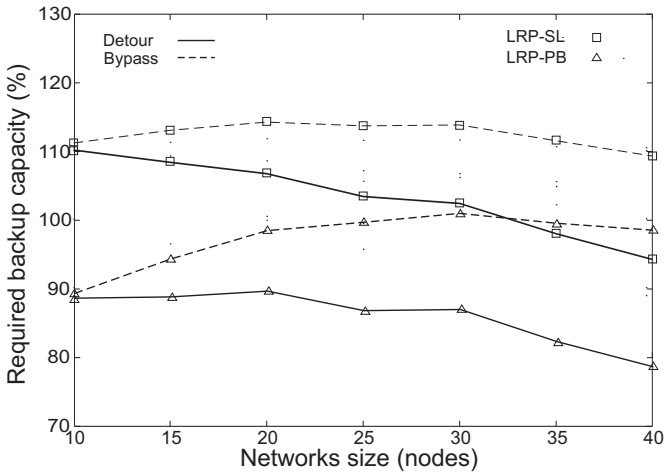


Figure 4.7: *Impact of the network size on the required backup capacity for the push back mechanism.*

work element and not a specific LSP. Thus, the LRP-PM bypass uses two bypasses for each link in the network to protect against its failure. One link bypass for the protection of LSPs that consist of one link only and one push back bypass for the protection of LSPs that consist of more than one link. This leads to $0.11-0.24$ additional backup LSPs per primary LSP relative to LRP-SL bypass and is still well below the configuration overhead for the one-to-one concept. Exact numbers can be found in [8].

4.2.5 Additional Performance Measures

The backup capacity requirements depending on the network connectivity and the average prolongation of the path length through backup LSPs are additional performance measures of interest. A detailed analysis for the one-to-one backup option can be found in [13] and for the facility backup option in [8]. The network connectivity in terms of the average node degree reduces the required backup capacity significantly, in particular for the mechanisms that enforce backup capacity sharing like LRP-SL and LRP-PB.

Regarding the average path prolongation, the facility backup performs worse

than the one-to-one backup since it sends the traffic around the failure location and not directly to the destination. Still, this value does not exceed two hops for LRP-PB bypass that uses the longest bypass structures. Interestingly, small networks see shorter prolongation values. In these networks it becomes more likely to find bypasses and detours of the same length as the original bypassed part or path to the destination, respectively.

4.2.6 Comparison of the Required Backup Capacity for Restoration, End-to-End Protection, and Local Protection

We consider the backup capacity requirements for the self-protecting multipath (SPM) [9, 18, 147], which is an efficient end-to-end protection scheme, for shortest path re-routing (SPR) and equal-cost multipath (ECMP) re-routing, which are the most basic restoration schemes, and for both MPLS-FRR concepts. All mechanisms were configured to protect all single link and node failures.

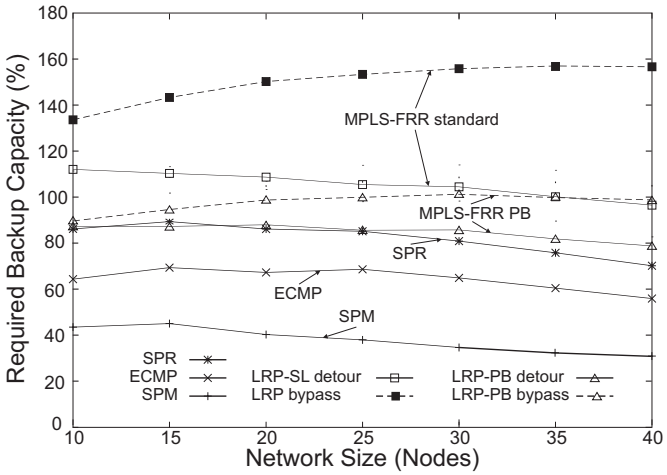


Figure 4.8: *Impact of the network size and the resilience mechanism (restoration, e2e protection, and local protection) on the required backup capacity.*

Figure 4.8 shows the averages of their required backup capacities depending

on the network size. For MPLS-FRR we show the results of LRP-SL detour and LRP bypass, the proposed standard path layout for the one-to-one and facility backup options, respectively, and LRP-PB detour/bypass, the most efficient backup path layout considered in this study. The SPM requires by far the least capacity, followed by ECMP and SPR. LRP-SL detour requires $21 - 26\%$ more capacity than SPR. LRP bypass requires $47 - 86\%$ more capacity than SPR. However, MPLS-FRR reacts within tens of milliseconds whereas SPR is a simple restoration mechanism and usually reacts only within seconds [132–134]. The reduced configuration overhead of the facility backup concept comes at the expense of additional required capacity if LRP is used as proposed in the standard. Here, LRP-PB helps to reduce the capacity requirements to almost the same level as for the one-to-one backup concept while keeping the configuration overhead low. The SPM seems to be the most attractive resilience mechanism since it requires the least capacity and it is relatively fast as it implements end-to-end protection. However, in contrast to MPLS-FRR, it needs load balancing capabilities (cf. Chapter 3) and it is not a standardized approach.

4.3 Mechanisms for IP Fast Reroute: Loop-free Alternates and Not-via Addresses

In this section, we describe the IP-FRR mechanisms loop-free alternates (LFAs) and not-via addresses in detail. We focus on these two mechanisms in our analysis since they are the most favored ones within the IETF routing working group (RTGWG). LFAs are simple as they avoid tunnels and they potentially lead to shorter detours, but they cannot protect all single failures. Some LFAs are able to protect only link failures, others protect also router failures. Some lead to routing loops in case of multiple failures, others are safe. Not-via addresses are more complex as new prefixes need to be distributed via routing protocols. They require tunnel-

ing which is undesirable as en-/decapsulation potentially reduces the forwarding speed of the routers and might lead to packet fragmentation due to MTU limitations. However, not-via addresses offer 100% failure coverage. The combination of both mechanisms is an option expected to merge their advantages. This issue is in the focus of our study. We also give recommendations for their combination regarding different resilience requirements.

4.3.1 Classification of Loop-Free Alternates

In this section, we review the definition of LFAs, we classify them according to their ability, and establish a new taxonomy.

Definition of LFAs

A loop free alternate (LFA) is a local alternative path from a source node S towards a destination node D in the event of a failure [139]. If S cannot reach its primary next hop P towards D anymore, it simply sends the traffic to another neighbor N that still can forward the traffic to D avoiding both the failed element and S and thus does not create routing loops. LFAs are pre-computed and installed in the forwarding information base (FIB) of a router for each destination. The Internet draft [139] specifies three criteria for LFAs that guarantee different levels of protection quality and loop avoidance. We illustrate these conditions and provide a taxonomy for the classification of neighbor nodes with respect to their ability to be used as LFAs. For compactness sake, in the following LFA refers both to the neighbor providing the local alternative path and the path itself. The context clarifies the respective meaning.

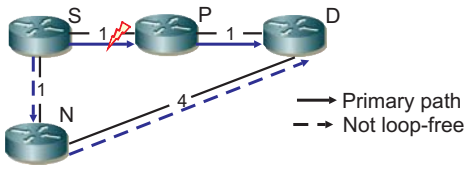
Loop-Free Condition (LFC)

We consider source S and destination D in Figure 4.9(a). The numbers associated with the links are the link metrics taken into account for shortest path routing. When link $S \rightarrow P$ fails, packets can only be rerouted over neighbor N. However,

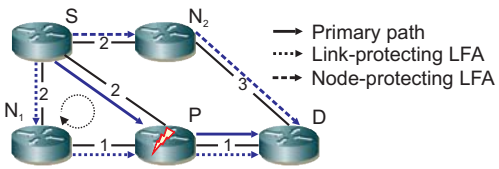
this creates a forwarding loop because the shortest path of N to D leads back to S. Therefore, N cannot be used as LFA by S to protect the failure of link $S \rightarrow P$. To avoid loops, the following loop-free condition (LFC) must be met:

$$\text{dist}(N, D) < \text{dist}(N, S) + \text{dist}(S, D). \tag{4.1}$$

In Figure 4.9(b), both neighbors N_1 and N_2 of source S fulfill this condition with regard to destination D.



(a) The neighbor N cannot be used as LFA because it does not meet the loop-free condition (LFC).



(b) Only the node-protecting LFA N_2 can be used to protect against the failure of node P.

Figure 4.9: *Illustration of the loop-free condition (LFC) and the node-protection condition (NPC) for LFAs.*

Node-Protection Condition (NPC)

We consider the failure of node P in Figure 4.9(b). When traffic is rerouted to neighbor N_1, the next hop is again P, the traffic is rerouted to S, and a routing loop occurs. Therefore, N_1 cannot be used as LFA by S to protect the failure of node P. However, N_2 can be used for that objective. A neighbor node N must meet the following node-protection condition (NPC) to protect against the failure of a node P:

$$\text{dist}(N, D) < \text{dist}(N, P) + \text{dist}(P, D) \tag{4.2}$$

An LFA meeting the LFC only is called link-protecting while an LFA also meeting the NPC is called node-protecting. Since the NPC implies the LFC[1], every node-protecting LFA is also link-protecting, but not vice-versa.

Downstream Condition (DSC)

We consider source S and destination D in Figure 4.10(a). N provides a node-protecting LFA for S. If two nodes P_S and P_N fail simultaneously, S reroutes its traffic to N. N cannot forward the traffic, either, and reroutes the traffic to S which is a node-protecting LFA for N in that case. Thus, a routing loop occurs. Such loops, which are due to multi-failures, can be avoided if an LFA obeys the downstream condition (DSC):

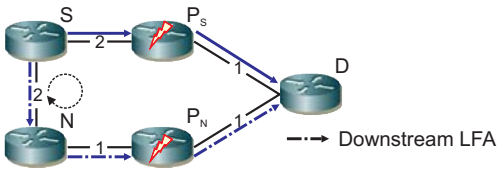
$$\text{dist}(N, D) < \text{dist}(S, D) \tag{4.3}$$

An LFA fulfilling this condition is called downstream LFA. Allowing only downstream LFAs guarantees loop avoidance for all possible failures because packets get always closer to the destination. In this case, N can be used as LFA for S in Figure 4.10(a) but not vice-versa which avoids the routing loop in our example. N must use another neighbor – if available – to protect against the failure of P_N.

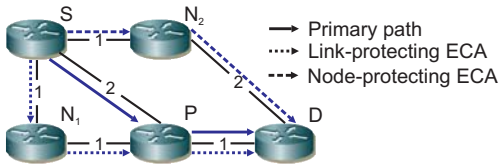
Equal-Cost Alternates (ECAs)

The IP-FRR framework [138] classifies equal-cost multipaths (ECMPs) as the most basic IP-FRR concept (cf. Section 2.4.5). Actually, ECMPs can also be seen as LFAs in the following sense. A special case of LFAs are equal-cost alternates (ECAs), i.e., alternative next hops such that the alternative path is not longer than the primary path. An example is depicted in Figure 4.10(b). Source S knows several paths of equal cost towards D. If its next hop P fails, it can use any of the

[1] $\text{dist}(N, D) <^{\text{NPC}} \text{dist}(N, P) + \text{dist}(P, D) \leq^{\text{(a)}} \text{dist}(N, S) + \text{dist}(S, P) + \text{dist}(P, D) =^{\text{(b)}} \text{dist}(N, S) + \text{dist}(S, D)$ – (a) follows from the triangular equation, (b) holds since the shortest path from S to D leads via P.



(a) Neighbor N provides a downstream LFA for S but not vice-versa. The use of only downstream LFAs avoids loops in the presence of multiple failures.



(b) The equal-cost alternates (ECAs) N_1 and N_2 provide alternate paths of the same length as the primary path. N_1 is just link-protecting while N_2 is node-protecting.

Figure 4.10: *Illustration of the downstream condition (DSC) and equal-cost alternates (ECAs).*

remaining equal-cost paths as LFA that do not contain the failed element. Thus, either N_1 or N_2 may be used as ECA and even both may be used at the same time. In particular, if the standard routing uses the ECMP option, the traffic hit by the failure is equally redistributed over the remaining paths. In the following, ECA refers both to the neighbor providing the local alternative path and the path itself.

It is easy to see that ECAs cannot create loops in case of multiple failures as they are always downstream LFAs. They are link-protecting but not necessarily node-protecting (see N_1 in Figure 4.10(b)). This also shows that downstream LFAs are not necessarily node-protecting.

Taxonomy of LFAs

The above conditions limit the number of neighbor nodes as potential LFAs and create thereby sets of neighbors with different ability to protect against failures and loops.

ECAs are always downstream LFAs (DSC). Downstream LFAs are always loop-free (LFC). Some neighbor nodes do not meet any of the corresponding conditions. Thus, the set of ECAs is contained in the set of downstream LFAs which is part of the set of general LFAs which are a subset of all neighbor nodes. This relation is depicted in Figure 4.11.

The NPC to guarantee node-protecting LFAs is orthogonal to the other conditions: both neighbor nodes in Figure 4.10(b) are ECAs, but only N_2 is node-protecting. N_1 in Figure 4.9(b) and N in Figure 4.10(a) are both downstream LFAs, but only N is node-protecting. N_2 in Figure 4.9(b) is a non-downstream LFA and node protecting while N in Figure 4.9(a) does not meet any condition. Examples for non-downstream non-node-protecting LFAs can also be constructed.

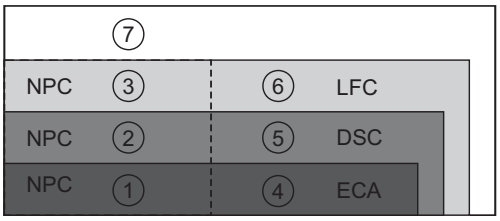


Figure 4.11: *Classification of neighbor nodes with regard to their ability as forwarding alternates to protect failures and to prevent loops.*

The Venn diagram in Figure 4.11 partitions the set of neighbor nodes into 7 different categories. We order them according to a possible preference for their usage as LFAs (the ultimate preference is the network operator's decision [139]):

1 node-protecting ECAs
2 node-protecting downstream LFAs
3 node-protecting LFAs that do not fulfill the downstream condition
4 ECAs that are just link-protecting
5 downstream-LFAs that are just link-protecting
6 LFAs that are just link-protecting and do not fulfill the DSC.

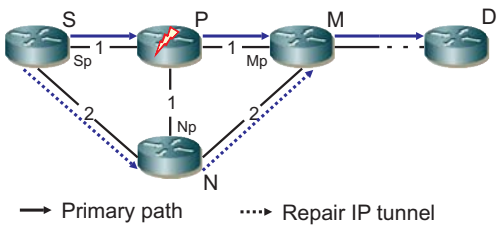
Neighbors not meeting any of the conditions 7 cannot be used as LFAs as they create routing loops.

LFAs cannot achieve 100% failure coverage [189–191]. However, they can be complemented by other IP-FRR mechanisms with a larger failure coverage.

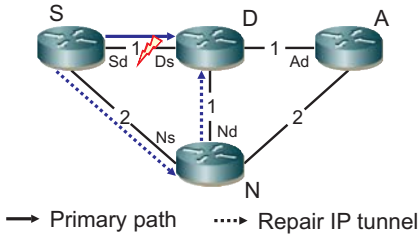
4.3.2 IP Fast Reroute Using Not-Via Addresses

The intention of this approach is to protect the failure of a node P or of its adjacent links by deviating affected traffic around P to the next-next hop (NNHOP) M using IP-in-IP tunneling. The path of this tunnel must not contain the failed node P. This is usually not achievable with normal IP forwarding since P is on the shortest path from S to M. Thus, special "not-via addresses" Mp are introduced such that packets addressed to Mp are forwarded to M not via P. Although the basic idea of IP-FRR using not-via addresses is tunneling to the the NNHOP, it is also possible to protect the last link of a path where no NNHOP exists.

Figure 4.12(a) illustrates this concept for the case that a NNHOP exists on the primary path. Node S must forward a packet destined to D, but the next hop (NHOP) P (or next link $S \rightarrow P$) fails. Then S encapsulates this packet in another IP packet addressed to the NNHOP using the not-via address Mp. This packet is forwarded from S over N to M which is the shortest path around node P. NNHOP M performs decapsulation and forwards the original packet to D.



(a) NNHOP exists: encapsulation to address Mp; the encapsulated packet is carried to M not via P.



(b) NHOP is destination: encapsulation to address Ds; the encapsulated packet is forwarded to one of its neighbors and then carried to D not via S.

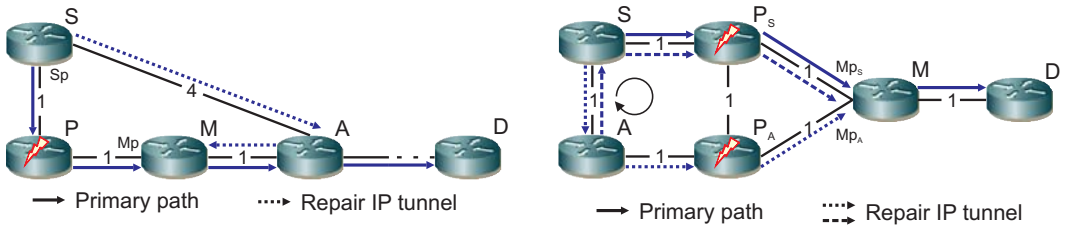
Figure 4.12: *Use of not-via addresses to protect the failure of intermediate nodes and links, and the last link.*

Figure 4.12(b) shows how not-via addresses can be used in case the NHOP D is already the destination. In contrast to above, node S assumes that only the next link instead of the NHOP has failed; otherwise, the packet cannot be delivered anyway. It encapsulates the packet and addresses it towards Ds. The semantic

of Ds at node S is that the direct link $S \rightarrow D$ must not be used. Therefore, the forwarding table at S provides another interface for forwarding the packet to another neighbor that passes it towards D. Since the packet is sent to Ds, it cannot loop back to S. Finally, D decapsulates the packet and the original packet has reached its destination. If indeed not only link $S \rightarrow D$ but node D has failed, the packet is discarded as soon as it reaches another neighbor of D.

IP-FRR using not-via addresses guarantees 100% failure coverage for single node and link failures unless the network gets physically disconnected by such a failure, i.e., the failed element is an articulation point. The concept is very similar to the MPLS-FRR facility backup option installing local bypasses to every NNHOP [8]. However, the backup paths in MPLS may follow explicit routes, therefore, MPLS-FRR has more degrees of freedom than IP-FRR using not-via addresses.

In the example of Figure 4.13(a), packets are carried from S to D over P, M, and A. If P fails, these packets are tunneled to Mp such that they take the path S, A, M, A, D which is unnecessarily long and wastes capacity, but does not create a loop.



(a) Unnecessarily long backup paths occur if the tunnel from S to the NNHOP M intersects with the downstream paths from M to D.

(b) Routing loops could occur if packets were recursively tunneled to not-via addresses in case of multiple failures.

Figure 4.13: *With not-via addresses, unnecessarily long backup paths may occur; recursive tunneling is prohibited to avoid routing loops.*

In Figure 4.13(b), S cannot deliver packets to D if nodes P_S and P_A fail simultaneously. In that case, S encapsulates packets to D in packets destined to Mp_S and these packets are carried to A. A cannot forward the packets to M

because P_A also fails. If A encapsulates them to the not-via address Mp_A and returns them to S, a routing loop occurs. Therefore, recursive tunneling using not-via addresses is prohibited [140].

IP-FRR using not-via addresses requires the network to provide additional entries in the forwarding tables for not-via addresses. Not-via addresses have the form Mp where p can be any node and M can be any of its neighbors. Therefore, the number of not-via addresses equals the number of unidirectional links in the network. The forwarding entries for the not-via addresses can be constructed by distributed routing algorithms [140].

4.3.3 Comparison of LFAs, Not-Via Addresses, and their Combined Usage

In this section, we compare the properties of LFAs and not-via addresses and discuss how both approaches may be combined.

Pros and Cons of LFAs and Not-Via Addresses

Tunneling Not-via addresses fully rely on IP tunneling. This involves en- and decapsulation of tunneled packets and may have a performance impact on router hardware. Further, it leads to increased packet lengths inside the tunnel and may result in packet fragmentation due to maximum transmission unit (MTU) limitations. Encapsulation applies a different re-write string to the front of the packet and most current hardware achieves this without performance degradation. Packet decapsulation at the tunnel endpoint, however, requires two lookup operations. The first to recognize the tunnel endpoint, the second for further forwarding with the inner IP address. Most modern hardware is designed to perform this at line rate. On legacy hardware this can slow down the handling of tunneled packets to almost half the line rate. So the major disadvantage caused by tunneling stems from packet decapsulation on legacy hardware.

Backup Path Length LFAs may allow slightly shorter repair paths. While LFAs are computed per destination prefix and deviate the packets directly to the destination, not-via addresses deviate the traffic around the failure back onto the original path.

Computational Routing Complexity In principle, each node must remove every other node P one by one from the base topology and perform a shortest path tree (SPT) computation in this reduced topology to the not-via addresses Np of P's neighbors N. Incremental SPT (iSPT) computations reduce this effort that is proportional to the number of nodes in the network to an equivalent of 5 to 13 SPT computations in real world networks with 40 to 400 nodes [140]. ECAs in particular are very easy to compute since they are obtained for free from the usual shortest path calculations. For general LFAs, the computational cost of determining individual repair paths for all destinations can be very high as well. Hence, the computational routing complexity and its assessment is hardware- and implementation-dependent.

Failure Coverage If there are no articulation points that disconnect the network in case of a failure, not-via addresses always achieve 100% failure coverage using a single resilience concept. This is usually impossible for LFAs [189–191].

Compatibility with Loop-Free Re-Convergence Schemes The computation of the not-via tunnels can be temporally decoupled from the computation of the basic routing. Thus, during routing re-convergence, the tunnels remain stable making not-via addresses compatible with additional mechanisms for loop-free re-convergence [180, 241]. This does not necessarily hold for LFAs since the re-convergence process may render LFA conditions invalid.

Protection of Multicast Traffic Not-via addresses deviate the traffic to the NNHOP through tunnels. Thus, the NNHOP can infer the usual interface from the not-via address and run the reverse path forwarding (RPF) check required for

multicast traffic correctly [140]. Protection of multicast traffic with LFAs seems complex and is currently not under discussion.

Adaptability to SRLGs The functionality of not-via addresses can be easily adapted to SRLGs. If SRLGs are known, the SPT computation for the respective not-via address is simply performed in the topology with all elements from the SRLG removed. This is much more complicated for LFAs.

Backup path lengths and in particular the potential problems involved in tunneling may favor the combined usage of both concepts, the remaining points advocate not-via addresses. In the following we provide further insights into this discussion.

Combined Usage of LFAs and Not-Via Routing for Different Resilience Requirements

In this work, we study three options for IP-FRR with a different level of failure protection and loop avoidance:

- **i** Protection against all single link failures
- **ii** Protection against all single link and all single router failures
- **iii** Protection against all single link and all single router failures with loop avoidance in the presence of multiple failures

Not-via addresses fulfill the strictest resilience requirement (iii). LFAs alone cannot even meet the loosest one because they cannot achieve 100% failure coverage. Therefore, we complement them by not-via re-routing where necessary. As LFAs have different properties (cf. Figure 4.11), only certain LFA types can be used in the above cases in the following order of preference:

- **i** (1), (4), (2), (5), (3), (6), and not-via.
- **ii** (1), (2), (3), and not-via; (4), (5), and not-via to protect the last link.
- **iii** (1), (2), and not-via; (4), (5), and not-via to protect the last link.

Note that LFAs that are only link-protecting (6) cannot be used for the protection of the last link for (ii) and (iii) since they may create loops when the destination node is down.

4.4 IP-FRR Performance Study: LFAs and Not-Via Addresses

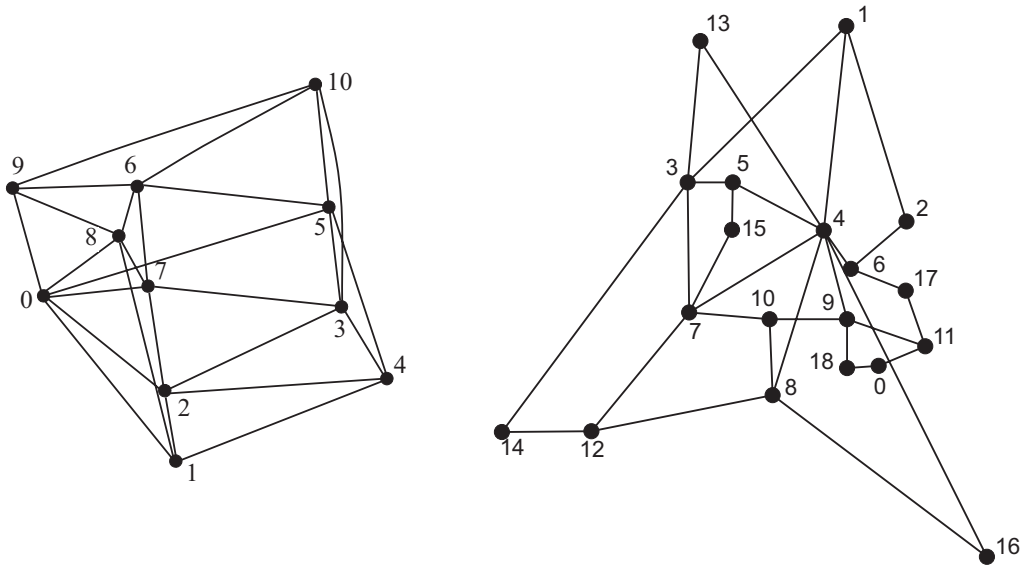
For the above resilience requirements, we analyze the combined applicability of LFAs and not-via addresses, the backup path prolongation, and the amount of decapsulated traffic in an experimental environment.

4.4.1 Experimental Environment

We use well-known realistic networks for our experimental environment to examine the mechanisms under study for their applicability in practice: COST239, GEANT, Labnet03, and NOBEL. We only present the results from COST239 (see Figure 4.14(a)) and from GEANT (see Figure 4.14(b)) here, since the other networks do not yield additional insights. The COST239 and the GEANT network are typical representatives of two different network types. For Labnet03 and Nobel there are quantitative, but no qualitative differences.

Even for real networks, traffic matrices are generally unavailable due to confidentiality reasons. Thus, we use the method proposed in [242] and enhanced in [243] for generating synthetic traffic matrices resembling real-world data. Note that traffic matrix traces are indeed available for the GEANT network, but we used the synthetically generated traffic matrices here as well to assure comparability.

We set all link weights to one and perform simple hop count routing as often used in unoptimized networks. We perform single shortest path first (SPF) routing. When multiple equal cost paths towards a destination are available, the



(a) COST239 (b) GEANT

Figure 4.14: *Networks under study.*

interface with the lowest ID is used as the active interface as specified for IS-IS [49].

We scaled the traffic matrices such that the maximum link utilization does not exceed 100% for SPF re-convergence and any of the considered failure scenarios.

4.4.2 Applicability of LFAs and Not-Vias

We first study the applicability of LFAs and not-vias at the individual network nodes. Figures 4.15 – 4.16 show the percentage of the destinations protected by different types of LFAs and not-vias for the 11 nodes in the COST239 and the 19 nodes of the GEANT network and resilience requirements (i) to (iii). The x-axes show the node IDs and the y-axes the percentage of destinations at a node covered by the respective mechanism in percent. We applied appropriate LFAs and not-via protection according to the recommendations in Section 4.3.3. Since there is a slightly different semantic (cf. Section 4.3.2) for not-via addresses for

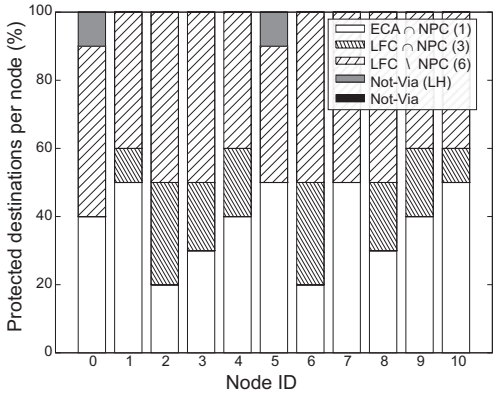
the last hop, we indicate not-vias used for the protection of the last hop (LH) towards a destination separately.

We start with general observations. In networks using simple hop count routing, only three out of six types of neighbors (cf. Figure 4.11) providing LFAs exist. First, ECAs that are only link-protecting (4) do not exist since there are no parallel links. Second, there are no downstream LFAs (2),(5). The downstream criterion requires that the alternate neighbor N is closer to the destination D than the deviating node S. Since the distance dist(S, N) from S to its neighbor N is always 1, this can only be true for equal cost paths.

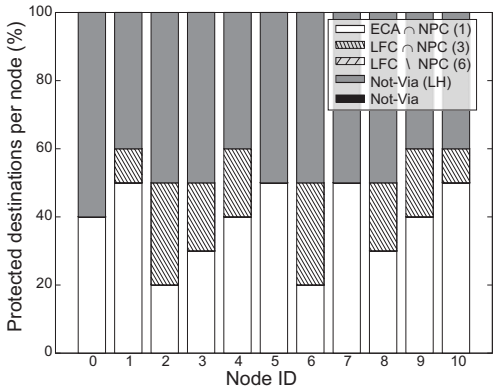
We now discuss the results from the COST239 network. The COST239 topology represents a class of networks that are well connected among the individual nodes. For most nodes any other node is reachable within at most two hops. In Figure 4.15(a) corresponding to resilience requirement (i) – link protection only – almost all destinations can be protected using LFAs. ECAs (1) protect between 20 to 50% of the destinations and node-protecting LFAs (3) vary from 0 to 30%. Link-protecting LFAs (6) are applicable for a high percentage of destinations between 40 to 50 %, mainly to protect the last hops of the relatively short paths. Almost no not-vias are necessary. Only two nodes require about 10% of not-vias for the last hop.

Figure 4.15(b) shows the results for the stricter resilience requirement (ii) – link and node protection. All link-protecting LFAs (6) are replaced with not-vias. For the strictest resilience requirement (iii) – link and node protection with general loop avoidance – shown in Figure 4.15(c), node-protecting LFAs (3) are not sufficient anymore and are again replaced by not-vias. Now, only ECAs and not-vias are applicable due to the non-existence of downstream LFAs.

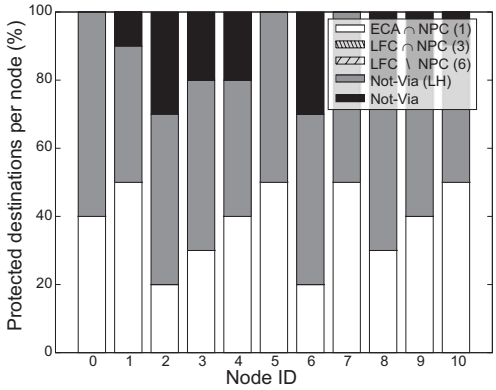
The GEANT topology, in contrast, represents a more sparsely connected class of network topologies. The paths between node pairs are significantly longer since the nodes lie on circles of three to five nodes. Concerning the results, the variation between the individual nodes is high. In Figure 4.16(a) for resilience requirement (i), node 16 is very different from the other nodes. It uses 100% link-protecting LFAs (6). This can be explained by its special location forming



(a) Link protection only.

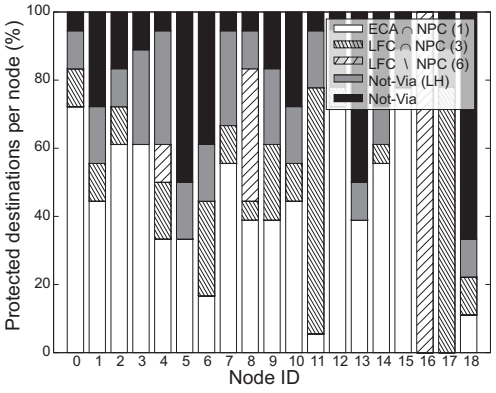


(b) Link and node protection.

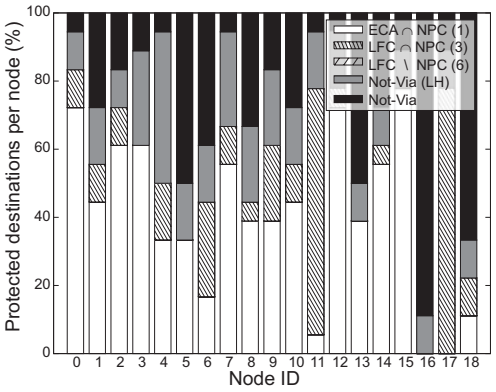


(c) Link and node protection - no loops during multiple failures.

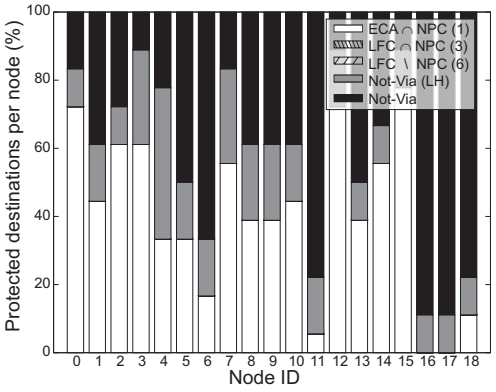
Figure 4.15: *Applicability of LFAs and not-via addresses in the COST239 network with different resilience requirements.*



(a) Link protection only.



(b) Link and node protection.



(c) Link and node protection - no loops during multiple failures.

Figure 4.16: *Applicability of LFAs and not-via addresses in the GEANT network with different resilience requirements.*

a triangle with nodes 4 and 8. Besides node 16, only two other nodes use these LFAs (6) while the number of node-protecting LFAs (3) varies greatly between 0 and almost 80%. In contrast to COST239, all nodes except for node 16, require not-vias for the protection of the last hops, and up to 70% of all destinations within a node's routing table can only be protected using not-via addresses.

For resilience requirement (ii) in Figure 4.16(b), again all link-protecting LFAs (6) cannot be used anymore. Consequently, node 16 requires 100% not-vias. For the strictest resilience requirement (iii) in Figure 4.16(c), again only ECAs (1) and not-vias are applicable. Now node 16 and 17 require 100% not-vias.

The conclusion from this analysis is threefold.

(a) In case of simple hop count routing three out of six types of LFAs do not exist.

(b) If loop avoidance in general failure cases is required (resilience requirement (iii)), LFAs other than ECAs cannot be used in networks that use simple hop count routing

(c) Average values for the coverage achieved by LFAs as shown in previous work are not sufficient performance measures: the existence of suitable LFAs largely depends on the network topology and in certain topologies LFAs cannot protect a single destination prefix at individual nodes under resilience requirements (ii) and (iii). The average values hide these variations. Hence, not-vias are not only necessary as an additional FRR mechanism for LFAs to achieve 100% failure coverage, for some nodes they are the only option.

4.4.3 Path Prolongation

A backup path should not be much longer than the original path for delay sensitive applications. Hence, we assess the path prolongation for all failure scenarios. Figure 4.17 shows the CCDF for the path prolongation for resilience requirements (i) and (iii) in the GEANT network. The x-axes shows the path prolongation x in number of hops, the y-axes shows the conditional probability that a path

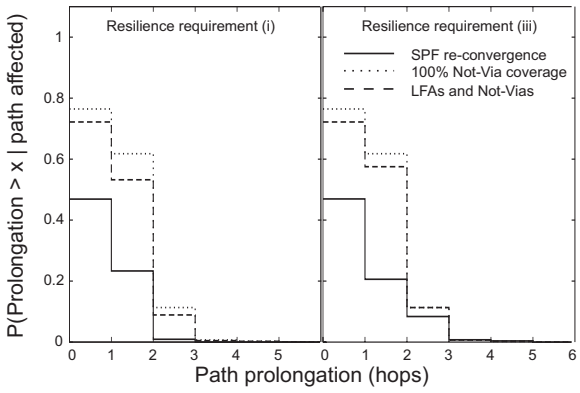


Figure 4.17: *Path prolongation in the GEANT network for resilience requirements (i) and (iii).*

affected by a failure increases by more than x hops. SPF re-convergence is the comparison baseline since the backup path cannot be shorter.

The length of about 50% of the paths does not increase for plain IP re-convergence. These are the paths where alternative paths of equal length exist between source and destination. This value decreases to around 25% if IP-FRR is applied since fewer ECAs are available for local repair at intermediate nodes. The difference between IP-FRR and SPF re-convergence is well noticable, however, the difference between 100% not-via coverage and the combination of LFAs with not-vias is small and well tolerable. This difference even decreases for the strictest resilience requirement (iii).

We omit the values for the COST239 network since there is no difference between both IP-FRR mechanisms, the difference between SPF and IP-FRR is similar to GEANT.

4.4.4 Decapsulated Traffic from Not-Via Tunnels

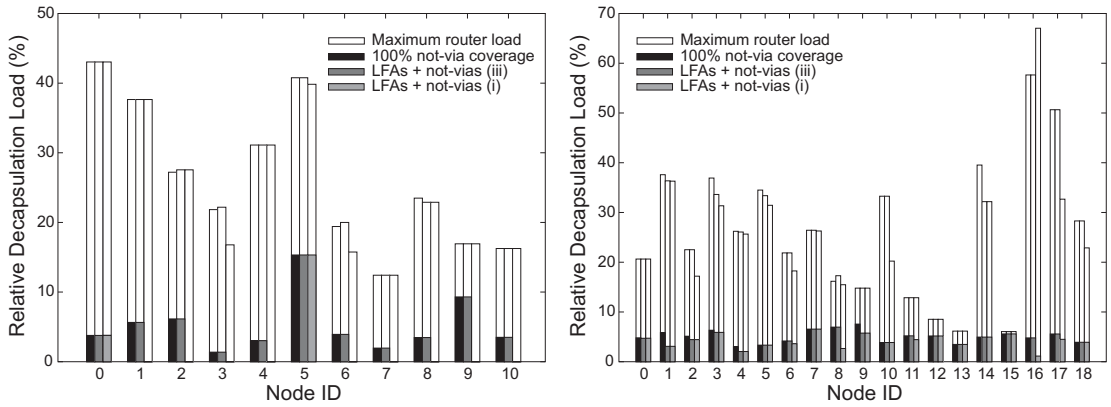
In Figures 4.18(a) and 4.18(b), we analyze the amount of traffic that must be decapsulated at the not-via tunnel endpoints. All numbers for the individual nodes are relative to the node capacity, which is the sum of the capacity of the incom-

ing interfaces of the node. Our performance metric is the maximum amount of decapsulated traffic observed in all protected failure scenarios. The bars in the background show the maximum amount of incoming traffic, i.e., the maximum router load, to relate the results to the overall traffic at a node. Note that the maximum router load is well below 100% since the load reaches its maximum for individual incoming links in different scenarios.

In the COST239 network (Figure 4.18(a)) with 100% not-via-based failure coverage, almost all nodes must decapsulate at most traffic equivalent to well below 10% of their capacity. Only node 5 shows a higher value of 15%. Surprisingly, there is no reduction of the maximum amount of decapsulated traffic with the combined usage for resilience requirement (iii). This does not mean that the deployment of LFAs does not reduce the amount of decapsulated traffic at all, however, the maximum over all failure scenarios cannot be reduced here. For combined coverage and resilience requirement (i), only nodes 0 and 5 still decapsulate packets. These are the only two nodes that require not-vias to protect 100% of their destinations. Interestingly, node 0 tunnels packets to node 5 and vice versa. This phenenomen is due to the network structure. While all other pairs of neighboring nodes form triangles with a third node allowing for the use of a link-protecting LFA, for nodes 0 and 5 only a quadrangle can be found. Again, the maximum amount of decapsulated traffic does not decrease at those two nodes.

The results are slightly different in the GEANT network (Figure 4.18(b)). The maximum values for 100% not-via coverage stay well below 8% of the node capacities. For combined usage and resilience requirement (iii), the maximum amount of decapsulated traffic reduces for one half of the nodes, but most nodes show only small differences. For resilience requirement (i), a further reduction is noticable for individual nodes, especially nodes 8 and 16, but all nodes must still decapsulate traffic.

In general, the combined usage of LFAs and not-vias does not reduce the maximum amount of decapsulated traffic significantly. In particular, if more than pure link protection is required.



(a) COST239 (b) GEANT

Figure 4.18: *Amount of decapsulated traffic per node relative to maximum node capacity for COST239 and GEANT.*

4.5 Summary: Recommendations for Fast Resilience

Given the presence of component failures in modern communication networks and the requirements of new emerging services for stringent service availability and reliability, MPLS- and IP-FRR techniques seem to be a promising answer. They provide a fast since local reaction.

MPLS-FRR has already been standardized and implemented by many commercial router vendors. The standards describe only signaling issues and the behavior of the label switched routers (LSRs) in case of network element failures. They do not recommend the layout of the backup paths themselves. These backup paths should be short, easy to configure, easy to calculate, and they should require only little additional backup capacity when backup capacity sharing is possible. Many operators use the shortest path that avoids the outage location for the backup path layout and, thus, intuitively achieve the first three of these requirements.

In this chapter, we presented a parametric study of the capacity requirements

of the one-to-one and facility backup options for MPLS-FRR. Our results show that the facility backup option in conjunction with the LRP path layout proposed as standard for bypasses [143] requires more backup capacity than the one-to-one option in conjunction with the LRP-SL path layout proposed as standard for detours [143]. This is due to a better distribution of the detoured traffic in failure cases achieved by LRP-SL. However, the configuration overhead of the facility backup concept is clearly smaller. If the LRP-SL path layout is also used for the facility option, the backup capacity requirements become nearly as low as for the one-to-one concept. Since the configuration overhead remains low, this is advisable.

Based on the results from our evaluation of the standard mechanisms, we proposed an additional simple mechanism that further increases the traffic spreading in the network, the so-called pushback mechanisms LRP-PB. These mechanisms decrease the backup capacity requirements additionally by up to 20%. MPLS-FRR remains the most expensive resilience concept in terms of capacity relative to e2e protection switching (SPM) and IP re-routing, but it is extremely fast and LRP-PB reduces the cost almost to the level of single shortest path re-routing (SPR). If MPLS-FRR is applied, the bypass concept should be preferred over the one-to-one option where possible. Its configuration overhead is clearly lower while the capacity requirements are tollerably higher.

IP-FRR is still under development but the first support in commercial products supporting is emerging on the market. In this chapter we studied the combined usage of two important IP-FRR mechanisms currently under standardization by the IETF: loop-free alternates (LFAs) and not-via addresses. In case of failures, LFAs deviate traffic to neighboring nodes providing an alternate path towards the destination that avoids the failed element and does not create loops. Not-via addresses bypass the failed element with local IP-in-IP tunnels.

We classified different sets of neighbors providing LFAs according to their ability and established a new taxonomy for LFAs. This taxonomy suggests an order of preferred combinations of LFAs and not-vias for three types of resilience requirements described here. Based on this, we presented a performance analysis

to answer the question whether the combination of both mechanisms is beneficial.

LFAs alone cannot achieve 100% failure coverage and must be complemented by other IP-FRR mechanisms like not-vias. Our analysis of their combined usage revealed that three out of six types of LFAs do not exist in networks using simple hop count routing. If single link and node failures should be protected, at least 50% of all destinations of a node require not-via protection on average. Depending on the network topology, the variation between individual nodes can be very high, leading to nodes for which not a single destination can be protected without not-via addresses.

IP-FRR mechanisms lead to longer backup paths than plain IP re-convergence. The combined usage of LFAs and not-via addresses leads only to slightly shorter backup paths than 100% not-via coverage. The same holds for the maximum amount of decapsulated traffic caused by not-via tunneling. The combined usage cannot reduce this amount significantly. There is a price to pay in terms of resource requirements for the deployment of IP-FRR mechanisms relative to plain IP re-convergence, but there is no difference between 100% not-via protection and the combined deployment of LFAs and not-vias.

These findings support the following recommendation. If 100% failure coverage with IP-FRR is required, not-via addresses should be applied as the only FRR mechanism since our results do not show convincing advantages of the combined application. A homogeneous solution also leads to a simpler network management.

5 Dimensioning of Resilient Networks

In this chapter, we evaluate dimensioning techniques for resilient networks. First we describe our basis for a fair comparison in Section 5.1. Section 5.2 presents the capacity requirements for capacity overprovisioning (CO) and admission control (AC) on a single link. This provides a good understanding of the dimensioning methods that later are applied in the network context. In Section 5.3 we develop our resilient capacity dimensioning framework for CO and AC in networks and investigate the impact of traffic shifts and redirected traffic on the required capacity for CO and AC in networks. Section 5.4 summarizes this chapter.

This chapter is based on basic principles described in Chapter 2, mainly in Section 2.5.

5.1 Basis for a Fair Comparison

Most CO studies use both a flow and a packet level model. The first models the number of active flows of a traffic aggregate in the network whereas the second produces the required extra bandwidth above the mean data rate of the traffic.

5.1.1 Packet Level Model

Above, we mentioned effective bandwidths which depend on the queuing behavior of the underlying packet level model. As we are primarily interested in the

comparison of CO and AC regarding the resource efficiency in networks, we assume the same packet level model in both network types, which leads to the same required bandwidth, i.e. effective bandwidth, for a flow with both mechanisms. An inadequate packet level model leads to QoS degradation in both systems. This consideration eliminates the uncertainty of the packet level model and, thus, yields the basis for a fair comparison.

5.1.2 Flow Level Model

We consider networks with real-time flows. Such a setting may be found in the DiffServ architecture (cf. Section 2.1) when we focus only on the bandwidth for high priority traffic. Real-time flows are mostly triggered by human beings. Thus, their inter-arrival time is exponentially distributed [214]. The Poisson model for flow arrivals is also advocated by [233, 244, 245] and evidence of Poisson inter-arrivals for VoIP calls is given in [234]. Therefore, we use a flow level model that is characterized by an exponentially distributed inter-arrival time and a general, independently and identically distributed call holding time. The offered load to a system is its average number of simultaneous flows if no flow blocking occurs due to AC. It is measured in the pseudo unit "Erlang" and it is calculated by $a = \frac{\lambda}{\mu}$ where λ is the arrival rate and $\frac{1}{\mu}$ the mean holding time of the flows.

5.1.3 Traffic Mix

The author of [147] suggests a simplified multirate model with $n_r = 3$ different bit rate types r_0, r_1, and r_2 with a bit rate of $c(r_0) = 64$ kbit/s, $c(r_1) = 256$ kbit/s, and $c(r_2) = 2048$ kbit/s. The random variable R_t indicates the requested rate in case of a flow arrival. Its distribution in Table 5.1 is parameterized such that the mean rate of the flows $E(c(R_t)) = 256$ kbit/s is independent of the parameter $t \in [0, 1]$ and that the coefficient of variation of their rate $c_{var}(c(R_t)) = 2.291 \cdot t_R$ depends linearly on it. Under the assumption that the flows of all rate types have the same mean holding time the rate-specific offered load can be calculated by $a_i = a \cdot p(r_i)$.

request type r_i	$c(r_i)$	$p(r_i)$
r_0	64 kbit/s	$\frac{28}{31} \cdot t^2$
r_1	256 kbit/s	$(1-t^2)$
r_2	2048 kbit/s	$\frac{3}{31} \cdot t^2$

Table 5.1: *Distribution of the flow rate R_t (effective bandwidth) depending on the parameter $t \in [0,1]$.*

5.1.4 Capacity Dimensioning for AC and CO on a Single Link

Capacity dimensioning on a single link with a multi-rate Poisson flow model and the usage of effective bandwidths is based on multirate queuing models. [147] developed a dimensioning method based on a multirate $M/G/n-0$ queue for links with AC. In [4, 10] we introduced a dimensioning method based on a multirate $M/G/\infty$ queue for links with CO.

Capacity Dimensioning for AC Using the Multirate $M/G/n-0$ Queue

AC limits the number of flows to prevent overload. It blocks a new flow if its effective bandwidth together with the sum of the effective bandwidths of the already admitted flows exceeds the link capacity. The probability for a flow to be blocked at its arrival is denoted by $p_b^{r_i}(C)$. It depends on the flow rate r_i and the link capacity C. We use a multirate $M/G/n-0$ queue without buffers to derive this flow blocking probability. For the above rate distribution, each request rate $c(r_i)$ can be expressed as an integral multiple $c_u(r_i) = \frac{c(r_i)}{u_c}$ of a basic capacity unit $u_c = 64$ kbit/s. The link capacity C measured in basic capacity units u_c corresponds to the number of servers n of the queue. The sum of the request sizes of the active flows – also expressed in capacity units u_c – determines the number of busy servers. The system state probabilities of the $M/G/n-0$ queue can be calculated by the well-known Kaufman/Roberts algorithm presented in [227].

A newly arriving flow f experiences blocking if the system is in a state with insufficient free capacity to accommodate its request size $c(r(f))$. The blocking probability for f is the sum of the probabilities of all states in which blocking occurs for a flow with rate $c(r(f))$. Flows with large request rates face a larger blocking probability than small flows. Thus, we use the average of the blocking probabilities $p_b^{r_i}(C)$ of all request types r_i weighted by their occurrence probability $p(r_i)$ and rate $c(r_i)$ as the blocking probability $p_b(C)$ for capacity dimensioning.

We now can dimension the link capacity $C = n \cdot u_c$ by choosing the number of servers n large enough that admission requests are rejected only with an acceptable target blocking probability p_b: $C = \min_{C'}\{p_b(C') \leq p_b\}$. The algorithm in [147] was designed to calculate this number in an efficient way.

Capacity Dimensioning for CO Using the Multirate $M/G/\infty$ Queue

CO does not block any flows. With CO, the number of flows on a link is theoretically unbounded. Therefore, we model a link by a multirate $M/G/\infty$ queue with infinitely many server units. The calculation of the equilibrium state probabilities of the number of busy servers of such a queue is known as the stochastic knapsack with infinite capacity [246]. Its solution was originally derived for the $M/M/\infty$ system, but due to its insensitivity to the holding time distribution it is also valid for $M/G/\infty$ systems.

The equilibrium state probabilities can be calculated as follows. The n_r different request types define $k = n_r$ classes and the k-dimensional state space is described by $\mathcal{X} = \{x = (x_0, x_1, \ldots, x_{k-1}) \in \mathbb{N}_0^k\}$ where x_i denotes the number of flows of request type r_i in the system. Hence, the type-specific rates $c(r_i)$ yield the theoretically required link capacity $c(x) = \sum_{i=0}^{k-1} c(r_i) \cdot x_i$ of state x, which may be clearly above the actual link capacity in certain states. The equi-

librium state probabilities in product-form are

$$p(x) = \prod_{i=0}^{k-1} \frac{a_i^{x_i}}{x_i!} e^{-a_i} \tag{5.1}$$

with a_i being the class-specific offered load in Erlang.

We discuss two different QoS violation probabilities for CO that both depend on the link bandwidth C.

p_v^a The first definition is consistent with the definition of the flow blocking probability p_b. It is the QoS violation seen by a newly arriving flow f. This probability p_v^a comprises the probability of all states in which a new flow sees a QoS violation after its arrival.

$$p_v^a(C) = 1 - \sum_{0 \le i < n_r} p(r_i) \cdot \sum_{\{x \in \mathcal{X} : c(x) \le C - c(r_i)\}} p(x) \tag{5.2}$$

p_v^t The second definition is the QoS violation probability p_v^t over time. Thus, it is simply

$$p_v^t(C) = 1 - \sum_{\{x \in \mathcal{X} : c(x) \le C\}} p(x) \tag{5.3}$$

Note that probability $p_v^t(C)$ is smaller than $p_v^a(C)$.

An overprovisioned link requires so many capacity units C that the probability for the flows to exceed this bandwidth is smaller than a given tolerable violation probability p_v. Thus, the required capacity is

$$C = \min_{C'} \{ p_v^{\{a,t\}}(C') \le p_v \}. \tag{5.4}$$

5.2 Capacity Requirements for CO and AC on a Single Link

In this section we present the capacity requirements for CO and AC on a single link and perform a sensitivity analysis regarding basic parameters. This provides a good understanding of the dimensioning methods that later are applied in the network context. It also shows that the input parameters for capacity dimensioning have a visible but moderate influence on the required capacity. This verifies that the findings for the special choice of a parameter set have a more general validity.

For this purpose, we first study the impact of various parameters on the required link capacity to illustrate the dimensioning methods described in Section 5.1.4. We also assess the amount of capacity that is missing to fully satisfy all request on overprovisioned links in case the request rates actually exceed the link bandwidths and argue for an enhancement of the traffic model that leads to more overload than the simple Poisson model.

5.2.1 Impact of the Dimensioning Method on the Required Capacity

We dimension a single link for CO and AC with $p_v^a, p_v^t, p_b = 10^{-3}$ (c.f. Section 5.1.4). All flows have a homogeneous effective bandwidth of 256 kbit/s, i.e. $t = 0$ (cf. Table 5.1). Figure 5.1 shows the absolute required capacity depending on the offered link load given in Erlang. The step function appearance at the beginning of the curves in the lower left margin of the figure is due to granularity effects for small offered load. If the target probabilities become too large, the link capacity always must be upgraded in steps of 256 kbit/s. Apart from that, the absolute required capacity increases almost linearly with the offered load for more than 100 Erlang, but the lines hardly differ and it is difficult to interpret the results.

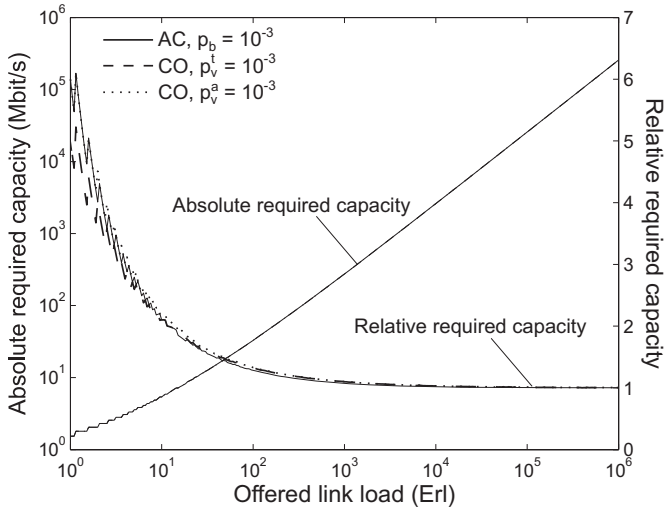


Figure 5.1: *Impact of the offered load on the absolute and relative required capacity of a single link for different dimensioning methods.*

Therefore, we also plot the relative required capacity as a multiple of the average offered traffic in the same figure. The average offered traffic is the average offered load times the average bandwidth of a flow of 256 kbit/s. It clearly shows that the relative amount of additional capacity decreases with an increasing offered load. This fact is called economy of scale.

AC requires less capacity than CO if p_v^a, the QoS violation probability seen by a newly arriving flow f, is used for capacity dimensioning with CO. AC blocks some of the traffic and thus slightly reduces the load in the system compared to CO. If p_v^t, the QoS violation probability over time, is applied as capacity dimensioning objective, the required capacity for CO is reduced to such an extent that it is smaller than the one for AC for very small offered load. However, the difference between all methods is negligible for medium or large offered load.

Since this difference is negligible, we denote p_v^t simply by p_v and use it in the following as the objective for capacity dimensioning with CO since it measures the QoS violations perceived by all flows in progress.

5.2.2 Impact of the Request Rate Variability and the Target Probabilities on the Capacity

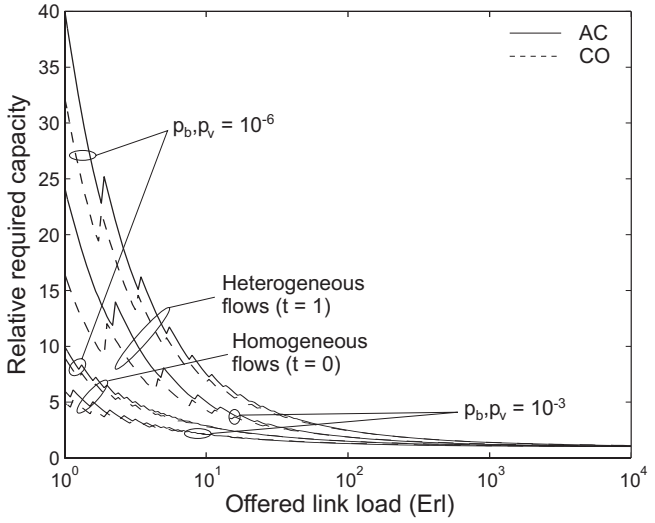


Figure 5.2: *Impact of the distribution of the effective flow bandwidth and the objective probabilities for capacity dimensioning on the required capacity of a single link.*

We next investigate the impact of the objective probabilities p_v and p_b and the request rate variability parameter t (cf. Table 5.1) on the required link capacity. We consider both a homogeneous traffic mix ($t=0$) and a strongly heterogenous traffic mix ($t=1$) for the objective probabilities $p_v, p_b = 10^{-3}$ and 10^{-6}. The results are compiled in Figure 5.2.

For $p_v = p_b$, AC and CO need almost the same amount of capacity. Smaller objective probabilities and heterogeneous effective bandwidths increase the required link capacity significantly, but only for little offered load. The influence of the variability of the effective bandwidth is clearly stronger than the one of the target probabilities.

In the following we use $t=1$ since it is more realistic than $t=0$ for Internet flows whose request rates can be highly variable.

5.2.3 Impact of the Target Probability for CO on the Actual QoS Violation

The severity of the QoS violation perceived by the user depends on the actual amount of capacity that is missing to fully satisfy all requests. Therefore, we calculate the average of the missing capacity M in case of CO relative to the provisioned link capacity C by

$$E[M] = \frac{1}{C} \cdot \sum_{\{x \in \mathcal{X} : c(x) > C\}} p(x) \cdot (c(x) - C) \tag{5.5}$$

where x is the state vector of the multirate $M/G/\infty$ queue that describes the number of flows in the system.

Figure 5.3 shows the missing capacity in percent for the above experiments with heterogeneous traffic for $p_v = 10^{-3}$ and 10^{-6}. The jerky leaps of the graph are again caused by granularity effects. We scaled the left y-axis in multiples of p_v. This makes immediately visible that the average of the percentage of the missing capacity is in the order of p_v, i.e. 10^{-3} and 10^{-6}, respectively.

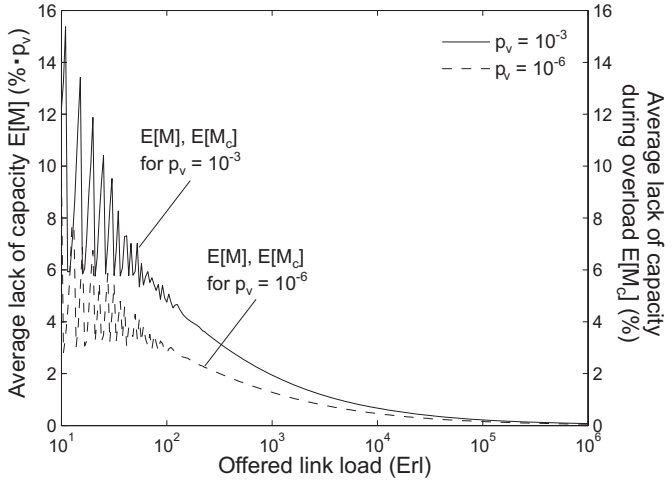


Figure 5.3: *Impact of the offered load and the target probability p_v on the overall and conditional average QoS violation $E[M]$ and $E[M_c]$ for CO.*

We also calculate the average of the missing capacity during overload situations.This is a conditional average $E[M_c] = \frac{E[M]}{p_v}$. According to the construction of the graph, the curves for $E[M]$ and $E[M_c]$ coincide, but they pertain to different y-axes. When the QoS is violated, approximately 4% or 8% capacity is missing for little offered load and a target probability of $p_v = 10^{-6}$ or 10^{-3}, respectively. For medium offered load around 1% or 2% are missing, and for large offered load the missing capacity is negligible regardless of p_v.

These values are surprisingly low which results from the smooth behavior of the Poisson model and the fact that we assumed constant offered load in our experiment. This allows only small statistical oscillations and does not model overload due to increased content attractiveness at certain locations.

For the rest of this monograph we choose a maximum flow blocking probability $p_b = 10^{-3}$ for AC and a maximum QoS violation probability $p_v = 10^{-6}$ for CO. The difference is motivated by the fact that flow blocking is annoying only for the affected user while QoS violation hits all flows in progress. Note that the required capacity and the QoS violation revealed only little sensitivity to the target parameters for medium and large offered load. The required capacity is mainly controlled by the offered load.

5.2.4 Impact of Transient Overload on the Capacity

In Section 2.5.1 we identified different sources of overload. The Poisson model accounts for overload due to stochastic fluctuations of the number of flows in the network. We now model the impact of transient overload on the capacity of a single link for a better understanding of our model for transient traffic shifts in networks developed later in this chapter. For this purpose, we assume a constant offered load for most of the time and a temporary increase of the normal offered load by an overload factor of f_l. AC can block excess traffic during time of overload and preserve QoS at the expense of blocked flows. In contrast, CO must provide so much capacity that the excess traffic can be carried. Otherwise the QoS will be unacceptable for all flows in the network.

Figure 5.4 shows the required capacity for CO and AC together with the flow blocking probability p_b^o for AC during time of overload. The results are shown for an offered load of $a = 10^2$ and 10^5 Erlang in the non-overload case.

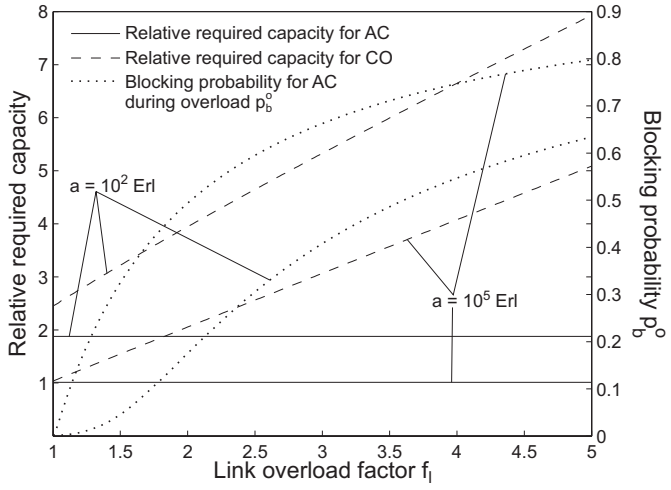


Figure 5.4: *Impact of the overload factor f_l on the required capacity and the blocked traffic of a single link for $a = 10^2$ and 10^5 Erlang.*

Since the required capacity for AC is dimensioned for the non-overload case, the respective curves are independent of the overload factor. However, the blocking probability for AC increases with the overload factor f_l. The blocking probability for 10^5 Erlang is larger than the one for 10^2 Erlang because there is less additional capacity available relative to the average traffic rate due to economy of scale.

The overload factor $f_l = 1$ denotes the non-overload case for CO. Here, CO requires visibly more capacity than AC for $a = 10^2$ Erlang because it uses $p_v = 10^{-6}$ as target probability for dimensioning instead of $p_b = 10^{-3}$. However, for $a = 10^5$ Erlang, the capacity requirements for CO and AC are almost equal for $f_l = 1$. With an increasing overload factor f_l, the required capacity for CO scales about linearly with f_l since it must be dimensioned for the offered load during the overload interval.

In fact, this result for a single link is trivial. Therefore, in the next section,

we consider different types of overload in networks that are not due to increased overall traffic, but rather model traffic shifts that cause local overload.

5.3 Capacity Requirements for CO and AC in Networks

In our analysis of the capacity requirements for CO and AC on a single link, we mainly modeled overload by means of the Poisson model. The Poisson model accounts for stochastic fluctuations. It varies the number of flows in the traffic aggregate, i.e., it models overload due to (a) (cf. Section 2.5.1). However, if we keep the average offered load constant, it produces very smooth traffic rates such that only little additional capacity is needed both in networks with CO and AC.

In this section, we investigate the impact of overload that results from traffic shifts within the network. The traffic shifts temporarily increase the offered load on individual links without increasing the overall traffic in the network. Such traffic shifts may result from increased content attractiveness at certain locations (b) or from redirected traffic due to network failures (c). For both issues we now must consider the entire network instead of a single link. We first extend our performance analysis method to networks and then investigate the impact of traffic shifts and redirected traffic on the required capacity for CO and AC.

5.3.1 Resilient Capacity Dimensioning Framework for CO and AC in Networks

We extend the traffic model and the dimensioning methods for CO and AC from Section 5.1.4 to networks. This yields a capacity dimensioning framework for CO and AC in networks that accounts for temporary traffic shifts.

Traffic Model

In order to incorporate temporary traffic shifts into our analysis, we use the gravity model [243] to generate a basic traffic matrix from which we derive traffic matrices with simple and complex traffic shifts. We further introduce a notation for network failures and the resulting (re)routing that also leads to traffic shifts.

Basic Traffic Matrix Most of the network experiments in this chapter are based on the Labnet03 reference network from Figure 5.5 [147]. But we also use random networks to obtain more general results. Here we describe the construction of the basic traffic matrix for the Labnet03 network based on the gravity model.

The set of nodes $\mathcal{V}$ together with the set of bidirectional edges $\mathcal{E}$ describe the topology of the network. All network nodes are both ingress and egress routers, i.e., they act both as traffic sources and destinations. The average border-to-border (b2b) load between two nodes in the network is constant and denoted by a_{b2b}. It determines the overall offered load in the network

$$a_{tot} = \sum_{v,w \in \mathcal{V}, v \neq w} a(v,w) = |\mathcal{V}| \cdot (|\mathcal{V}|-1) \cdot a_{b2b}.$$

The generation of the traffic matrix is based on the population of the cities and their surroundings as compiled in [147]. For two cities corresponding to the nodes v and w with population sizes $\pi(v)$ and $\pi(w)$, the b2b offered load $a_{b2b}(v,w)$ amounts to

$$a_{b2b}(v,w) = \begin{cases} \frac{a_{tot} \cdot \pi(v) \cdot \pi(w)}{\sum_{x,y \in \mathcal{V}, x \neq y} \pi(x) \cdot \pi(y)} & \text{for } v \neq w, \\ 0 & \text{for } v = w. \end{cases} \tag{5.6}$$

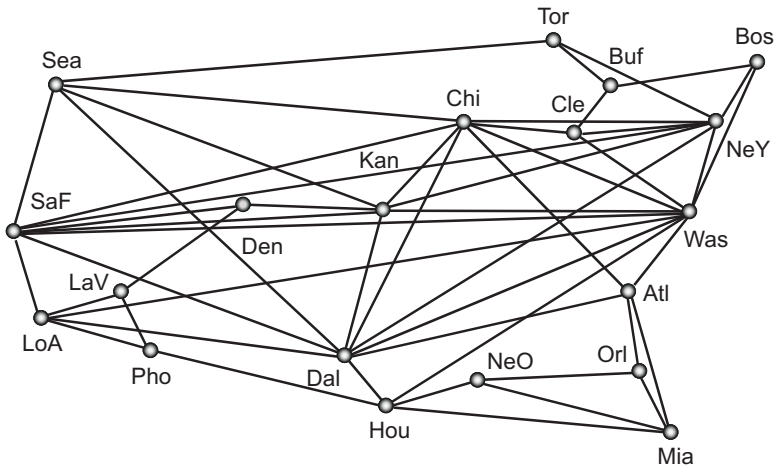


Figure 5.5: *Topology of the Labnet03 network with 20 nodes and 53 bidirectional links.*

Hot Spot Model for Transient Traffic Shifts A hot spot v in the network models the increased traffic attraction of a single city. We describe it by a hot spot factor f_h and a modified population function

$$\pi^{v}_{overload}(w) = \begin{cases} \pi(w) & \text{if } w \neq v \\ f_h \cdot \pi(w) & \text{if } w = v. \end{cases} \tag{5.7}$$

The modified population function is used as input for Equation (5.6). This overload model is conservative since it does not increase the overall traffic in the network. It causes a traffic shift which changes the structure of the traffic matrix. As a consequence, an increased or decreased load on individual links can be observed. Note that every node $v \in \mathcal{V}$ is a potential hot spot and even several hot spots may occur simultaneously. Therefore, we characterize a hot spot scenario by the set of routers with increased attractiveness, e.g. $h = \{v, w\}$ is a double hot spot where an increased traffic attraction is observed for nodes v and w. In the following, $\mathcal{H}$ denotes the set of considered hot spot scenarios and it contains always the normal scenario $h = \emptyset$. Note that traffic variations may also be caused

by other influences, e.g. inter-domain rerouting [247], that may as well increase the overall traffic volume in the network.

Network Failures and Routing Changes The connectivity of the network after a failure depends on the failure topology and the applied restoration or protection switching mechanism. In our experiments, we use shortest path routing since it is the basis for the most frequently used Interior Gateway Protocols (IGPs) OSPF [48] and IS-IS [49, 50].

We characterize a network failure s by the set of failed network elements, e.g. links or routers. The QoS during network failures depends both on the connectivity in a failure scenario and on the available capacity of the backup paths. Since we consider only single link failures in our investigation, the connectivity is not a problem. But for full resiliency we must dimension the required capacity in such a way that it prevents overload due to the redirected traffic for all protected failure scenarios $\mathcal{S}$. This set contains the failure-free case $s = \emptyset$ by default.

The traffic aggregate between v and w is denoted by $g(v, w)$. The set of all aggregates in the network is $\mathcal{G}$. A failure case s influences the routing of an aggregate $g \in \mathcal{G}$ within the network. We describe the routing by the function $u(s, l, g)$ that describes the percentage of the traffic rate $c(g)$ that uses link l in a specific failure case $s \in \mathcal{S}$, i.e., the routing in the failure-free case is given by the function $u(\emptyset, l, g)$. This notation is very general since it expresses the routing of arbitrary restoration and protection mechanisms and copes well with load balancing.

Capacity Dimensioning in Networks in the Presence of Traffic Shifts and Network Failures

We extend the capacity dimensioning methods for CO and AC on a single link from Section 5.1.4 to networks and adapt them to traffic shifts and network failures. This establishes the concept of "resilient capacity overprovisioning" which is the heart of our dimensioning framework.

Since we consider traffic shifts due to increased content attractiveness and

network failures, a network scenario $z = (h, s)$ is determined by its traffic matrix depending on the hot spot scenario h and the failure scenario s. Conversely, the functions $h(z)$ and $s(z)$ yield the respective hot spot and failure scenarios.

Dimensioning of Link Capacities in a Network for CO When we calculate the offered load $a(z, l)$ for link l in a specific networking scenario z we must consider the load contribution of all traffic aggregates for that link:

$$a(z, l) = \sum_{g \in \mathcal{G}} a(h(z), g) \cdot u(s(z), l, g). \tag{5.8}$$

Based on this value and the given target probability p_v, the capacity dimensioning algorithm for CO presented in Section 5.1.4 computes the capacity $c(z, l)$ of that link for networking scenario z. The required link capacity for a set of networking scenarios $\mathcal{Z}$ is then simply the maximum link capacity of all its networking scenarios $z \in \mathcal{Z}$:

$$c(l) = c(\mathcal{Z}, l) = max_{z \in \mathcal{Z}} \left(c(z, l) \right). \tag{5.9}$$

Dimensioning of Link Capacities in a Network for AC We dimension the capacity for the BBB NAC described in Section 2.5.3 since this NAC method is resilient to network failures if configured appropriately. For each traffic aggregate $g \in G$, a b2b budget exists with a capacity of $c(g)$ that can be dimensioned based on the offered load $a(\emptyset, g)$ with the link dimensioning algorithm for AC presented in Section 5.1.4. Note that in networks with resilient AC, failures but no hot spots need to be respected since overload due to hot spots can be blocked. Thus, the capacity for link l in the networking scenario z amounts to

$$c(z, l) = \sum_{g \in \mathcal{G}} c(g) \cdot u(s(z), l, g). \tag{5.10}$$

The required capacity for a set of networking scenarios $\mathcal{Z}$ is again calculated according to Equation (5.9).

5.3.2 Performance Measure and Networking Scenarios under Study

We shortly describe the performance measure for networks and the selected sets of networking scenarios under study as the basis for the results presented later in this section.

Performance Measure Similar to Section 5.2, we select the relative required capacity as our performance measure. However, its definition for single links must be adapted to networks. The absolute required network capacity is the capacity of all links in the network and amounts to $C_{abs} = \sum_{l \in \mathcal{E}} c(l)$. The average traffic rate under normal conditions is

$$C_{avg} = E(c(R_t)) \cdot \sum_{l \in \mathcal{E}} a(z = (\emptyset, \emptyset), l).$$

Thus, we define the relative required network capacity by $C_{rel} = \frac{C_{abs}}{C_{avg}}$.

Sets of Networking Scenarios under Study The sets of networking scenarios in this section are of particular interest for our study. We assess their size for the test network in Figure 5.5 to give an idea of the complexity of our investigation. The sets $\mathcal{Z}^{i,0}$ do not contain failure scenarios and are used for the examination of the impact of single and double hot spot scenarios without link failures.

- $\mathcal{Z}^{0,0} = \{(\emptyset, \emptyset)\}$; "the basic traffic matrix in the failure-free scenario", $|\mathcal{Z}^{0,0}| = 1$.
- $\mathcal{Z}^{1,0} = \mathcal{Z}^{0,0} \cup \{$"all single hot spots in the failure-free scenario"$\}$, $|\mathcal{Z}^{1,0}| = |\mathcal{Z}^{0,0}| + \binom{|\mathcal{V}|}{1} = 1 + 20 = 21$.
- $\mathcal{Z}^{2,0} = \mathcal{Z}^{1,0} \cup \{$"all double hot spots in the failure-free scenario"$\}$, $|\mathcal{Z}^{2,0}| = |\mathcal{Z}^{1,0}| + \binom{|\mathcal{V}|}{2} = 21 + 190 = 211$.

The sets $\mathcal{Z}^{i,1}$ contain networking scenarios with all single link failures for the examination of the impact of single and double hot spot scenarios in the presence of link failures.

- $\mathcal{Z}^{0,1} = \mathcal{Z}^{0,0} \cup \{$"all single link failure scenarios without hot spots"$\}$, $|\mathcal{Z}^{0,1}| = |\mathcal{Z}^{0,0}| + \binom{|\mathcal{E}|}{1} = 1 + 53 = 54$.
- $\mathcal{Z}^{1,1} = \mathcal{Z}^{0,1} \cup \{$"all single link failure scenarios combined with all simultaneous single hot spots"$\}$, $|\mathcal{Z}^{1,1}| = |\mathcal{Z}^{0,1}| + |\mathcal{Z}^{0,1}| \cdot \binom{|\mathcal{V}|}{1} = 54 + 54 \cdot 20 = 1134$.
- $\mathcal{Z}^{2,1} = \mathcal{Z}^{1,1} \cup \{$"all single link failure scenarios combined with all simultaneous double hot spots"$\}$, $|\mathcal{Z}^{2,1}| = |\mathcal{Z}^{1,1}| + |\mathcal{Z}^{0,1}| \cdot \binom{|\mathcal{V}|}{2} = 1134 + 54 \cdot 190 = 11394$.

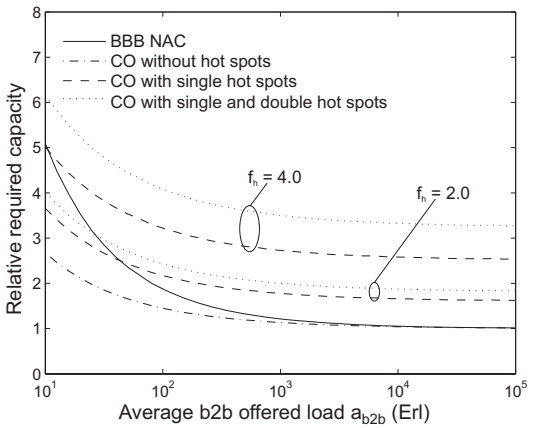
5.3.3 Numerical Results

We now study the relative required overall capacity for networks with CO and compare it to networks with AC. The results present the impact of simple and complex traffic shifts in networks with and without resilience requirements. The comparisons were conducted in Labnet03 (cf. Figure 5.5) and in random networks of different size. We first present the capacity requirements in non-resilient networks and then in resilient networks. However for better comparability, we print Figures 5.6 and 5.8 for non-resilient networks together with Figures 5.7 and 5.9 for resilient networks.

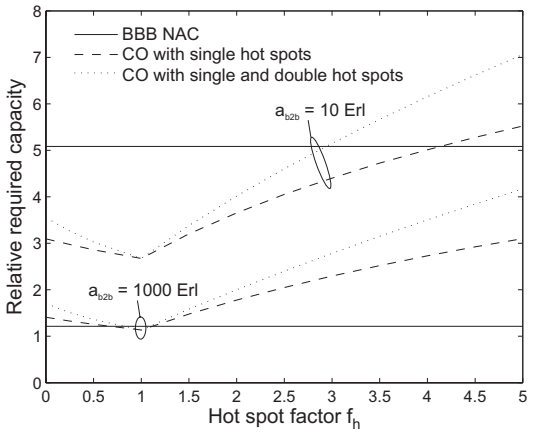
Capacity Requirements in Non-Resilient Networks

We illustrate the impact of hot spot scenarios on the required capacity for CO and AC in non-resilient networks.

Experiments with Labnet03 Figure 5.6(a) shows the relative required capacity in Labnet03 depending on the average offered load a_{b2b} between any

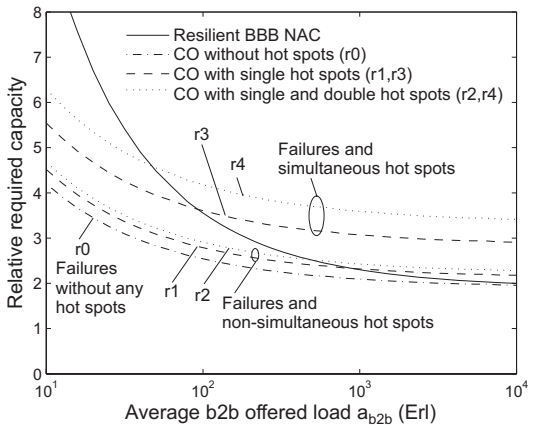


(a) Influence of the b2b offered load a_{b2b} in Labnet03 for hot spot factors of $f_h = 2$ and 4.

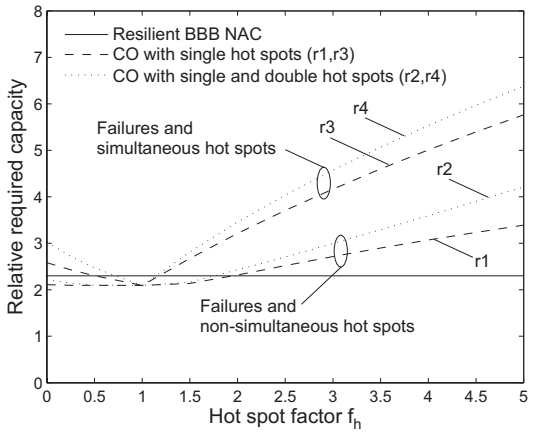


(b) Influence of the hot spot factor f_h in Labnet03 for a b2b offered load of $a_{b2b} = 10$ and 1000 Erlang.

Figure 5.6: *Relative required capacity in the* non-resilient *Labnet03 with capacity overprovisioning (CO) and admission control (AC), respectively.*



(a) Influence of the b2b offered load a_{b2b} in Labnet03 for a hot spot factor of $f_h = 2$.



(b) Influence of the hot spot factor f_h in Labnet03 for a b2b offered load of $a_{b2b} = 1000$ Erlang.

Figure 5.7: *Relative required capacity in the* resilient *Labnet03 with capacity overprovisioning (CO) and admission control (AC), respectively.*

two border routers. The network capacity is dimensioned for the networking scenarios $\mathcal{Z}^{0,0}$ (without hot spots), $\mathcal{Z}^{1,0}$ (single hot spots only), and $\mathcal{Z}^{2,0}$ (single and double hot spots), and for BBB NAC. The hot spot factor is set to $f_h = 2$ and to $f_h = 4$, respectively.

Like in the single link experiments, the relative required capacity decreases for all curves with an increasing load. Surprisingly, CO without hot spots ($\mathcal{Z}^{0,0}$) requires less capacity than AC. This is due to the following reason. CO can take advantage of the fact that the offered load on a link is larger than the load for a single budget. The capacity dimensioning for a specific link for CO is based on the overall load of all aggregates carried over this link (cf. Equation (5.8)). In contrast, the BBB NAC considers only the load of a single aggregate for each b2b budget and the link capacity is the sum of the capacity requirements for all b2b budgets carried over this link (cf. Equation (5.10). Thus, CO benefits from increased economy of scale which leads to less required capacity for CO than for AC. For sufficiently large offered load, this advantage for CO vanishes.

CO with single hot spots requires more capacity than AC since it must provide enough resources for all possible traffic shifts. CO for double hot spots needs visibly more resources than CO for single hot spots. An increase of the hot spot factor from $f_h = 2$ to $f_h = 4$ also increases the resource requirements for CO considerably.

Figure 5.6(b) shows the relative required network capacity for an offered b2b load of $a_{b2b} = 10$ and 1000 Erlang depending on the hot spot factor f_h. The capacity curves for $a_{b2b} = 1000$ Erlang reveal an almost linear growth, but the growth is smaller than f_h. This is different to the experiment on the single link (cf. Figure 5.4) which can be explained as follows. The links adjacent to a hot spot carry all the "hot spot traffic" from and to this hot spot. The rate of these aggregates scales almost with f_h. However, the transit traffic on these links is not increased or even decreased by the hot spot. As a consequence, the required capacity for the adjacent links grows less than by f_h since their traffic consists of increased hot spot and slightly decreased transit traffic.

The capacity curves for single hot spots require less resources than those for

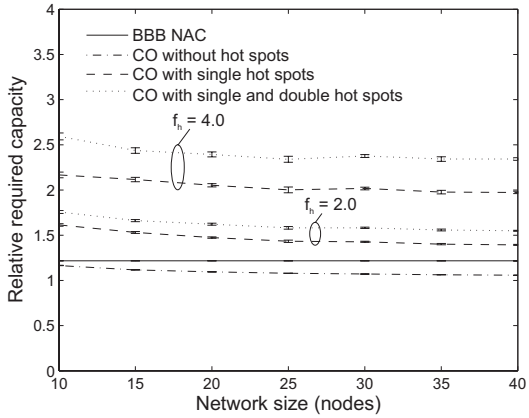
double hot spots. They meet for $f_h = 1$ since this is the value for CO without any hot spots. Hot spot factors $f_h < 1$ produce "cold spots", i.e., the attractiveness of a certain node is reduced which also effects a traffic shift. However, a cold spot leads only to a small increase of the required capacity.

The required network capacity for AC is independent of the hot spot factor and produces, therefore, horizontal lines. For very little offered load of $a_{b2b} = 10$ Erlang, AC requires significantly more resources than CO, but for a large offered load of $a_{b2b} = 1000$ Erlang, AC works efficiently enough such that it saves capacity by blocking excess traffic in overload situations.

Experiments with Random Networks We conduct a parametric study using random networks to investigate the impact of the network size on the relative required network capacity for CO and AC. Similar to our MPLS-FRR performance study (cf. Section 4.2), we construct the random networks with n nodes and an average node degree of $deg_{avg} = 3$, i.e. with $m = \frac{n \cdot deg_{avg}}{2}$ bidirectional links, using the algorithm given from [147]. This algorithm guarantees a connected graph and keeps the degree of every node between $2 \leq deg_{avg} \leq 4$. We altogether used 140 networks, 20 networks for each network size of $n \in \{10, 15, 20, 25, 30, 35, 40\}$ nodes. Like above, we dimension the capacity of these networks for CO without hot spots, with single hot spots only, and with single and double hot spots.

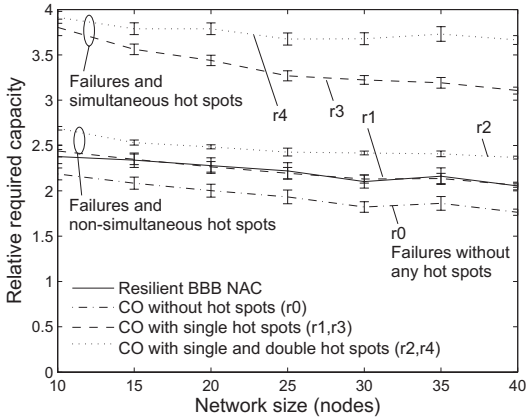
Figure 5.8(a) illustrates the results for an average b2b offered load of $a_{b2b} = 1000$ Erlang and for hot spot factors of $f_h = 2$ and 4. Each point corresponds to the relative required capacity averaged over all 20 networks of a single size, the bars below and above the curves indicate the confidence intervals. The relative required capacity for CO without hot spots decreases slightly for an increasing network size. Since larger networks lead to more offered load per link, CO benefits from an increased economy of scale. In contrast, BBB NAC cannot benefit from that since the average offered load per budget is independent of the network size. For a hot spot factor of $f_h = 2$, single hot spots only lead to about 50% more capacity whereas single and double hot spots lead to 75% more capacity than the

average traffic rate in the network. Doubling the hot spot factor to $f_h = 4$ also doubles the additional capacity requirements to 100-150%.



(a) Influence of the network size for a b2b offered load of $a_{b2b} = 1000$ Erlang and hot spot factors $f_h \in \{2, 4\}$.

Figure 5.8: *Relative required capacity in* non-resilient *random networks with capacity overprovisioning (CO) and admission control (AC), respectively.*



(a) Influence of the network size for a b2b offered load of $a_{b2b} = 1000$ Erlang and a hot spot factor of $f_h = 2$.

Figure 5.9: *Relative required capacity in* resilient *random networks with capacity overprovisioning (CO) and admission control (AC), respectively.*

Capacity Requirements in Resilient Networks

We illustrate the impact of hot spot scenarios on the required capacity for CO and AC in networks with resilience requirements.

Experiments with Labnet03 We consider the following 5 types of networking scenarios for our performance analysis of CO.

r0 $\mathcal{Z} = \mathcal{Z}^{0,1}$, i.e. resilience against link failures without elasticity for any hot spots.

r1 $\mathcal{Z} = \mathcal{Z}^{0,1} \cup \mathcal{Z}^{1,0}$, i.e. resilience against link failures with elasticity for non-simultaneous (i.e. not during failure situations) single hot spots.

r2 $\mathcal{Z} = \mathcal{Z}^{0,1} \cup \mathcal{Z}^{2,0}$, i.e. resilience against link failures with elasticity for non-simultaneous single and double hot spots.

r3 $\mathcal{Z} = \mathcal{Z}^{1,1} \cup \mathcal{Z}^{2,0}$, i.e. resilience against link failures with elasticity for non-simultaneous single and double hot spots and simultaneous (i.e. also during failure situations) single hot spots.

r4 $\mathcal{Z} = \mathcal{Z}^{2,1} \cup \mathcal{Z}^{2,0}$, i.e. resilience against link failures with elasticity for simultaneous and non-simultaneous single and double hot spots.

Concerning the relevance of these networking scenarios for capacity dimensioning in practice, we make the following considerations. We assume the probability of a link failure to be smaller than the one for a hot spot, i.e. $p_l < p_h$. Single link failures must be protected as well as double hot spots. However, we expect that the simultaneous occurrence of a single link failure together with a simultaneous hot spot is so unlikely that we do not need to provide additional capacity for those scenarios. Under these assumptions, option r2 is appropriate for resilient CO in practice.

Experiments with Labnet03 Figures 5.7(a) and 5.7(b) show the relative required capacity for CO and AC with resilience against single link failures in Labnet03. They correspond to Figures 5.6(a) and 5.6(b), but we show the results for the above mentioned options only for $f_h = 2$.

Figure 5.7(a) shows that resilient CO and AC require both substantially more capacity than CO or AC without resilience against link failures. They both require backup capacities for redirected traffic on the links. The limit the capacity requirements converges to for large a_{b2b} depends on the network topology and the applied restoration or protection switching mechanism. Note that the backup capacity can be minimized by routing optimization [16, 146].

The curves for resilient CO (r1) and (r2) require only marginally more capacity than the curve for (r0). This means that the backup capacity for single link failures almost suffices to absorb traffic shifts due to single and double hot spots for a hot spot factor of $f_h = 2$. As a consequence, resilient CO for application in practice (r2) requires only little more capacity than resilient BBB NAC for realistic

load values. We also plotted the options r3 and r4 for resilient CO in the figures to illustrate that they need about 100% more capacity than r0, r1, and r2. This extra capacity is needed during single or even double hot spots to simultaneously accommodate redirected traffic caused by link failures.

Figure 5.7(b) keeps the offered load fixed at $a_{b2b} = 1000$ Erlang and varies the hot spot factor f_h. Resilient CO (r2) is as efficient as resilient AC for hot spot factors up to about $f_h = 2$.

The two Figures 5.7(a) and 5.7(b) show that the relative required capacity for resilient CO depends on the offered load a_{b2b}, the hot spot factor f_h, and the resilience option. In contrast, for resilient AC it depends only on the offered load a_{b2b}.

Experiments with Random Networks Finally, we conduct our analysis with resiliency in random networks from Section 5.3.3 such that the results in Figure 5.9(a) are comparable to those in Figure 5.8(a). The figure shows that resilient CO without elasticity for simultaneous hot spots during link failures (r0, r1, r2) needs a similar amount of capacity like resilient AC for $a_{b2b} = 1000$ and $f_h = 2$. Resilient CO with elasticity for simultaneous hot spots again requires about 100% more additional resources. This observation is apparently independent of the network size.

5.4 Summary: Dimensioning of Resilient Networks

Network dimensioning is a complex task. It must provide sufficient capacity such that service level agreements (SLAs) with customers can be fulfilled. For this purpose, network providers apply two basically different methods to avoid congestion in the network in case of unexpected traffic load: capacity overprovisioning (CO) and admission control (AC).

Capacity overprovisioning (CO) provides abundant capacity on the links to avoid QoS degradation due to overload in the network. CO must take into account any kind of overload: (a) overload due to statistical variations of the normal traffic matrix, (b) overload due to changed traffic matrices caused by traffic shifts through popular sites (or by changes of the inter-AS routing), and (c) overload due to redirected traffic caused by network failures.

The model in this chapter is the first to tackle all three sources of overload. So far overprovisioning techniques were based on simple rules of thumb leading to massive capacity underutilization in core networks. In contrast, our work introduces the notion of resilient overprovisioning and proposes a capacity dimensioning method for keeping the QoS violation probability p_v below a given limit for important considered networking scenarios $z \in \mathcal{Z}$.

This method is especially useful for a comparison of CO with AC methods. In addition, the idea of resilient CO can be certainly adapted to other traffic and overload models, e.g. to overload caused by routing changes of inter-AS routing.

Admission control (AC) is the counterpart to CO. Resilient AC is a requirement since the majority of overload situations in the Internet results from network failures [136]. We dimensioned the link capacities for networks with AC in such a way that the flow blocking probabilities p_b are kept low.

We examined the impact of all three sources of overload (a-c) on the required capacity by the performance measure "relative required capacity". This is the required capacity relative to the average traffic rate. We compared them for networks with CO and AC. The offered system load, the strength of traffic shifts, and the network size were key parameters for our investigation. The most important results of our study are the following.

- The target probabilities p_v and p_b for capacity dimensioning have only a small impact on the required capacity for CO and AC.
- The statistical fluctuations of the Poisson model for flows do not lead to significant overload and QoS violations. Therefore, additional overload models are needed.

- In networks without hot spots and failures, CO requires about the same capacity as AC or even less as it can take better advantage of economy of scale.
- Single hot spot scenarios lead to a significant increase of the required capacity for CO.
- Additional double hot spot scenarios increase these capacity requirements slightly.
- Resilience against link failures leads to increased capacity requirements for networks with CO and AC since both types require backup capacity for the redirected traffic.
- Resilient CO requires about the same network capacity as resilient AC to protect the network against failures and against overload due to single and double hot spots because the backup capacity can be used to absorb hot spots.
- We made these observations in a test network and confirmed them by a study of random networks of different size.

These findings can be generalized to other sources of overload, e.g. changes of the interdomain routing, since backup capacity can be reused to protect QoS against any kind of overload. Finally, since resilient CO requires about the same network capacity as resilient AC and AC is significantly more complex to deploy than CO, we conclude that CO is even more attractive than AC in networks with resilience requirements.

6 Conclusion

Future generation networks (FGNs) will carry new services with real-time constraints and strict availability and reliability requirements. Therefore, FGNs must facilitate end-to-end quality-of-service (QoS) guarantees and resiliency to failures. Another key requirement for FGNs is their efficient operation for the reduction of costs.

In this monograph we studied three aspects of provisioning and control for resilient FGNs: load balancing for multipath Internet routing, fast resilience concepts, and advanced dimensioning techniques for resilient networks.

Static load balancing on the flow level is not very accurate. For a moderate aggregation level of the load balanced traffic, the deviation from the target value can be as high as 30%. Such a large inaccuracy must be considered by the network resource management and is in fact counterproductive since multipath routing is often applied for traffic engineering purposes in order to save capacity.

Dynamic load balancing algorithms are a good alternative to reduce the inaccuracy to values as low as 6%, but they may cause packet reordering due to the dynamic reassignment of flows to other paths. On average, a flow is reassigned approximately every 25s. In addition, if flows undergo consecutive load balancing stages at different nodes along their paths, anti-polarization mechanisms are absolutely necessary and the number of consecutive stages should be kept low since this increases the probability for packet reordering. The results in Chapter 3 show that load balancing algorithms must be applied with care.

MPLS and IP fast reroute (FRR) are suitable concepts to achieve fast resilience in FGNs. The two options one-to-one and facility backup for MPLS-FRR have

significantly different backup capacity requirements if the standard path layout is applied. The one-to-one backup needs noticeably less backup capacity due to a better distribution of the backup traffic in the network. However, simple improvements of the path layout also reduce the capacity requirements for the facility backup. It remains slightly more expensive than the one-to-one option, but it is preferred in general due to its low configuration overhead.

Loop-free alternates (LFAs) and not-via addresses are two important IP-FRR approaches discussed within the IETF. The applicability of LFAs varies strongly between individual network nodes and the achieved degree of failure protection is limited. The joint application of LFAs and not-via addresses is therefore often seen as an appropriate option to combine the simplicity of LFAs with the full coverage of the more complex not-via addresses. However, a detailed analysis of aspects like the amount of decapsulated traffic does not reveal clear advantages of such a combination. Hence, not-via addresses should be applied as the only IP-FRR mechanism if 100% failure coverage with IP-FRR is required. Overall, there is a price to pay in terms of resource requirements for the deployment of FRR to achieve fast resilience. Chapter 4 evaluated this tradeoff between resource requirements and the benefits of fast resilience.

Advanced provisioning methods for resilient FGNs must incorporate any kind of overload: (a) overload due to statistical variations of the normal traffic matrix, (b) overload due to traffic shifts caused by popular content, and (c) overload due to redirected traffic during network failures. Appropriate packet- and flow-level traffic models characterize overload due to (a). So-called hot spot models reproduce overload due to (b), and the consideration of the changed routing during a set of protected failures accounts for overload due to (c).

Admission control (AC) can block overload due to (a) and (b) while capacity overprovisioning (CO) must consider all sources of overload. Hence, AC is superior to CO in networks where resiliency is not an issue, i.e., where overload due to (c) is neglected. However, in networks with resilience requirements, i.e. in resilient FGNs, resilient CO requires about the same network capacity as resilient AC since the backup capacity for failure protection can be used to absorb

hot spots. Since AC is significantly more complex to deploy than CO, CO is even more attractive than AC in resilient FGNs. Efficient CO requires careful traffic modeling. The capacity dimensioning framework from Chapter 5 facilitates this task.

In conclusion, simple and efficient provisioning and control for resilient FGNs is a challenging task. CO and fast rerouting as presented in this work is certainly an attractive solution to achieve QoS guarantees and fast resilience. When multi-path routing is applied, the effects of the inaccuracy of load balancing algorithms on the resource management must be considered.

Bibliography of the Author

— Conference Papers —

[1] M. Menth and R. Martin, "Performance Evaluation of the Extensions for Control Message Retransmissions in RSVP," in 7^{th} *IEEE International Workshop on Protocols for High-Speed Networks (PfHSN)*, (Berlin, Germany), pp. 35–49, Apr. 2002.

[2] R. Martin and M. Menth, "Improving the Timeliness of Rate Measurements," in 12^{th} *GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB) together with* 3^{rd} *Polish-German Teletraffic Symposium (PGTS)*, (Dresden, Germany), pp. 145–154, Sept. 2004.

[3] R. Martin, M. Menth, and V. Phan-Gia, "Performance of TCP/IP with MEDF Scheduling," in 3^{rd} *International Workshop on Quality of future Internet Services (QofIS)*, (Barcelona, Spain), pp. 94–103, Sept. 2004.

[4] R. Martin, M. Menth, and J. Charzinski, "Comparison of Border-to-Border Budget Based Network Admission Control and Capacity Overprovisioning," in 4^{th} *IFIP-TC6 Networking Conference (Networking)*, (Waterloo, ON, Canada), pp. 1056–1068, May 2005.

[5] R. Martin, M. Menth, and J. Charzinski, "Comparison of Link-by-Link Admission Control and Capacity Overprovisioning," in 19^{th} *International Teletraffic Congress*, (Beijing, China), pp. 1527–1538, Sept. 2005.

[6] M. Menth and R. Martin, "Network Resilience through Multi-Topology Routing," in 5^{th} *International Workshop on Design of Reliable Communication Networks (DRCN)*, (Island of Ischia (Naples), Italy), pp. 271–277, Oct. 2005.

[7] R. Martin and M. Menth, "Backup Capacity Requirements for MPLS Fast Reroute," in 7^{th} *ITG Workshop on Photonic Networks*, (Leipzig, Germany), pp. 95–102, Apr. 2006.

[8] R. Martin, M. Menth, and K. Canbolat, "Capacity Requirements for the Facility Backup Option in MPLS Fast Reroute," in *IEEE Workshop on High Performance Switching and Routing (HPSR)*, (Poznan, Poland), June 2006.

[9] M. Menth, R. Martin, and U. Spoerlein, "Network Dimensioning for the Self-Protecting Multipath: A Performance Study," in *IEEE International Conference on Communications (ICC)*, (Istanbul, Turkey), June 2006.

[10] M. Menth, R. Martin, and J. Charzinski, "Capacity Overprovisioning for Networks with Resilience Requirements," in *ACM SIGCOMM*, (Pisa, Italy), Sept. 2006.

[11] J. Milbrandt, R. Martin, M. Menth, and F. Höhn, "Risk Assessment of End-to-End Disconnection in IP Networks due to Network Failures," in 6^{th} *IEEE Workshop on IP Operations and Management (IPOM)*, (Dublin, Ireland), Oct. 2006.

[12] R. Martin, M. Menth, and M. Hemmkeppler, "Accuracy and Dynamics of Hash-Based Load Balancing Algorithms for Multipath Internet Routing," in *IEEE International Conference on Broadband Communication, Networks, and Systems (BROADNETS)*, (San Jose, CA, USA), Oct. 2006.

[13] R. Martin, M. Menth, and K. Canbolat, "Capacity Requirements for the One-to-One Backup Option in MPLS Fast Reroute," in *IEEE Interna-*

tional Conference on Broadband Communication, Networks, and Systems (BROADNETS), (San Jose, CA, USA), Oct. 2006.

[14] M. Menth, R. Martin, and U. Spoerlein, "Impact of Unprotected Multi-Failures in Resilient SPM Networks: a Capacity Dimensioning Approach," in *IEEE Globecom*, (San Francisco, California, USA), Nov. 2006.

[15] R. Martin and M. Menth, "Impact of Multi-Failures in Survivable Networks," in 8^{th} *ITG Workshop on Photonic Networks*, (Leipzig, Germany), May 2007.

[16] M. Menth, R. Martin, and U. Spörlein, "Robust IP Link Costs for Multilayer Resilience," in *IEEE International Conference on Communications (ICC)*, (Atlanta, GA, USA), May 2007.

[17] R. Martin, M. Menth, and U. Spoerlein, "Integer SPM: Intelligent Path Selection for Resilient Networks," in *IFIP-TC6 Networking Conference (Networking)*, (Atlanta, GA, USA), May 2007.

[18] M. Menth, R. Martin, and U. Spoerlein, "Optimization of the Self-Protecting Multipath for Deployment in Legacy Networks," in *IEEE International Conference on Communications (ICC)*, (Glasgow, Scotland), June 2007.

[19] R. Martin, M. Menth, and M. Hemmkeppler, "Accuracy and Dynamics of Multi-Stage Load Balancing for Multipath Internet Routing," in *IEEE International Conference on Communications (ICC)*, (Glasgow, Scotland), June 2007.

[20] M. Menth, R. Martin, A. M. Koster, and S. Orlowski, "Overview of Resilience Mechanisms Based on Multipath Structures," in *DRCN*, (La Rochelle, France), Oct. 2007.

[21] M. Menth, R. Martin, and U. Spörlein, "Failure-Specific Self-Protecting Multipaths – Increased Capacity Savings or Overengineering?," in *International Workshop on Design of Reliable Communication Networks (DRCN)*, (La Rochelle, France), Oct. 2007.

[22] T. Cicic, A. F. Hansen, A. Kvalbein, M. Hartmann, R. Martin, and M. Menth, "Relaxed Multiple Routing Configurations for IP Fast Reroute," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, (Salvador, Bahia, Brazil), Apr. 2008.

— Journals —

[23] M. Menth and R. Martin, "Service Differentiation with MEDF Scheduling in TCP/IP Networks," *Computer Communications Journal*, vol. 29, pp. 812–819, Apr. 2006.

[24] M. Menth, R. Martin, and J. Charzinski, "Capacity Overprovisioning for Networks with Resilience Requirements," *ACM SIGCOMM Computer Communications Review*, vol. 36, pp. 87–98, Oct. 2006.

General References

[25] ISO, "ISO 8402: Quality Management and Quality Assurance – Vocabulary." URL: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=20115, 1994.

[26] ISO, "ISO 9000: Quality Management Systems – Fundamentals and Vocabulary." URL: http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=42180, 2005.

[27] Alliance for Telecommunication Industry Solutions (ATIS), "ATIS Telecom Glossary 2007 (ATIS-0100523.2007)." http://www.atis.org/glossary/, 2007.

[28] ITU-T, “Recommendation E.600: Terms and Definitions of Traffic Engineering.” http://www.itu.int/rec/T-REC-E.600/en, Mar. 1993.

[29] ITU-D Study Group 2, ed., *Handbook Teletraffic Engineering*. 2006.

[30] ITU-T, “Recommendation E.543: Grades of Service in Digital International Telephone Exchagnes.” http://www.itu.int/rec/T-REC-E.543/en, Nov. 1988.

[31] ITU-T, ed., *Handbook Quality of Service and Network Performance*. ITU, 2004.

[32] The ATM Forum Technical Committee, “Traffic Management Specification Version 4.0.” http://www.ipmplsforum.org/ftp/pub/approved-specs/af-tm-0056.000.pdf, Apr. 1996.

[33] 3rd Generation Partnership Project (3GPP), “3GPP TS 23.107 V7.1.0 Quality of Service (QoS) Concept and Architecture (Release 7).” http://www.3gpp.org/ftp/Specs/html-info/23107.htm, Sept. 2007.

[34] P. P. White, “RSVP and Integrated Services in the Internet: A Tutorial,” *IEEE Communications Magazine*, vol. 35, pp. 100–106, May 1997.

[35] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “RFC 2475: An Architecture for Differentiated Services.” http://www.rfc-editor.org/rfc/rfc2475.txt, Dec. 1998.

[36] S. Shenker and J. Wroclawski, “RFC 2216: Network Element Service Specification Template.” http://www.rfc-editor.org/rfc/rfc2216.txt, Sept. 1997.

[37] J. Wroclawski, “RFC 2211: Specification of the Controlled-Load Network Element Service.” http://www.rfc-editor.org/rfc/rfc2211.txt, Sept. 1997.

[38] S. Shenker, C. Partridge, and R. Guerin, "RFC 2212: Specification of Guaranteed Quality of Service." http://www.rfc-editor.org/rfc/rfc2212.txt, Sept. 1997.

[39] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "RFC 2205: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification." http://www.ietf.org/rfc/rfc2205.txt, Sept. 1997.

[40] ITU-T, "Recommendation G.1000: Communications Quality of Service: A Framework and Definitions." http://www.itu.int/rec/T-REC-G.1000/en, Nov. 2001.

[41] ITU-T, "Recommendation E.800: Terms and Definitions Related to Quality of Service and Network Performance Including Dependability." http://www.itu.int/rec/T-REC-E.800/en, Aug. 1994.

[42] ITU-T, "Recommendation P.862: Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs." http://www.itu.int/rec/T-REC-P.862/en, Feb. 2001.

[43] J. Tapolcai, P. Cholda, K. Wajda, A. Jaijszczyk, and D. Verchere, "Joint Quantification of Resilience and Quality of Service," in *IEEE International Conference on Communications (ICC)*, (Istanbul, Turkey), June 2006.

[44] D. Soldani, M. Li, and R. Cuny, eds., *QoS and QoE Management in UMTS Cellular Systems*. John Wiley & Sons, LTD, 2006.

[45] Cisco Systems, Inc., San Jose, CA, USA, *Cisco Express Forwarding Overview*, May 2004.

[46] Cisco Systems, Inc., San Jose, CA, USA, *How Does Load Balancing Work?*, Feb. 2005.

[47] Juniper Networks, Inc., Sunnyvale, CA, USA, *JUNOS^{TM} Software: Routing Protocols Configuration Guide – Release 9.0*, Feb. 2008.

[48] J. Moy, "RFC 2328: OSPF Version 2." http://www.rfc-editor.org/rfc/rfc2328.txt, Apr. 1998.

[49] Digital Equipment Corp., "RFC 1142: OSI IS-IS Intra-domain Routing Protocol." http://www.rfc-editor.org/rfc/rfc1142.txt, Feb. 1990.

[50] ISO, "ISO 10589: Intermediate System to Intermediate System Routing Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionless-Mode Network Service," 1992.

[51] D. Thaler and C. Hopps, "RFC 2991: Multipath Issues in Unicast and Multicast Next-Hop Selection." http://www.rfc-editor.org/rfc/rfc2991.txt, Nov. 2000.

[52] G. Schollmeier, J. Charzinski, A. Kirstädter, C. Reichert, K. J. Schrodi, Y. Glickman, and C. Winkler, "Improving the Resilience in IP Networks," in *IEEE Workshop on High Performance Switching and Routing (HPSR)*, (Torino, Italy), June 2003.

[53] A. Akella, S. Seshan, and A. Shaikh, "Multihoming Performance Benefits: An Experimental Evaluation of Practical Enterprise Strategies," in *USENIX Technical Conference*, (Boston, MA, USA), June 2004.

[54] D. K. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang, "Optimizing Cost and Performance for Multihoming," in *ACM SIGCOMM*, (Portland, OR, USA), Aug. 2004.

[55] V. Paxson, "End-to-End Internet Packet Dynamics," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 277–292, 1997.

[56] S. Bohacek, J. Hespanha, J. Lee, C. Lim, and K. Obraczka, "A New TCP for Persistent Packet Reordering," *IEEE/ACM Transactions on Networking*, vol. 14, Apr. 2006.

[57] J. C. R. Bennett, C. Partridge, and N. Shectman, "Packet Reordering is not Pathological Network Behavior," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 789–798, Dec. 1999.

[58] F. Baker, "RFC 1812: Requirements for IP Version 4 Routers." http://www.rfc-editor.org/rfc/rfc1812.txt, June 1995.

[59] M. Laor and L. Gendel, "The Effect of Packet Reordering in a Backbone Link on Application Throughput," *IEEE Network Magazine*, vol. 16, pp. 28–36, Sept. 2002.

[60] R. Ludwig and R. H. Katz, "The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions," *ACM SIGCOMM Computer Communications Review*, vol. 30, pp. 30–36, Jan. 2000.

[61] E. Blanton and M. Allman, "On Making TCP More Robust to Packet Reordering," *ACM SIGCOMM Computer Communications Review*, vol. 32, Jan. 2002.

[62] M. Zhang, B. Karp, S. Floyd, and L. Peterson, "RR-TCP: A Reordering-Robust TCP with DSACK," in *IEEE International Conference on Network Protocols (ICNP)*, (Atlanta, GA, USA), Nov. 2003.

[63] M. Shreedhar and G. Varghese, "Efficient Fair Queuing Using Deficit Round-Robin," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 375–385, June 1996.

[64] S. Rost and H. Balakrishman, "Rate-Aware Splitting of Aggregate Traffic," tech. rep., MIT, 2003.

[65] C. Estan and G. Varghese, "New Directions in Traffic Measurement and Accounting," in *ACM SIGCOMM*, (Pittsburgh, PA, USA), Aug. 2002.

[66] A. Zinin, *Cisco IP Routing, Packet Forwarding and Intra-domain Routing Protocols*, ch. 5.5.1. Addison Wesley, 2002.

[67] Z. Cao, Z. Wang, and E. Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing," in *IEEE Infocom*, (Tel Aviv, Israel), 2000.

[68] International Organization for Standardization (ISO), *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures*. ISO/IEC 13239:2002.

[69] International Telecommunication Union – Telecom Standardization (ITU-T), *Error-correcting procedures for DCEs using asynchronous-to-synchronous conversion*. Recommendation V.43 (03/02).

[70] W. N. Ross, "A Painless Guide to CRC Error Detection Algorithms." http://www.ross.net/crc/, May 1996.

[71] T. W. Chim, K. L. Yeung, and K.-S. Lui, "Traffic Distribution over Equal-Cost-Multi-Paths," *Computer Networks*, vol. 49, pp. 465–475, Nov. 2005.

[72] S. Sinha, S. Kandula, and D. Katabi, "Harnessing TCPs Burstiness using Flowlet Switching," in *$3rd$ ACM Workshop on Hot Topics in Networks (HotNets)*, (San Diego, CA), Nov. 2004.

[73] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic Load Balancing Without Packet Reordering," *ACM SIGCOMM Computer Communications Review*, vol. 37, pp. 53–62, Apr. 2007.

[74] I. Gojmerac, T. Ziegler, F. Ricciato, and P. Reichl, "Adaptive Multipath Routing for Dynamic Traffic Engineering," in *IEEE Globecom*, (San Francisco, CA, USA), Nov. 2003.

[75] S. Fischer, N. Kammenhuber, and A. Feldmann, "REPLEX - Dynamic Traffic Engineering Based on Wardrop Routing Policies," in *CoNEXT (formerly QoFIS, NGC, MIPS)*, (Lisboa, Portugal), Dec. 2006.

[76] G. Schollmeier, J. Charzinski, A. Kirstädter, C. Reichert, K. J. Schrodi, Y. Glickman, and C. Winkler, "Improving the Resilience in IP Networks,"

in *IEEE Workshop on High Performance Switching and Routing (HPSR)*, (Torino, Italy), June 2003.

[77] B. Augustin, T. Friedman, and R. Teixeira, "Measuring Load-balanced Paths in the Internet," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, (San Diego, CA, USA), Oct. 2007.

[78] B. Augustin, T. Friedman, and R. Teixeira, "Multipath Tracing with Paris Traceroute," in 5^{th} *IEEE Workshop on End-to-End Monitoring Techniques and Services (E2EMON)*, (Munich, Germany), May 2007.

[79] C. Hoogendoorn, K. Schrodi, M. Huber, C. Winkler, and J. Charzinski, "Towards Carrier-Grade Next Generation Networks," in *International Conference on Communication Technology (ICCT)*, (Beijing, China), Apr. 2003.

[80] M. Menth, J. Milbrandt, and S. Kopf, "Impact of Routing and Traffic Distribution on the Performance of Network Admission Control," in 9^{th} *IEEE Symposium on Computers and Communications (ISCC)*, (Alexandria, Egypt), pp. 883–890, June 2004.

[81] P. H. Fredette, "The Past, Present, and Future of Inverse Multiplexing," *IEEE Communications Magazine*, vol. 32, pp. 42–46, Apr. 1994.

[82] J. Duncanson, "Inverse Multiplexing," *IEEE Communications Magazine*, vol. 32, pp. 34–41, Apr. 1994.

[83] BONDING Consortium, *Interoperability Requirements for n x 56/64 kbit/s calls. Version 1.0*, Sept. 1992.

[84] K. Sklower, B. Lloyd, G. McGregor, D. Carr, and T. Coradetti, "RFC 1990: The PPP Multilink Protocol (MP)." http://www.rfc-editor.org/rfc/rfc1990.txt, Aug. 1996.

[85] H. Adiseshu, G. Parulkar, and G. Varghese, "A Reliable and Scalable Striping Protocol," *ACM SIGCOMM Computer Communications Review*, vol. 26, Aug. 1996.

[86] J.-Y. Jo, Y. Kim, H. J. Chao, and F. Merat, "Internet Traffic Load Balancing using Dynamic Hashing with Flow Volume," in *SPIE ITCom*, (Boston, MA, USA), Apr. 2002.

[87] The ATM Forum Technical Committee, *Inverse Multiplexing for ATM (IMA) Specification. Version 1.0*, July 1997. AF-PHY-0086.000.

[88] J. Frimmel, "Inverse Multiplexing: Tailor-made for ATM," *Telephony*, vol. 231, pp. 28–34, July 1996.

[89] F. M. Chiussi, D. A. Khotimsky, and S. Krishnan, "Advanced Frame Recovery in Switched Connection Inverse Multiplexing for ATM," in *IEEE International Conference on ATM (ICATM)*, (Colmar, France), June 1999.

[90] I. Keslassy, C.-S. Chang, N. McKeown, and D.-S. Lee, "Optimal Load Balancing," in *IEEE Infocom*, (Miami, FL, USA), Mar. 2005.

[91] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," *IEEE Transactions on Communications*, vol. 35, pp. 1347–1356, Dec. 1987.

[92] Y. Lee, J. Lou, J. Luo, and X. Shen, "An Efficient Packet Scheduling Algorithm With Deadline Guarantees for Input-Queued Switches," *IEEE/ACM Transactions on Networking*, vol. 15, pp. 212–225, Feb. 2007.

[93] S.-T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, "Matching Output Queueing with a Combined Input/Output-Queued Switch," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1030–1039, June 1999.

[94] C.-S. Chang, D.-S. Lee, and Y.-S. Jou, "Load Balanced Birkhoff-von Neumann Switches, Part I: One-stage Buffering," in *IEEE Workshop on High*

Performance Switching and Routing (HPSR), (Dallas, TX, USA), May 2001.

[95] C.-S. Chang, W.-J. Chen, and H.-Y. Huang, "On Service Guarantees for Input Buffered Crossbar Switches: A Capacity Decomposition Approach by Birkhoff and von Neumann," in *IEEE International Workshop on Quality of Service (IWQoS)*, (London, UK), June 1999.

[96] I. Keslassy, S.-T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown, "Scaling Internet Routers Using Optics," in *ACM SIGCOMM*, (Karlsruhe, Germany), Aug. 2003.

[97] C.-S. Chang, D.-S. Lee, and C.-M. Lien, "Load Balanced Birkhoff-von Neumann Switches, Part II: Multi-stage Buffering," *Computer Communications*, vol. 25, pp. 623–634, 2002.

[98] I. Keslassy and N. McKeown, "Maintaining Packet Order in Two-Stage Switches," in *IEEE Infocom*, (New York, NY, USA), June 2002.

[99] I. Keslassy, S.-T. Chuang, and N. McKeown, "A Load-Balanced Switch with an Arbitrary Number of Linecards," in *IEEE Infocom*, (Hong Kong), Mar. 2004.

[100] W. Shi, M. H. MacGregor, and P. Gburzynski, "Load Balancing for Parallel Forwarding," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 790–801, Aug. 2005.

[101] G. Dittmann and A. Herkersdorf, "Network Processor Load Balancing for High–Speed Links," in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, (San Diego, CA, USA), pp. 727–735, 2002.

[102] W. Shi, M. H. MacGregor, and P. Gburzynski, "A Scalable Load Balancer for Forwarding Internet Traffic: Exploiting Flow-level Burstiness,"

in *ACM/IEEE Symposium on Architecture for Networking and Communication Systems (ANCS)*, (Princeton, NJ, USA), Dec. 2005.

[103] W. Shi, M. H. MacGregor, and P. Gburzynski, "An Adaptive Load Balancer for Multiprocessor Routers," in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, (San Jose, CA, USA), pp. 671–679, July 2004.

[104] L. Kencl and J.-Y. Le Boudec, "Adaptive Load-Sharing for Network Processors," in *IEEE Infocom*, (New York, NY, USA), pp. 545–554, June 2002.

[105] W. Shi and L. Kencl, "Sequence-Preserving Adaptive Load Balancers," in *ACM/IEEE Symposium on Architecture for Networking and Communication Systems (ANCS)*, (San Jose, CA, USA), pp. 134–152, Dec. 2006.

[106] P. Savola and T. Chown, "A Survey of IPv6 Site Multihoming Proposals," in *International Conference of Telecommunications (ConTEL)*, (Zagreb, Croatia), June 2005.

[107] X. Liu and L. Xiao, "A Survey of Multihoming Technology in Stub Networks: Current Research and Open Issues," *IEEE Network Magazine*, vol. 21, pp. 32–40, June 2007.

[108] T. Bates and Y. Rekhter, "RFC 2260: Scalable Support for Multi-homed Multi-provider Connectivity." http://www.rfc-editor.org/rfc/rfc2260.txt, Jan. 1998.

[109] P. Srisuresh and K. Egevang, "RFC 3022: Traditional IP Network Address Translator (Traditional NAT)." http://www.rfc-editor.org/rfc/rfc3022.txt, Jan. 2001.

[110] F. Guo, J. Chen, W. Li, and T.-c. Chiueh, "Experiences in Building A Multihoming Load Balancing System," in *IEEE Infocom*, (Hong Kong), Mar. 2004.

[111] Y.-D. Lin, S.-C. Tsao, and U.-P. Leong, "On-the-Fly TCP Path Selection Algorithm in Access Link Load Balancing," in *IEEE Globecom*, (Dallas, TX, USA), Dec. 2004.

[112] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh, "A Comparison of Overlay Routing and Multihoming Route Control," in *ACM SIGCOMM*, (Portland, OR, USA), Aug. 2004.

[113] A. Dhamdhere and C. Dovrolis, "ISP and Egress Path Selection for Multihomed Networks," in *IEEE Infocom*, (Barcelona, Spain), pp. 2314–2325, Apr. 2006.

[114] C. de Launois and M. Bagnulo, "The Paths Towards IPv6 Multihoming," *IEEE Communication Surveys and Tutorials*, vol. 8, 2006.

[115] D. G. Thaler and C. V. Ravishankar, "Using Name-Based Mappings to Increase Hit Rates," *IEEE/ACM Transactions on Networking*, vol. 6, no. 1, 1998.

[116] K. W. Ross, "Hash-Routing for Collections of Shared Web Caches," *IEEE Network Magazine*, vol. 11, pp. 37–44, Nov. 1997.

[117] V. Cardellini and M. Colajanni, "Dynamic Load Balancing on Web-server Systems," *IEEE Internet Computing*, vol. 3, pp. 28–39, May 1999.

[118] L. Aversa and A. Bestavros, "Load Balancing a Cluster of Web Servers - Using Distributed Packet Rewriting," in *IEEE International Conference on Performance, Computing, and Communications (IPCCC)*, (Phoenix, AZ, USA), pp. 24–29, Feb. 2000.

[119] M.-S. Kim, M.-J. Choi, and J. W. Hong, "Highly Available and Efficient Load Cluster Management System using SNMP and Web," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, (Florence, Italy), Apr. 2002.

[120] S. R. Mohanty and L. N. Bhuyan, "Fair Scheduling over Multiple Servers with Flow-dependend Server Rate," in *IEEE Local Computer Networks*, (Tampa, FL, USA), Nov. 2006.

[121] D. Breitgand, R. Cohen, A. Nahir, and D. Raz, "On Fully Distributed Adaptive Load Balancing," in *IFIP/IEEE Workshop on Distributed Systems: Operations and Management (DSOM)*, (San Jose, CA, USA), Oct. 2007.

[122] A. D. Amis and R. Prakash, "Load-Balancing Clusters in Wireless Ad Hoc Networks," in *IEEE Symposium on Application-Specific Systems and Software Engineering Technology*, (Richardson, TX, USA), pp. 25–32, Mar. 2000.

[123] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi, "On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks," in 1^{st} *ACM International Symposium on Mobile Ad-Hoc Networking and Computing*, (Boston, MA, USA), pp. 3–10, Aug. 2000.

[124] Y. Ganjali and A. Keshavarzian, "Load Balancing in Ad Hoc Networks: Single-Path Routing vs. Multi-Path Routing," in *IEEE Infocom*, (Hong Kong), Mar. 2004.

[125] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica, "Load Balancing in Structured P2P Systems," in *International Workshop on Peer-to-Peer Systems (IPTPS)*, (Berkeley, CA, USA), Feb. 2003.

[126] D. R. Karger and M. Ruhl, "Simple Efficient Load Balancing Algorithms for Peer-to-Peer Systems," in 16^{th} *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, (Barcelona, Spain), June 2004.

[127] J. Byers, J. Considine, and M. Mitzenmacher, "Simple Load Balancing for Distributed Hash Tables," in *International Workshop on Peer-to-Peer Systems (IPTPS)*, (Berkeley, CA, USA), Feb. 2003.

[128] P. B. Godfrey and I. Stoica, "Heterogeneity and Load Balance in Distributed Hash Tables," in *IEEE Infocom*, (Miami, FL, USA), Mar. 2005.

[129] O. K. Tonguz and E. Yanmaz, "On the Theory of Dynamic Load Balancing," in *IEEE Globecom*, (San Francisco, CA, USA), Dec. 2003.

[130] E. Yanmaz, O. K. Tonguz, and R. Rajkumar, "Is There an Optimum Dynamic Load Balancing Scheme?," in *IEEE Globecom*, (St. Louis, MO, USA), Nov. 2005.

[131] G. Iannaccone, C.-n. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of Link Failures in an IP backbone," in *ACM SIGCOMM Internet Measurement Workshop (IMW)*, (Marseille, France), pp. 237–242, Nov. 2002.

[132] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving Sub-Second IGP Convergence in Large IP Networks," *ACM SIGCOMM Computer Communications Review*, vol. 35, pp. 35–44, July 2005.

[133] O. Bonaventure, C. Filsfils, and P. Francois, "Achieving Sub-50 Milliseconds Recovery Upon BGP Peering Link Failures," in *CoNEXT (formerly QoFIS, NGC, MIPS)*, (Paris, France), July 2005.

[134] A. Basu and J. G. Riecke, "Stability Issues in OSPF Routing," in *ACM SIGCOMM*, (San Diego, CA, USA), pp. 225–236, Aug. 2001.

[135] Nokia, "The Five Nines IP Network." http://nds2.ir.nokia.com/NOKIA_COM_1/About_Nokia/Press/White_Papers/pdf_files/5_Nines_IP_Network_net.pdf, Jan. 2001.

[136] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of Failures in an IP Backbone," in *IEEE Infocom*, (Hong Kong), Mar. 2004.

[137] P. Pan, G. Swallow, and A. Atlas, “RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels.” http://www.rfc-editor.org/rfc/rfc4090.txt, May 2005.

[138] M. Shand and S. Bryant, “IP Fast Reroute Framework.” http://www.ietf.org/internet-drafts/draft-ietf-rtgwg-ipfrr-framework-08.txt, Feb. 2008. Expires Aug. 28 2008 (work in progress).

[139] A. Atlas and A. Zinin, “Basic Specification for IP Fast-Reroute: Loop-free Alternates.” http://www.ietf.org/internet-drafts/draft-ietf-rtgwg-ipfrr-spec-base-12.txt, Mar. 2008. Expires Sep. 28, 2008 (work in progress).

[140] S. Bryant, M. Shand, and P. S., “IP Fast Reroute Using Not-via Addresses.” http://www.ietf.org/internet-drafts/draft-ietf-rtgwg-ipfrr-notvia-addresses-02.txt, Feb. 2009. Expires Aug. 28, 2008 (work in progress).

[141] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, “Fast IP Network Recovery using Multiple Routing Configurations,” in *IEEE Infocom*, (Barcelona, Spain), Apr. 2006.

[142] 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, pp. 359–372, June 2007.

[143] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery*. Morgan Kaufmann / Elsevier, 1. ed., 2004.

[144] B. Mukherjee, *Optical WDM Networks*. Springer, 2. ed., 2006.

[145] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot, “IGP Link Weight Assignment for Transient Link Failures,” in 18^{th} *International Teletraffic Congress (ITC)*, (Berlin, Germany), Sept. 2003.

[146] B. Fortz and M. Thorup, “Robust Optimization of OSPF/IS-IS Weights,” in *International Network Optimization Conference (INOC)*, (Paris, France), pp. 225–230, Oct. 2003.

[147] M. Menth, *Efficient Admission Control and Routing in Resilient Communication Networks*. PhD thesis, University of Würzburg, Faculty of Computer Science, Am Hubland, July 2004.

[148] A. Raj and O. C. Ibe, "A Survey of IP and Multiprotocol Label Switching Fast Reroute Schemes," *Computer Networks*, vol. 51, pp. 1882–1907, June 2007.

[149] L. Jorge and T. Gomes, "Survey of Recovery Schemes in MPLS Networks," in *Conference on Dependability of Computer Systems (DEPCOS-RELCOMEX)*, (Szklarska Poreba, Poland), May 2006.

[150] J.-P. Vasseur and S. Sivabalan, "RFC 4561: Definition of a Record Route Object (RRO) Node-Id Sub-Object." http://www.rfc-editor.org/rfc/rfc4561.txt, June 2006.

[151] A. Farrel, A. Ayyangar, and J. Vasseur, "RFC 5151: Inter-Domain MPLS and GMPLS Traffic Engineering – Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions." http://www.rfc-editor.org/rfc/rfc5151.txt, Feb. 2008.

[152] R. Cetin, T. D. Nadeau, and A. S. K. Koushik, "Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base for Fast Reroute." http://www.ietf.org/internet-drafts/draft-ietf-mpls-fastreroute-mib-08.txt, Nov. 2007. Expires May 2008 (work in progress).

[153] Cisco Systems, Inc., San Jose, CA, USA, *MPLS Traffic Engineering (TE) – Fast Reroute (FRR) Link and Node Protection*, Dec. 2006.

[154] Juniper Networks, Inc., Sunnyvale, CA, USA, *JUNOS^{TM} Software: MPLS Applications Configuration Guide – Release 9.0*, Feb. 2008.

[155] R. Aggarwal, D. Papadimitriou, and S. Yasukawa, "RFC 4875: Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE)

for Point-to-Multipoint TE Label Switched Paths (LSPs).” http://www.rfc-editor.org/rfc/rfc4875.txt, May 2007.

[156] J. L. Le Roux, R. Aggarwal, J. P. Vasseur, and M. Vigoureux, “P2MP MPLS-TE Fast Reroute with P2MP Bypass Tunnels.” http://www.ietf.org/internet-drafts/draft-ietf-mpls-p2mp-te-bypass-02.txt, Mar. 2008. Expires Aug. 2008 (work in progress).

[157] H. Saito and M. Yoshida, “An Optimal Recovery LSP Assignment Scheme for MPLS Fast Reroute,” *International Telecommunication Network Strategy and Planning Symposium (Networks)*, pp. 229–234, 2002.

[158] S. Balon, L. Mélon, and G. Leduc, “A Scalable and Decentralized Fast-Rerouting Scheme with Efficient Bandwidth Sharing,” *Computer Networks*, vol. 50, pp. 3043–3063, Nov. 2006.

[159] K. Kar and M. Kodialam, “Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications,” *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 2566–2579, Dec. 2000.

[160] M. Kodialam and T. V. Lakshman, “Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels using Aggregated Link Usage Information,” in *IEEE Infocom*, (Anchorage, Alaska), Apr. 2001.

[161] D. Wang and G. Li, “Efficient Distributed Solution for MPLS Fast Reroute,” in 4^{rd} *IFIP-TC6 Networking Conference (Networking)*, (Waterloo, ON, Canada), May 2005.

[162] M. Alicherry and R. Bhatia, “Pre-Provisioning Networks to Support Fast Restoration with Minimum Over-Build,” in *IEEE Infocom*, 2004.

[163] M. Tacca, K. Wu, and A. Fumagalli, “Local Detection and Recovery from Multi-Failure Patterns in MPLS-TE Networks,” in *IEEE International Conference on Communications (ICC)*, (Istanbul, Turkey), June 2006.

[164] G. Li, D. Wang, and R. Doverspike, "Efficient Distributed MPLS P2MP Fast Reroute," in *IEEE Infocom*, (Barcelona, Spain), Apr. 2006.

[165] D. W. Hong, C. S. Hong, and W.-S. Kim, "A Segment-based Protection Scheme for MPLS Network Survivability," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, (Vancouver, Canada), Apr. 2006.

[166] L. Li, M. M. Buddhikot, C. Chekuri, and K. Guo, "Routing Bandwidth Guaranteed Paths with Local Restoration in Label Switched Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 437–449, Feb. 2005.

[167] Z. Zhong, S. Nelakuditi, Y. Yu, S. Lee, J. Wang, and C.-N. Chuah, "Failure Inferencing based Fast Rerouting for Handling Transient Link and Node Failures," in *IEEE Global Internet Symposium*, (Miami, Fl, USA), pp. 2859–2863, Mar. 2005.

[168] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, "Fast IP Network Recovery using Multiple Routing Configurations," in *IEEE Infocom*, (Barcelona, Spain), pp. 23–29, Apr. 2006.

[169] S. Bryant, C. Filsfils, S. Previdi, and M. Shand, "IP Fast Reroute Using Tunnels." http://www.ietf.org/internet-drafts/draft-bryant-ipfrr-tunnels-03.txt, Nov. 2007. Expired May 19, 2008 (work in progress).

[170] S. Nelakuditi, S. Lee, Y. Yu, and Z.-L. Zhang, "Failure Insensitive Routing for Ensuring Service Availability," in *IEEE International Workshop on Quality of Service (IWQoS)*, 2003.

[171] S. Lee, Y. Yu, S. Nelakuditi, Z.-L. Zhang, and C.-N. Chuah, "Proactive vs. Reactive Approaches to Failure Resilient Routing," in *IEEE Infocom*, (Hong Kong), Mar. 2004.

[172] J. Wang and S. Nelakuditi, "IP Fast Reroute with Failure Inferencing," in *ACM SIGCOMM Workshop on Internet Network Management (INM)*, (Kyoto, Japan), Aug. 2007.

[173] J. Wang, Z. Zhong, and S. Nelakuditi, "Handling Multiple Network Failures through Interface Specific Forwarding," in *IEEE Globecom*, (San Francisco, CA, USA), Nov. 2006.

[174] G. Apostolopoulos, "Using Multiple Topologies for IP-only Protection Against Network Failures: A Routing Performance Perspective," Tech. Rep. TR377, Institute of Computer Science (ICS) of the Foundation for Research and Technology - Hellas (FORTH), Heraklion, Crete, Greece, 2006. http://www.ics.forth.gr/ftp/tech-reports/2006/2006.TR377_Routing_Performance_Perspective.pdf.

[175] P. Psenak, S. Mirtorabi, A. Roy, L. Nguyen, and P. Pillay-Esnault, "RFC 4915: Multi-Topology (MT) Routing in OSPF." http://www.rfc-editor.org/rfc/rfc4915.txt, June 2007.

[176] N. Rawat, R. Shrivastava, and D. Kushi, "OSPF Version 2 MIB for Multi-Topology (MT) Routing." http://www.ietf.org/internet-drafts/draft-ietf-ospf-mt-mib-02.txt, Apr. 2008. Expires Oct. 4, 2008 (work in progress).

[177] T. Przygienda, N. Shen, and N. Sheth, "RFC 5120: M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)." http://www.rfc-editor.org/rfc/rfc5120.txt, Feb. 2008.

[178] A. F. Hansen, O. Lysne, T. Cicic, and S. Gjessing, "Fast Proactive Recovery from Concurrent Failures," in *IEEE International Conference on Communications (ICC)*, (Glasgow, UK), June 2007.

[179] U. Hengartner, S. Moon, and C. Diot, "Detection and Analysis of Routing Loops in Packet Traces," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, (Marseilles, France), Nov. 2002.

[180] S. Bryant and M. Shand, "A Framework for Loop-free Convergence." http:// www.ietf.org /internet-drafts /draft-ietf- rtgwg-lf-conv-frmwk-02.txt, Feb. 2008. Expires Aug. 17, 2008 (work in progress).

[181] P. Francois, O. Bonaventure, M. Shand, S. Bryant, and S. Previdi, "Loop-Free Convergence Using Order FIB Updates." http://www.ietf.org /internet-drafts/draft-ietf-rtgwg-ordered-fib-02.txt, Feb. 2008. Expires Aug. 28, 2008 (work in progress).

[182] P. Francois and O. Bonaventure, "Avoiding Transient Loops during IGP Convergence in IP Networks," in *IEEE Infocom*, (Miami, Florida), Mar. 2005.

[183] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Dynamics of Hot-Potato Routing in IP Networks," in *ACM SIGMETRICS*, (New York City, NY, USA), June 2004.

[184] R. Teixeira, A. Shaikh, T. Griffin, and G. M. Voelker, "Network Sensitivity to Hot-Potato Disruptions," in *ACM SIGCOMM*, (Portland, OR, USA), Aug. 2004.

[185] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS Weights in a Changing World," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 756–767, May 2002.

[186] A. Iselt, K. Andreas, A. Pardigon, and T. Schwabe, "Resilient Routing Using MPLS and ECMP," in *IEEE Workshop on High Performance Switching and Routing (HPSR)*, (Phoenix, AZ, USA), Apr. 2004.

[187] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An Approach to Alleviate Link Overload as Observed on an IP Backbone," in *IEEE Infocom*, (San Francisco, CA, USA), Apr. 2003.

[188] A. Sridharan, S. B. Moon, and C. Diot, "On the Correlation between Route Dynamics and Routing Loops," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, (Miami, FL, USA), Oct. 2003.

[189] P. Francois and O. Bonaventure, "An Evaluation of IP-based Fast Reroute Techniques," in *CoNEXT (formerly QoFIS, NGC, MIPS)*, (Toulouse, France), pp. 244–245, Oct. 2005.

[190] A. F. Hansen, T. Cicic, and S. Gjessing, "Alternative Schemes for Proactive IP Recovery," in 2^{nd} *Conference on Next Generation Internet Design and Engineering (NGI)*, (Valencia, Spain), Apr. 2006.

[191] M. Gjoka, V. Ram, and X. Yang, "Evaluation of IP Fast Reroute Proposals," in *IEEE International Conference on COMmunication System softWAre and MiddlewaRE (COMSWARE)*, (Bangalore, India), Jan. 2007.

[192] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads," in *ACM SIGMETRICS*, (San Diego, CA, USA), June 2003.

[193] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An Information-Theoretic Approach to Traffic Matrix Estimation," in *ACM SIGCOMM*, (Karlsruhe, Germany), Aug. 2003.

[194] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving Traffic Demands for Operational IP Networks: Methodology and Experience," in *ACM SIGCOMM*, (Stockholm, Sweden), Aug. 2000.

[195] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic Matrix Estimation on a Large IP Backbone - A Comparison on Real Data," in *ACM SIGCOMM Internet Measurement Workshop (IMW)*, (Taormina, Sicily, Italy), pp. 149–160, Oct. 2004.

[196] A. Medina, N. Taft, S. K., B. S., and D. C., "Traffic Matrix Estimation: Existing Techniques and New Directions," in *ACM SIGCOMM*, (Pittsburgh, PA, USA), Aug. 2002.

[197] K. Papagiannaki, N. Taft, and A. Lakhina, "A Distributed Approach to Measure IP Traffic Matrices," in *ACM SIGCOMM Internet Measurement Workshop (IMW)*, (Taormina, Sicily, Italy), pp. 161–174, Oct. 2004.

[198] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, "Traffic Matrices: Balancing Measurements, Inference and Modeling," in *ACM SIGMETRICS*, (Banff, AL, Canada), June 2005.

[199] V. Erramilli, M. Crovella, and N. Taft, "An Independent-Connection Model for Traffic Matrices," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, (Rio de Janeiro, Brazil), Oct. 2006.

[200] F. P. Kelly, *Stochastic Networks: Theory and Applications*, vol. 4, ch. 8 Notes on Effective Bandwidths, pp. 141–168. Oxford University Press, 1996.

[201] R. J. Gibbens and P. J. Hunt, "Effective Bandwidth for the Multitype UAS Channel," *Queuing Systems*, vol. 16, pp. 17–27, Oct. 1991.

[202] G. Kesidis, J. Walrand, and C.-S. Chang, "Effective Bandwidths for Multiclass Markov Fluids and Other ATM Sources," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 424–428, Aug. 1993.

[203] S. L. Spitler and D. C. Lee, "Integrating Effective-Bandwidth-Based QoS Routing and Best Effort Routing," in *IEEE Infocom*, (San Francisco, CA, USA), Mar. 2003.

[204] N. X. Liu and J. S. Baras, "Measurement and Simulation Based Effective Bandwidth Estimation," in *IEEE Globecom*, (Dallas, TX, USA), Nov. 2004.

[205] A. Davy, D. Botvich, and B. Jennings, "Process for QoS-Aware IP Network Planning Using Accounting Data and Effective Bandwidth Estimation," in *IEEE Globecom*, (Washington, DC, USA), pp. 2690–2695, Nov. 2007.

[206] S. Shenker, "Fundamental Design Issues for the Future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1176–1188, Sept. 1995.

[207] A. Pras, R. van de Meent, and M. Mandjes, "QoS in Hybrid Networks – An Operator's Perspective," in 13^{th} *IEEE International Workshop on Quality of Service (IWQoS)*, (Passau, Germany), June 2005.

[208] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, "Packet-Level Traffic Measurements from the Sprint IP Backbone," *IEEE Network Magazine*, vol. 17, pp. 6–16, Nov. 2003.

[209] A. Odlyzko, "Data Networks are Lightly Utilized, and will Stay that Way," *Review of NEtwork Economics*, vol. 2, no. 3, pp. 210–237, 2003.

[210] L. Breslau and S. Shenker, "Best-Effort versus Reservations: A Simple Comparative Analysis," in *ACM SIGCOMM*, (Vancouver, BC, Canada), pp. 3–16, Aug. 1998.

[211] O. Heckmann and J. Schmitt, "Best-Effort versus Reservations Revisited," in 13^{th} *IEEE International Workshop on Quality of Service (IWQoS)*, (Passau, Germany), June 2005.

[212] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A Nonstationary Poisson View of Internet Traffic," in *IEEE Infocom*, (Hong Kong, China), pp. 1558–1569, Mar. 2004.

[213] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Transactions on Networking*, vol. 2, Feb. 1994.

[214] V. Paxson and S. Floyd, "Wide-Are Traffic: The Failure of Poisson Modelling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, June 1995.

[215] M. E. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 835–846, Dec. 1997.

[216] M. Grossglauser and J.-C. Bolot, "On the Relevance of Long-Range Dependence in Network Traffic," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 629–640, Oct. 1999.

[217] I. Cao and K. Ramanan, "A Poisson Limit for Buffer Overflow Probabilities," in *IEEE Infocom*, (New York, NY, USA), June 2002.

[218] Z.-L. Zhang, V. J. Riberio, S. Moon, and C. Diot, "Small-Time Scaling Behaviors of Internet Backbone Traffic: An Empirical Study," in *IEEE Infocom*, (San Francisco, CA), Apr. 2003.

[219] C. Fraleigh, F. Tobagi, and C. Diot, "Provisioning IP Backbone Networks to Support Latency Sensitive Traffic," in *IEEE Infocom*, (San Francisco, CA, USA), Mar. 2003.

[220] R. van de Meent, M. Mandjes, and A. Pras, "Gaussian Traffic Everywhere?," in *IEEE International Conference on Communications (ICC)*, (Istanbul, Turkey), June 2006.

[221] R. van de Meent, A. Pras, M. Mandjes, H. van den Berg, and L. Nieuwenhuis, "Traffic Measurement for Link Dimensioning: A Case Study," in *DSOM*, (Heidelberg, Germany), pp. 106–117, Oct. 2003.

[222] H. van den Berg, M. Mandjes, and R. van de Meent, "QoS-Aware Bandwidth Provisioning for IP Network Links," *Computer Networks*, vol. 50, pp. 631–647, Apr. 2006.

[223] R. van de Meent and M. Mandjes, "Evaluation of 'User-Oriented' and 'Black-Box' Traffic Models for Link Provisioning," in 1^{st} *Conference on Next Generation Internet Design and Engineering (NGI)*, (Rome, Italy), Apr. 2005.

[224] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-Term Forecasting of Internet Backbone Traffic: Observations and Initial Models," in *IEEE Infocom*, (San Francisco, CA), Apr. 2003.

[225] H. Tuan Tran and T. Ziegler, "Adaptive Bandwidth Provisioning with Explicit Respect to QoS Requirements," in *International Workshop on Quality of future Internet Services (QofIS)*, (Stockholm, Sweden), 2003.

[226] J. Choe and N. B. Shroff, "A Central-Limit-Theorem-Based Approach for Analyzing Queue Behavior in High-Speed Networks," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 659–671, Oct. 1998.

[227] J. Roberts, U. Mocci, and J. Virtamo, *Broadband Network Teletraffic - Final Report of Action COST 242*. Berlin, Heidelberg: Springer, 1996. ISBN 3-540-61815-5.

[228] S. Jamin, P. Danzig, S. J. Shenker, and L. Zhang, "Measurement-Based Admission Control Algorithms for Controlled-Load Services Packet Networks," in *ACM SIGCOMM*, (Cambridge, MA, USA), Aug. 1995.

[229] M. Grossglauser and N. C. Tse, David, "A Framework for Robust Measurement-Based Admission Control," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 293–309, June 1999.

[230] M. Menth, J. Milbrandt, and S. Oechsner, "Experience-Based Admission Control (EBAC)," in *IEEE Symposium on Computers and Communications (ISCC)*, (Alexandria, Egypt), pp. 903–910, June 2004.

[231] M. Menth and F. Lehrieder, "Performance Evaluation of PCN-Based Admission Control," in *International Workshop on Quality of Service (IWQoS)*, (Enschede, Netherlands), June 2008.

[232] G. van Hoey, D. de Vleeschauwer, B. Steyaert, V. Ingelbrecht, and H. Brunel, "Benefit of Admission Control in Aggregation Network Dimensioning for Video Services," in *IFIP-TC6 Networking Conference (Networking)*, (Athens, Greece), pp. 357–368, May 2004.

[233] J. W. Roberts, "Traffic Theory and the Internet," *IEEE Communications Magazine*, vol. 39, pp. 94–99, Jan. 2001.

[234] T. Dinh, B. Sonkoly, and S. Molnár, "Fractal Analysis and Modeling of VoIP Traffic," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Vienna, Austria), pp. 123–130, June 2004.

[235] S. Sarvotham, R. Riedi, and R. Baraniuk, "Connection-level analysis and modeling of network traffic," in *ACM SIGCOMM Workshop on Internet Network Management (INM)*, (San Fransisco, CA, USA), pp. 99–103, Nov. 2001.

[236] K. C. Claffy and N. Brownlee, "Understanding Internet Traffic Streams: Dragonflies and Tortoises," *IEEE Communications Magazine*, vol. 40, pp. 110–117, Oct. 2002.

[237] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*. McGraw-Hill, 3^{rd} ed., 2000.

[238] M. Menth, J. Milbrandt, and A. Reifert, "Sensitivity of Backup Capacity Requirements to Traffic Distribution and Resilience Constraints," in 1^{st} *Conference on Next Generation Internet Design and Engineering (NGI)*, (Rome, Italy), Apr. 2005.

[239] B. M. Waxman, "Routing of Multipoint Connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1617–1622, Dec. 1988.

[240] E. W. Zegura, K. L. Calvert, and M. J. Bonahoo, "A Quantitative Comparison of Graph-Based Models for Internet Topology," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 770–783, Dec. 1997.

[241] P. Francois, M. Shand, and O. Bonaventure, "Disruption-Free Topology Reconfiguration in OSPF Networks," in *IEEE Infocom*, (Anchorage, Alaska, USA), May 2007.

[242] A. Nucci, A. Sridharan, and N. Taft, "The Problem of Synthetically Generating IP Traffic Matrices: Initial Recommendations," *ACM SIGCOMM Computer Communications Review*, vol. 35, pp. 19–32, July 2005.

[243] M. Roughan, "Simplifying the Synthesis of Internet Traffic Matrices," *ACM SIGCOMM Computer Communications Review*, vol. 35, pp. 93–96, Oct. 2005.

[244] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, "On the Nonstationarity of Internet Traffic," in *ACM SIGMETRICS*, (Cambridge, MA, USA), June 2001.

[245] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, *Nonlinear Estimation and Classification*, ch. Internet Traffic Tends Toward Poisson and Independent as the Load Increases. New York, NY: Springer, 2002.

[246] K. W. Ross and D. H. K. Tsang, "The Stochastic Knapsack Problem," *IEEE/ACM Transactions on Networking*, vol. 37, pp. 740–747, July 1989.

[247] T. Schwabe and C. G. Gruber, "Traffic Variations Caused by Inter-domain Re-routing," in *International Workshop on the Design of Reliable Communication Networks (DRCN)*, (Ischia Island, Italy), Oct. 2005.

ISSN 1432-8801