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

Efficient Admission Control and Routing for Resilient Communication Networks

Michael Menth

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Bericht 03/04

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

Efficient Admission Control and Routing for Resilient Communication Networks

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

vorgelegt von

Michael Menth

aus

Würzburg

Würzburg 2004

Eingereicht am: 18.05.2004
bei der Fakultät für Mathematik und Informatik
1. Gutachter: Prof. Dr.-Ing. P. Tran-Gia
2. Gutachter: Prof. Dr.-Ing. R. Steinmetz
Tag der mündlichen Prüfung: 08.07.2004

Danksagung

An dieser Stelle möchte ich allen danken, die mich während meiner Zeit als Doktorand begleitet und direkt oder indirekt unterstützt haben.

In besonderer Weise gilt mein Dank meinem Doktorvater Prof. Dr.-Ing. Phuoc Tran-Gia, der an seinem Lehrstuhl einen fruchtbaren Boden für kreatives Arbeiten über aktuelle Technologien geschaffen hat. Die stimulierende Umgebung bestand vor allem in einer Mischung aus wissenschaftlich höchst interessanten Industrieprojekten und dem direkten Zugang zur internationalen Forschergemeinde. Kernkompetenzen wie die angemessene Präsentation eigener Arbeiten und das Erkennen deren Potentials sind wichtige Ergebnisse seiner Schule.

Ich bin Herrn Prof. Dr.-Ing. Ralf Steinmetz zu großem Dank verpflichtet, weil er sich trotz zeitlicher Engpässe sofort bereit erklärte, das Zweitgutachten für meine Doktorarbeit zu übernehmen, und durch seine zügige Korrektur einen raschen Abschluss meines Promotionsverfahrens ermöglichte.

Meinen Dank möchte ich auch Prof. Dr. Klaus Schilling und Prof. Dr. Dietmar Seipel aussprechen, die neben meinem Doktorvater als Prüfer für meine Disputation fungierten. Immerhin bedeutete die Beschäftigung mit dem für Sie fachfremden Stoff eine Zusatzarbeit im Vorfeld der Prüfung, noch dazu in der Hektik des ausklingenden Sommersemesters.

Die gute Atmosphäre am Lehrstuhl für Verteilte Systeme ist in hohem Maße meinen Kollegen – Andreas Binzenhöfer, Dr. Mathias Dümmler, Klaus Heck, Robert Henjes, Tobias Hoßfeld, Stefan Köhler, Dr. Kenji Leibnitz, An-

dreas Mäder, Rüdiger Martin, Jens Milbrandt, Dr. Vu Phan-Gia, Rastin Pries, Dr. Oliver Rose, Dirk Staehle, Dr. Kurt Tutschku, Dr. Norbert Vicari und Patricia Wilcox – zuzuschreiben, weil ihr kooperatives und hilfsbereites Verhalten sowie ihr soziales Engagement den Arbeitsplatz zu mehr als einem Ort der Arbeit werden ließ. Ebenso habe ich die Einbindung in die Fakultät als Fachstudienberater Informatik als sehr bereichernd empfunden, weil diese Funktion den angenehmen Kontakt mit vielen Professoren der Fakultät, der Fachschaft und den Studenten aller Semester förderte.

Als wissenschaftlicher Mitarbeiter arbeitete ich viel mit studentischen Hilfskräften, Praktikanten, Studienarbeitern und Diplomanden (Frithjof Eckart, Sebastian Gehrsitz, Susanne Halstead, Matthias Hartmann, Norbert Hauck, Jan Junker, Stefan Kopf, Hans-Carl Oberdalhoff, Thomas Obeth, Simon Oechsner, Andreas Reifert, Florian Zeiger, Andreas Völker und Tom Wirth) zusammen und möchte mich bei dieser Gelegenheit für ihren Einsatz bei der Vorbereitung von Lehrveranstaltungen und für ihre Beiträge zur Forschung recht herzlich bedanken, denn ich war oft auf ihre Verlässlichkeit und Genauigkeit angewiesen. An dieser Stelle gebührt mein Dank auch Frau Alt, die uns als Sekretärin des Lehrstuhls die tägliche Arbeit durch ihr schnelles und zuverlässiges Wirken erleichterte.

Der Erfolg von Projekten hängt zum großen Teil von der Expertise und Einsatzbereitschaft der Kooperationspartner ab und ich bin froh, stets gute Erfahrungen in dieser Hinsicht gemacht zu haben. Darum danke ich Prof. Dr. Stefan Schneeberger, Dr. Herbert Heiß, Thomas Reim und Matthias Schmid, mit denen ich ein zweijähriges Projekt bei Siemens ICM bestritt, sowie Prof. Dr. Cornelis Hoogendoorn, Dr. Joachim Charzinski, Dr. Nils Heldt, Dr. Chris Winkler und Karl Schrodi, stellvertretend für noch viele andere Kollegen bei Siemens ICN, Siemens CT und an den Partnerinstituten im KING Projekt, das die Grundlage für den ersten Teil meiner Arbeit lieferte. Als Verantwortlicher für das Midterm-Seminar des EU-Projektes COST-279 möchte ich den vielen internationalen Helfern danken, die durch ihren bereitwilligen Beitrag den Zwischenbericht zu einem gemeinschaftlichen Erfolg werden ließen.

Schließlich möchte ich auch jene – sofern noch nicht erwähnt – aufzählen,

die durch fachkundige Diskussionen zum Gelingen meiner Arbeit beigetragen haben: Markus Breitenbach, Ulrich Ehrenberger, Thomas Engel, Dr. Gerhard Haßlinger, Prof. Dr. Villy B. Iversen, Prof. Dr. Edward Knightly, Prof. Dr. Udo Krieger, Prof. Dr. Michel Mandjes, Prof. Dr. Michal Pióro, Dr. Oliver Pfaffenzeller und Dr. Jim Roberts.

Zum Schluss danke ich meiner Freundin Heike Schebler für ihr Verständnis. Sie musste nicht nur während ausgedehnter Konferenzreisen auf meine Anwesenheit verzichten, sondern mich auch oft in der Freizeit mit der Wissenschaft teilen. Mein abschließender Dank richtet sich an meine Eltern. Sie haben mir die Möglichkeit gegeben, mich meinen Neigungen entsprechend entwickeln zu können, mir wichtige Werte im Leben vermittelt und mich immer bei meinen Zielen unterstützt.

Acknowledgements

At this occasion I would like to express my gratitude to all who accompanied me during my time as a PhD student and supported me either directly or indirectly.

I owe special thanks to my supervisor Prof. Dr.-Ing. Phuoc Tran-Gia who provided at his department a fertile soil for creative work on current technologies. The stimulating environment consisted in a combination of scientifically highly interesting industry projects and the contact to the international research community. Skills like the adequate presentation of one's own work and the recognition of its potential are important results of the education by him.

I am obliged to Prof. Dr.-Ing. Ralf Steinmetz because he accepted immediately to serve as second supervisor for my thesis in spite of his busy schedule. His fast review completion speeded up my graduation considerably.

Prof. Dr. Klaus Schilling and Prof. Dr. Dietmar Seipel acted as examiners in my disputation beside my supervisor. I am grateful to them since the study of my external work imposed an additional workload on them in the rush of the ending summer semester.

The friendly atmosphere at the Department of Distributed Systems can be accredited to a high degree to my colleagues – Andreas Binzenhöfer, Dr. Mathias Dümmler, Klaus Heck, Robert Henjes, Tobias Hoßfeld, Stefan Köhler, Dr. Kenji Leibnitz, Andreas Mäder, Rüdiger Martin, Jens Milbrandt, Dr. Vu Phan-Gia, Rastin Pries, Dr. Oliver Rose, Dirk Staehle, Dr. Kurt Tutschku, Dr. Norbert Vicari, and Patricia Wilcox. Their cooperation, their willingness to help, and their

social engagement turned the department into more than a simple place of work.

In the same way, I enjoyed my integration into the faculty as student's advisor for many years. This function fostered the contact with many professors of the faculty, student representatives, and students through all semesters.

As a research and teaching assistant I often cooperated with student assistants, laboratory students, and Master students – Frithjof Eckart, Sebastian Gehrsitz, Susanne Halstead, Matthias Hartmann, Norbert Hauck, Jan Junker, Stefan Kopf, Hans-Carl Oberdalhoff, Thomas Obeth, Simon Oechsner, Andreas Reifert, Florian Zeiger, Andreas Völker, and Tom Wirth. I would like to thank all of them for their effort in assisting me with the preparation of lectures and exercises, and for their contribution to my research work because I often depended on their reliability and accuracy. At this moment, my thanks also belongs to our secretary Mrs. Alt because she supported our daily work by her fast and diligent help.

The success of projects depends to a major degree on the expertise and the commitment of the partners and I am glad to have made only positive experience in this respect. Therefore, I thank Prof. Dr. Stefan Schneeberger, Dr. Herbert Heiß, Thomas Reim, and Matthias Schmid with whom I worked on a two-years project at Siemens ICM. I thank also Prof. Dr. Cornelis Hoogendoorn, Dr. Joachim Charzinski, Dr. Nils Heldt, Dr. Chris Winkler, and Karl Schrodi representative for the many colleagues at Siemens ICN and the partner institutes in the KING project, which served as the basis for the first part of my work. As co-organizer for the midterm seminar of the European project COST-279 I would like to say thanks to the many international researchers who made the midterm report a success by their voluntary contribution.

Finally, I would also like to name those – unless not yet mentioned – who contributed through competent discussions and advice to the success of this work: Markus Breitenbach, Ulrich Ehrenberger, Thomas Engel, Dr. Gerhard Haßlinger, Prof. Dr. Villy B. Iversen, Prof. Dr. Edward Knightly, Prof. Dr. Udo Krieger, Prof. Dr. Michel Mandjes, Prof. Dr. Michal Pióro, Dr. Oliver Pfaffenzeller, and Dr. Jim Roberts.

At the end, I am grateful to my fiance Heike Schebler for her understanding. She often had to share me with science during leisure time and to renounce on my presence during many extended conference stays. My final thank-you is directed to my parents. They gave me the possibility to grow according to my talents and preferences. They taught me the right ideals for life and supported me in all my strivings.

Contents

1 Introduction

In today's telecommunication scenery there is a coexistence of circuit-switched telephone networks and packet-switched data networks. Since maintenance and operation of both network infrastructures are expensive, there is a clear trend for their convergence, which leads to next generation networks (NGNs). These networks should be a low-cost packet-switched solution with real-time transport capabilities for telephony and multimedia applications. In addition, NGNs should be fault-tolerant to support business-critical processes.

The base technology for NGNs will be the Internet Protocol (IP) due to its success and vast deployment in the last two decades. This protocol is simple to use, most of the potential end devices implement it, and IP networks are easy to maintain. However, IP technology lacks real-time communication properties like Quality of Service (QoS) guarantees in terms of packet loss and delay. Moreover, conventional IP networks have only limited fault-tolerance, which is based on signaling and the recalculation of routing tables.

There are two different basic approaches to enhance today's IP technology towards NGNs. QoS may be achieved by capacity overprovisioning, i.e., the network is provided with so much bandwidth that network congestion hardly occurs. But there is no method to determine the appropriate amount of bandwidth. Furthermore, it increases the capital expenses (CAPEX) in terms of capacity costs by a so far unknown multiple. From an operational expenses (OPEX) point of view, overprovisioning is an appealing option because it keeps human assisted

operation costs low. No new hardware and software features are required, primitive billing systems are sufficient, and only little cooperation among network entities is necessary. The other option is network supported QoS for which admission control (AC) is the key feature. The network admits real-time flows only if enough resources are available such that packet loss and delay requirements can be met. Otherwise, a request is blocked which is the equivalent to a busy tone from a telephone switching center under heavy load. On the one hand, the OPEX increase because AC makes router operation more complex, it requires interoperability among different Internet service providers (ISPs) to achieve end-to-end (e2e) QoS. Therefore, it needs more human interaction and control than capacity overprovisioning. On the other hand, AC limits the CAPEX to a modest amount because it turns potential QoS violations due to capacity shortage into call blocking and, which is most important, it serves as an insurance against unexpected overload due to new applications, BGP route changes, or link and router failures.

1.1 Contribution

This work focuses on control mechanisms for NGNs, in particular on AC and fault tolerance. We give a short introduction to IP and MPLS technology, and discuss the state of the art concerning QoS issues. The contribution of this work starts with an overview of today's AC approaches and we make a distinction between *link* and *network* AC (LAC, NAC). LAC has been well researched in the context of Asynchronous Transfer Mode (ATM) networks in the 1990ies [1] and it gives answer to the question: How much traffic can be supported by a single link? In contrast, NAC limits the network-wide traffic volume which is by nature a distributed problem. We propose a basic categorization of NAC methods from a resource allocation point of view and suggest a framework for performance evaluation regarding the resource efficiency, i.e., the average utilization of the required bandwidth is the performance criterion. This has impact on the CAPEX and is of major interest for ISPs who decide to solve the QoS problem by NAC.

We compare fundamental NAC methods by numerical results in different networking scenarios and analyze their performance behavior.

In NGNs, QoS should not be compromised by local outages, i.e., the network must be resilient to link and node failures. Resilience may be achieved by hardware redundancy on the physical layer but this is expensive because mostly 100% or more backup capacity is needed. Rerouting on the network layer is a cost-attractive alternative since less backup capacity is necessary to achieve the same result [2]. As overload is most likely to occur due to partial network failures [3], NAC must be resilient in these cases to maintain QoS. This is done by reserving enough enough backup capacity to carry flows that are rerouted onto a backup path due to a link or node failure. As a consequence, resilience requirements influence the resource efficiency of all NAC methods, which is also investigated in this work. Some NAC concepts require reservation states at intermediate routers of a reserved path. These are problematic if the path of a flow is relocated due to a network failure. They must be restored on the deviation path in real-time to make the outage invisible to the end user. This is very complex because, potentially, the states cannot be accessed any more. Therefore, a truly stateless core network eases the implementation of resilient NAC.

For the practical application of budget-based NAC (BNAC), the physical network capacity must be mapped to virtual capacity budgets such that no unintended overbooking can occur. We propose and compare several options for that kind of capacity assignment: on a single link, in a network, and in a network with resilience requirements. Our final result is an accelerated, efficient, and fair assignment algorithm for NAC budgets that takes resilience requirements into account.

Our experiments regarding resilience requirements show that routing and rerouting have a tremendous impact on the resource efficiency. Therefore, we want to take advantage of this potential by routing optimization. We consider protection switching methods, i.e., backup paths are established during connection setup and in case that the primary path fails, the traffic is switched to the backup paths. Multiprotocol Label Switching (MPLS) is suitable for the implementation

of such mechanisms since it provides virtual connections and route pinning over packet-switched communication protocols like IP. We suggest several protection switching mechanisms, in particular the Self-Protecting Multi-Path (SPM). They are based on multi-path routing that offers degrees of freedom for the optimization of the load balancing to minimize the required backup capacity. They are simple to implement because they do not need signaling in failure cases. The optimized SPM requires only 17% additional capacity to protect the network against all link and node failures while the conventional Shortest Path (SP) IP routing (e.g. OSPF or IS-IS) needs 80% more resources. Hence, our proposed routing optimization saves more than one third of the CAPEX in networks with resilience requirements. Note that these concepts can be well combined with both capacity overprovisioning and resilient NAC.

1.2 Outline

This work is structured as follows. Chapter 2 gives an introduction to the Internet Protocol (IP) and Multiprotocol Label Switching (MPLS) technology, the structure of the Internet, and discusses Quality of Service (QoS) issues. Chapter 3 gives an overview of basic admission control (AC) mechanisms by proposing a classification. Then, the focus is on budget-based network admission control (BNAC), which is subdivided into four categories. They are explained and illustrated by examples. We describe our performance evaluation methodology in detail and compare the performance of the basic BNAC types in various networking scenarios. We adapt our evaluation framework towards resilience requirements and present the performance results under this new aspect. Finally, we propose and compare algorithms to configure resilient network AC (NAC) in operational networks. Chapter 4 gives an introduction to routing issues with regard to fault tolerance and summarizes relevant results from the literature. The conclusion of this overview leads to simple protection switching structures with load balancing capabilities that we optimize with a linear program (LP) formulation. We com-

pare the required backup capacity for several new protection switching mechanisms and under various side conditions. As the SPM is the most promising protection switching approach, we consider its performance relative to shortest path rerouting in various existing networks. We present a comparative study with respect to different network characteristics using randomly constructed networks. Chapter 5 summarizes this work.

2 Basic Technologies for an NGN Architecture

In this chapter, we give an introduction to the Internet Protocol (IP) and describe it within a larger context because it will probably be the fundamental base technology for next generation networks (NGNs). We explain Multiprotocol Label Switching (MPLS) since it allows for an implementation of our protection switching mechanisms, that we propose in Chapter 4. Finally, we discuss Quality of Service and reliability issues because they are missing in today's technologies and have to be added for NGNs.

2.1 Internet Protocol Technology

The Internet Protocol (IP) has evolved in the past 35 years to the most important communication technology worldwide and, therefore, it will be the fundamental base technology for NGN solutions. After a short introduction to communication protocols and the concept of protocol layering, we present IP in detail. To complete the picture, we give some examples for higher layer protocols that enable end systems to communicate seamlessly with each other. We illustrate the structure of today's Internet, explain the addressing scheme of IP, and show how IP datagrams are forwarded according to routing tables. These routing tables are

composed automatically by routing protocols that are an essential element of IP technology.

2.1.1 Communication Protocols

Communication protocols are necessary to enable communication of remote systems. They exchange messages that have to be interpreted in the same way by all participants. If the remote systems are heterogenous and stem from different vendors, these protocol specifications must be publicly available. In the IP context, they are standardized by the Internet Engineering Task Force (IETF) [4] and these standards are called “Request for Comments” (RFCs).

We consider web surfing to explain the principle of the protocol stack and to illustrate the use of protocols in a top-down fashion. When a user clicks on a hyperlink containing a uniform resource locator (URL), e.g. “http://www.menth.net/index.html”, the web browser generates a request message to the computer with the name “www.menth.net” to get the file “index.html”. When the message is received by the remote computer, a web server program processes this message and sends the desired content back to the web browser, which eventually renders the content on the screen. The commands in such messages and the required actions, e.g. sending the requested file, are defined by the Hypertext Transfer Protocol (HTTP) [5] such that web browser and the web server can communicate. The pattern that a client program, i.e. the web browser, contacts a server program at a well-known location location is called client–server communication, which is a well known principle.

When two processes on remote machines, e.g. client and server programs, exchange messages, the packets must be addressed with a port number of the sender and receiver process that the destination machine can deliver the data to the correct process and to reveal its origin. In the above example, the HTTP request message is equipped with both port numbers in a format which is standardized by the TCP protocol that we explain later in this section. Since HTTP relates to the application and TCP to the transport of general messages, the first one is

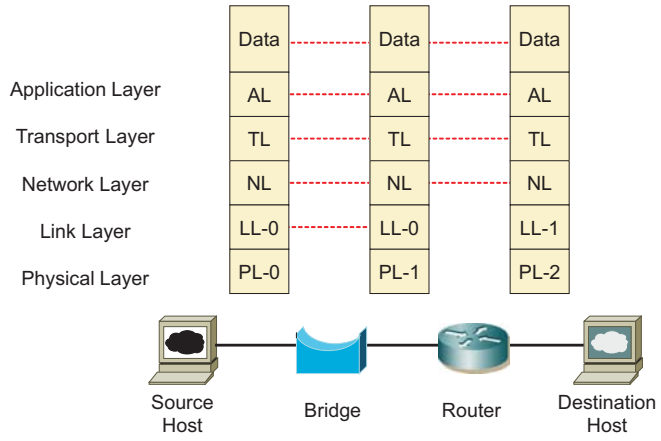


Figure 2.1: *Representation of a data packet on different links along the way.*

called an application layer (AL) protocol and the second one is called a transport layer (TL) protocol. The TCP-related data is called a protocol header that is attached to the HTTP message which is in this case the protocol payload of TCP.

The packet containing the HTTP and TCP information must be conveyed to the destination computer, possibly over several intermediate hops. The Internet Protocol (IP), a network layer (NL) protocol, standardizes the addressing of the destination machine and some other aspects. The IP header is prepended to the TCP header such that the resulting IP packet, also called IP datagram, contains HTTP/TCP/IP information. The consecutive application of various protocols is called protocol layering or stacking.

The logical link control (LLC) takes care that IP datagrams are translated into a series of zeros and ones such that its start and end can be recognized from a bit stream of consecutive packets. In addition, they add a checksum to the series of bits to verify whether the information has been transmitted correctly. The

Point-to-Point Protocol (PPP) [6, 7, 8] performs that task on a point-to-point link. Another widely used protocol of a broader scope is the High-Level Data Link Control Protocol (HDLC). The media access control (MAC) regulates usage of the physical medium by the machines which is challenging if several computers use a shared medium. For example, the Ethernet protocol controls how a common bus is accessed by several stations and adds the hardware addresses to the packets. The LLC and MAC are constitute the link layer (LL).

The physical layer (PL) transforms the bits into physical signals that can be interpreted by the next station that receives the message. The protocol stack in Figure 2.1 applies to a typical Internet scenario and deviates from the original and rather academic Open System Interconnection model defined by the International Standardization Organization (OSI/ISO). The packet size grows if headers are consecutively stacked. As packets are passed on from the source computer over several intermediate station to the destination machine, the information related to the application, transport, and network Layer protocols remains unchanged, while all information of the link layer and below is renewed if network borders are crossed and the physical layer is changed by any node on the way.

2.1.2 The IP Protocol

First, we motivate the need for a network layer abstraction like IP to enable transparent communication across network boundaries. Then we explain details of the IP protocol.

Inter-Networking

There are many types of physical media for data transportation that require hardware-specific protocols for operation. Link layer protocols are also hardware and vendor-specific. They are deployed in different networks but they are not necessarily compatible. Hence, communication based on the link layer is only possible within a single homogeneous network infrastructure. However, data ex-

change among multiple and heterogeneous networks is a prerequisite for global communication and for applications like email or the worldwide web.

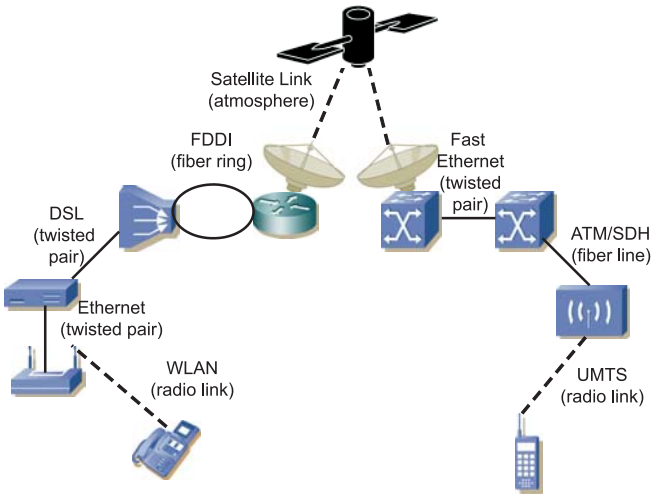


Figure 2.2: *The Internet Protocol is a network layer protocol and provides a uniform addressing scheme for heterogeneous networks.*

Figure 2.2 shows the data path from a residential user calling someone on a wireless phone with an IP application. The data packets are transported from a wireless phone via a Wireless Local Access Network (WLAN, IEEE 802.11) to the WLAN access point. This is connected by an Ethernet protocol over a twisted pair wire to the Digital Subscriber Line (DSL) modem that is connected by the telephone line (twisted pair) to the DSL access router in the switching center of the Internet service provider (ISP) using the Point-to-Point Protocol (PPP) [9]. The data are forwarded over an optical fiber according to the Fiber Distributed Data Interface (FDDI) network [10, 11] to a satellite link that is operated again by

PPP. The next network is based on Fast Ethernet that runs over coaxial cable to an ATM network which is based on the Synchronous Digital Hierarchy (SDH) to manage the underlying optical network. It interconnects the base station, called NodeB, in the Universal Mobile Telecommunication System (UMTS) Terrestrial Radio Access Network (UTRAN) which transmits the data to the mobile phone over the air interface according to UMTS specifications.

This scenario motivates the need for a network layer abstraction with a unifying addressing scheme to transport higher layer data transparently over heterogeneous networks. The Internet Protocol (IP) provides this functionality.

The IP Header

Figure 2.3 shows the layout of the IP header [12]. The current version of IP is indicated by the first 4 bits, and the next 4 bits reveal the header length in 32 bit words. The length of the header is variable due to optional fields and the header is padded with zeros to full 32 bit words. The type of service field (TOS, 8 bits) can be used to indicate a priority class. The next 16 bits show the length of the whole datagram including the header in bytes. The 16 bit identifier field is required when a packet is fragmented into several smaller pieces due to a Maximum Transfer Unit (MTU) of a link on its way from its source towards its destination. Then, the pieces have the same identifier and the 13 bits offset indicates the amount of payload in units of 8 bytes that have already been sent by earlier fragments. The 3 bits flag controls the fragmentation process. The Time-to-Live (TTL) number is set to an initial value that is decremented by 1 in each router. If the TTL reaches zero, the packet is discarded. In such a case, the sender is notified by an Internet Control Message Protocol (ICMP) packet. This is useful for analysis purpuses, e.g., the “traceroute” tool for network analysis takes advantage of that mechanism to discover the routers of the path to a specific destination. The protocol number identifies the protocol type in the payload, e.g., number 6 stands for TCP and number 17 stands for UDP, which are both explained in the next section. The checksum protects the complete IP header and helps to validate its integrity. If

the evaluation of the checksum indicates an error at the destination, the packet is discarded. The fourth and the fifth word carry the source and the destination address of the packet. Finally, IP options can be added. For example, source routing can be implemented, i.e., a list of routers can be given in the options field that have to be visited on the way to the destination. However, options are rarely used as they slow down the forwarding process considerably because routers are optimized for the standard IP header processing in the so-called fast path.

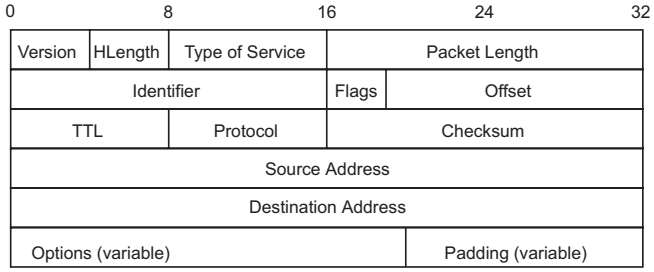


Figure 2.3: *Format of the IP header.*

Currently, version 4 of IP (IPv4) is in use. The new IP version 6 (IPv6) has been standardized for years and it is expected to replace IPv4. The most important change of IPv6 is the extension of the address space from 4 to 16 octets (bytes) because as more and more devices need to be addressable, IPv4 addresses will run short. Network Address Translation (NAT) mitigates this phenomenon but more addresses are definitely required. However, the transition from IPv4 to IPv6 in the Internet reveals to be a tedious process.

2.1.3 Higher Layer Protocols

So far, we have illustrated how worldwide connectivity is achieved on the basis of IP datagrams. In this section, we consider the transport and the application layer

abstraction on top of the IP network layer. We give some examples that play a role in the context of real-time communication.

Transport Layer Protocols

Transport layer protocols organize the multiplexing of data streams from different applications into an IP packet stream and enable a remote machine to assign the received data to the corresponding processes. The transport layer is the lowest abstraction that application programmers are faced with. From a programming technical point of view, data are transmitted through so-called sockets that allow a program to exchange messages with another process on a distant computer. A socket is identified by a source and destination IP address, which is carried by the network layer, and by a source and destination port number of two bytes each, that are carried by the transport layer. A port is like mailbox for a process within a computer such that packets that are addressed to a certain port can be delivered to the correct process. Server programs have well-known ports that are automatically used by software of client applications to contact the respective server programs. For example, web servers listen usually on port 80.

When service differentiation of different flows is required, e.g. for prioritization or policing purposes (cf. Section 2.3), the packets of a single flow must be recognized. A flow is defined by flow descriptor which usually consists of the source and destination IP addresses and source and destination port numbers. Therefore, the respective information in the network and transport layer is inspected to check whether a packet belongs to a specific flow. However, if the IP payload is encrypted for security reasons, the port numbers in the transport layer can no longer be inferred by intermediate routers. This problem is solved by the flow label in IPv6. As transport layer protocols fulfill also other essential but protocol-specific tasks, we explain briefly UDP and TCP, which are the most frequently used transport layer protocols in the Internet [13].

Transmission Control Protocol The Transmission Control Protocol (TCP) [14] provides reliable transmission between two end systems. All transmitted data segments have to be acknowledged to assure the complete and in-order delivery of the data. The actions required by the TCP protocol are described by state machines and require a session context, i.e. session-specific information like the number of the last unacknowledged transmitted segment. Therefore, TCP is a connection-oriented protocol which is more complex than the connectionless UDP counterpart.

TCP also performs flow control based on a window mechanism, i.e., sender and receiver agree upon a certain receiver buffer size that limits the amount of data that the sending process may transmit without having received acknowledgements for all previous data segments. Congestion control is performed by the TCP sender when packet loss is detected through missing acknowledgements which is in wireline networks mostly due to congestion. In this case, the sender decreases its sending window size drastically which reduces the amount of unacknowledged data in the network and throttles its transmission rate. To recover afterwards from an overreaction, the sending window size is increased again. Hence, TCP is not suitable for real-time communication with delay constraints as its sending rate is rather controlled by the network state than by the application.

User Datagram Protocol The User Datagram Protocol (UDP) [15] is very simple and does not provide reliable transmission. Its header is 8 bytes long and contains the source and destination port, two bytes for the length of the payload, and a checksum (also two bytes) to enable end systems to detect bit errors in the UDP header. No flow and congestion control is applied. Therefore, UDP is used for real-time applications whose traffic is not intended to be slowed down by occasional packet losses.

Other Protocols on Top of Plain IP For some purposes the addressing of a specific port is not mandatory, e.g., if the network nodes communicate with each other independently of any application. The Internet Control Message

Protocol (ICMP) [16, 17] is used by hosts, routers, and gateways to communicate network layer-specific information to each other, e.g., notifications about expired TTLs are sent over ICMP. Another example for direct message transport over IP is the Resource Reservation Protocol (RSVP) [18].

Application Layer Protocols

Standardized Application Layer protocols are required to enable different applications of different vendors to interoperate. Usually, they use the Transport Layer capabilities of TCP or UDP. As outlined above, TCP is not suited for mission-critical real-time communication but UDP does not provide reliable data transfer nor in-order delivery. This and other functionality is added by real-time transport protocols RTP and RTCP. A remote control for streaming purposes is realized by RTSP. In contrast, SIP and H.323 help to initiate a session with a person or entity whose IP address and communication abilities are currently unknown.

Protocols for Real-Time Transport The Real-Time Transport Protocol (RTP) assigns synchronization source identifiers to different media streams [19, 20]. This allows a sender to multiplex several streams, e.g. voice and video, of a single application into a single packet flow and it allows receivers to identify multimedia streams from different senders, e.g. in case of a video conference. Some additional information is provided to synchronize the payload of the RTP packets. Every RTP header carries a payload type number that identifies the format of the carried stream. The sequence number addresses the deficiency of UDP to deliver packets in-order. The timestamp of the most recent sample in the payload is given related to the sampling rate of the respective encoder but not to the wall clock time.

The accompanying RTP Control Protocol (RTCP) is also standardized in [19, 20]. It sends messages periodically to map the timestamps of different streams to wall clock time in such a way that a lip synchronized playout of voice and video can be achieved by applications. In addition, RTCP is used to provide sender

reports to identify the sender and its streams, and receiver reports to give feedback on the received transmission quality. The session control in the application can take advantage of this information to adapt to good or bad channel conditions. The frequency of the reports depends both on the rates of the streams and on the number of participants in a session because only a small fraction of the bandwidth should be consumed for control purposes.

Protocols for Media Streaming Streaming voice or video is non-interactive real-time communication, i.e., the communication is unidirectional. As long as no live interaction of the audience is required, a transmission delay in the order of seconds is acceptable for live transmissions, e.g., if the program of a local radio station is offered over the Internet. Stored video is another example for video streaming if the playback starts as soon as enough frames are buffered. In addition, the consumer wants to have a VCR-like control of the media [21], i.e., he wants to fast forward or backward and jump to some bookmarks. The Real-Time Streaming Protocol (RTSP) [22] standardizes this kind of control between client and server.

Protocols for Real-Time Communication Setup and Control A challenge for ubiquitous communication is contacting the callee if the IP address of his currently used communication device is unknown. The Session Initiation Protocol (SIP) [23] solves this issue by a registrar, the so-called SIP proxy. If a caller wants to initiate a session, he might contact the callee, e.g. Bob, directly by sending an INVITE message to bob@193.60.210.89, i.e. to his computer, or via a SIP proxy to sip:bob@domain.org. If Bob is not at his usual working desk but reachable at another IP address or telephone number, or if he has received his IP address by the Dynamic Host Configuration Protocol (DHCP), he might have registered that address at the proxy before so that the invitation is redirected to the correct device. Then, a media encoding format is negotiated that both the caller's and the callee's device can handle. In addition, it provides mechanisms for call management, e.g., participants can be invited during a session, the media

encoding format can be changed, and new streams can be added.

The H.323 protocol is standardized by the International Telecommunication Union (ITU) and achieves the same objectives as SIP. The equivalent to the SIP proxy is called gatekeeper. The H.323 protocol suite is an umbrella standard that is more specific about other protocols, e.g., it mandates RTP as transport protocol for media streams and each terminal must support G.711 encoded speech. It even describes how Internet phones have to interoperate through gateways with the public circuit-switched telephone network.

Examples of Other Application Layer Protocols There are many other widely used application layer protocols, e.g., the Hypertext Transfer Protocol (HTTP) [5] which is the foundation of the worldwide web, the Simple Mail Transfer Protocol (SMTP) [24] which standardizes email exchange, or the File Transfer Protocol (FTP) [25] which is used for file downloads. The Domain Name System (DNS) [26, 27] maps domain names of computers like "www3.informatik.uni-wuerzburg.de" to their corresponding IP numbers and is, therefore, used for almost any communication setup. We do not explain them in this work because they are not real-time communication-specific. They will be used in an NGN environment in the same way as in the traditional Internet since NGNs must be downward compatible to the current technology.

2.1.4 The Structure of the Internet

The Internet consists of many interconnected independent administrative units, so-called autonomous systems (AS). It is organized in a pseudo-hierarchical structure as illustrated in Figure 2.4 [28] and whose levels are called tiers.

The Hierarchical Structure The networks of the tier-1 Internet service providers (ISPs) constitute the backbone of the Internet. They are fully connected among each other and have international coverage. Tier-2 ISPs have regional or national coverage. To reach a large portion of the entire Internet, they are con-

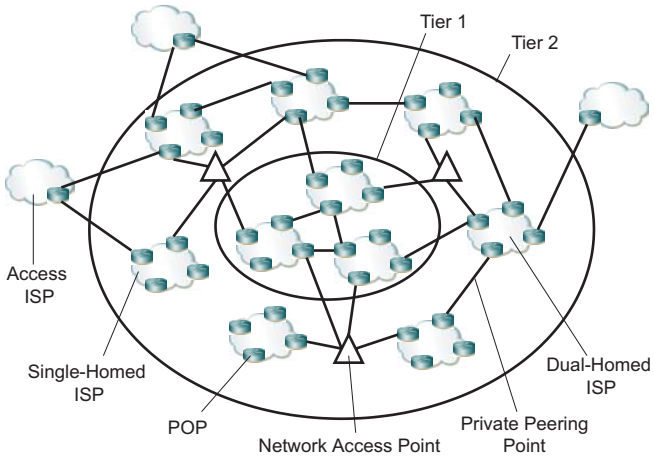


Figure 2.4: *The pseudo-hierarchical interconnection of ISPs.*

nected to one or several tier-1 ISPs. Below the tier-2 ISPs are the lower-tier ISPs, which connect to the Internet via one or more tier-2 ISPs. At the bottom of the hierarchy are the access ISPs, selling Internet access directly to end users and content providers. If they are connected to only one higher-tier ISP, they are called stub-ASs. Access networks mostly reveal a strongly hierarchical topology while networks of higher-tier ISPs have a more regular structure [29, 30, 31]. ASs that transport traffic originated from or destined for other ASs are called transit networks. They are assigned a unique 16 bit AS number (ASN) by the “Internet Corporation for Assigned Names and Numbers” (ICANN) [32] for routing purposes.

A provider ISP charges a customer ISP a fee, which typically depends on the bandwidth of the link connecting the two. To save costs, tier-2 ISPs may also choose to connect directly to each other, which is called peering. Some tier-1

ISPs also act as tier-2 or lower-tier ISP and sell Internet access directly to large companies or institutions. Increasingly, lower-tier ISPs are connected to several higher-tier ISPs to remain still connected to the Internet if one upward exit of the network fails. An ISP is called single-, dual-, or multi-homed depending on its number of provider ISPs. In conclusion, the hierarchical structure is not entirely strict.

POPs and Direct Peering Points A Point of Presence (POP) is simply a group of one or more routers in an ISP's network to which routers of other ISPs can connect, no matter whether they are at the same level in the hierarchy, below or above. To connect to a provider's POP, the customer ISP typically leases a high-speed link from a third-party telecommunications provider and directly connects one of its routers to a router at the provider's POP. A tier-1 provider typically has many POPs scattered across different geographical locations in its network, and multiple customer ISPs connect into each of these POPs. Two tier-1 ISPs may also peer with each other at several pairs of POPs.

Network Access Points In addition to such private peering points, ISPs often interconnect at Network Access Points (NAPs), also called Internet Exchange Points (IXPs) , that are owned and operated by either some third-party telecommunications company or by an Internet backbone provider. Because the NAPs relay and switch tremendous volumes of traffic, they often consist of complex high-speed ATM switching networks, concentrated in a single building. The trend is for the tier-1 ISPs to interconnect with each other directly at private peering points, and for tier-2 ISPs to interconnect with other tier-2 ISPs and with tier-1 ISPs at NAPs [33].

2.1.5 Addressing and Forwarding

First, we give insights into the structure of IP addresses and then we explain the forwarding of IP datagrams by routers which depends fundamentally on the

addressing scheme.

IP Addressing

The boundary between a host or a router and a physical communication link is called an interface. By nature, routers have several of them. IP addresses are assigned to interfaces rather than to machines, therefore, hosts mostly have one IP address whereas routers have several IP addresses. These numbers are 4 octets in length, i.e. 32 bits, and are often denoted in dotted-decimal notation, e.g. 132.187.105.113, in which each byte of the address is written in its decimal form and is separated by a period from other bytes in the address. The n leftmost bits in the IP address are called the network prefix or network mask which is denoted by $a.b.c.d/n$. The rightmost part signifies the interfaces within the corresponding network. Originally, n was restricted to $n = \{8, 16, 24\}$ for class A (/8), class B (/16), class C (/24), and class D (/24) addresses. Class A addresses can be parameterized by the network prefix 0/1, class B by 128/2, class C by 192/3, and class D by 224/4. Class D addresses are reserved for multicast purposes. Class C addresses can cover only 254 computers within a network because host number 0 is invalid by definition and host number 255 is used for broadcast purposes. In contrast, class A addresses are very valuable due to their large address space but there are only 126 of them as 0/8 and 127/8 are reserved values. Since this classful allocation of IP addresses leads to an unnecessary limitation of network prefixes and network sizes, the Classless Interdomain Routing (CIDR) [34] since 1993 allows the prefix size n to take conceptually any value between 1 and 32. This assignment rule holds for ASs. A further subdivision of networks into smaller units within such an authority is called subnetting [35].

Datagram Forwarding

A routing table specifies exactly to which outgoing interface a router has to forward an IP datagram. We discuss this by a modified example taken from [28], presented in Table 2.1.

Table 2.1: *An example routing table.*

Destination	Interface
127/24	127.0.0.1
192.168.2/8	192.168.2.5
192.168.2.96/6	192.168.2.96
192.55.114/8	193.55.114.6
193.168.3/8	192.168.3.5
224/24	193.55.114.6
0/32	193.55.114.129

The routing table consists of pairs of network prefixes and corresponding outgoing interfaces. Routing is the possibly distributed calculation of the routing table and the determination of the correct outgoing interface for an IP datagram according to the routing table. Hence, the longest match between the destination network mask in the routing table and the destination address determines the outgoing interface. For example, any IP address matching 192.168.2.96/6 also matches 192.168.2/8 but due to this rule, datagrams are forwarded on interface 192.168.2.96 instead of 192.168.2.5. All addresses that do no match any special network prefix are destined to the default destination (0/32) and forwarded on the corresponding interface. The 127.0.0.1 entry is the so-called loop-back interface which returns IP packets back to the machine itself. This mechanism is used for debugging purposes. The last entry is also special as it regards multicast addresses.

The network prefixes a.b.0/17 and a.b.128/17 can be aggregated to a new network prefix a.b/16, which is called route aggregation or summarization. Routing tables are small if the routing of the entire address space can be represented in a very compact way. Thus, the traffic leaving a common interface should be parameterizable by a few network prefixes. Hence, route aggregation makes IP forwarding quite scalable provided that the IP addresses are assigned at least in a

pseudo-hierarchical manner. IP addresses are assigned blockwise by the ICANN and they can also be obtained from an ISP which results in a hierarchical structure in the sense that all IP addresses beginning with the prefix of an ISP's network mask can be reached through its network. Exceptions can be handled by the longest match first rule in the routing tables.

2.1.6 Routing Protocols

The IP packets are forwarded according to routing tables that are configured in each router. This is mostly done automatically by routing protocols [36]. They determine for each router the next hop on the way for every destination in the Internet by exchanging reachability or topological information.

As ISPs are in general not willing to disclose information about their network to competitors and as the entire Internet is too large for the exchange of detailed routing information, the routing in the entire Internet is done in a hierarchical fashion that reflects the structure of the Internet.

Each AS represents an autonomous routing domain where the routing of local addresses can be done independently of other AS. This is called intra-AS routing which is performed by intra-AS or interior gateway routing protocols (IGPs). A gateway is a router that enables packets to cross an AS boundary, examples are peering routers or routers in an NAP. If an IP packet is addressed to a foreign AS, it needs to cross a number of ASs. This path is determined by inter-AS or exterior gateway protocols (EGPs).

Intra-AS Routing

Intra-AS or intra-domain routing protocols can be classified into distance vector protocols and link state protocols. Interfaces are associated with link costs that are additive with respect to a path. The cost metrics may be hop count, delay, utilization, or others. Both protocol types determine a lowest-cost path for a route to avoid loops. We explain these concepts and discuss the Routing Information

Protocol (RIP) and the Open Shortest Path First (OSPF) routing protocol as examples.

Distance Vector Protocols The distance vector protocol approach requires each router to maintain a distance table that holds the next hop router and the associated costs for each destination within the routing domain. Initially, the table holds only the router itself and its directly attached neighbor as destination with a path cost of zero or the respective interface costs. A vector of the reachable destinations together with the path costs is transmitted periodically to all neighboring routers. If a router A receives such a distance vector from a router B, it adds the costs to reach B to the respective path costs and compares them to the entries in its own distance table. If no entry for a destination exists or if the new cost to a destination is smaller than in the distance table, then the respective next hop router in the table is replaced by B and the new cost. The algorithm eventually converges. If an interface is no longer active, its cost is set to a large value such that unreachability can be assumed if the cost of a path is larger than this value. The propagation of this information is very slow, so transient loops are created for a while, which can be solved by the "poisoned reverse" approach [28].

The Routing Information Protocol (RIP) version 2 [37] exchanges RIP advertisement every 30 seconds over UDP. If a router does not get an update from its neighbor once within 180 seconds, it assumes that this neighbor is no longer reachable. In the first version of RIP the hop count was the mandatory link metric, i.e., the link costs were all one and the maximum cost of a path was restricted to 15. Hence, a maximum network diameter of 15 hops was a prerequisite for the application of that protocol.

Link State Protocols Link state protocols require routers to broadcast the identities and costs of their attached interfaces to all other routers in the network. Hence, routers can infer the complete network topology including costs by evaluating the link state messages, the so-called link state packages (LSPs). Based

on this complete view, each router can locally compute a minimum cost path to every destination in the network by Dijkstra's shortest path algorithm [38]. The routing table is composed according to this result.

The Open Shortest Path First (OSPF) protocol version 2 [39] broadcasts the link state advertisements every 30 seconds or if a topology change has been recognized. As the messages are sent directly on top of the IP protocol, OSPF takes also care of reliable message transfer. The OSPF protocol also checks whether links are operational via so-called "Hello" messages that are sent periodically to each attached neighbor, and allows an OSPF router to obtain a neighboring router's database of network-wide link states. All exchanges between OSPF routers are authenticated, i.e., only trusted routers can participate in the routing process, which prevents malicious intruders to inject wrong information. The Equal Cost Multi-Path (ECMP) option allows to create multiple paths to a destination provided they have the same costs. In addition, both unicast and multicast routing is supported.

The OSPF protocol allows to subdivide an AS into "areas" where the OSPF protocol is performed independently. Each area has at least one area border router that has a similar responsibility as the AS gateway routers. All area border routers constitute a backbone whose primary role is to route traffic among different areas in the AS. This mechanism supports the scalability of the routing by reducing the amount of exchanged link state advertisements.

The Intermediate System to Intermediate System Routing Exchange Protocol (IS-IS) is also a link state routing protocol and it is after OSPF the mostly utilized IGP in the Internet.

Inter-AS Routing

As mentioned above, gateway routers connect to neighboring ASs and are in charge of exchanging traffic destined for other ASs. Therefore, all of them must be reachable from a network, which requires that it is present in the gateway's routing table. Currently, there are about 16,000 ASs in the Internet [40]. There-

fore, the routing table for interdomain destinations can become very large. Hence, one goal of interdomain routing is route aggregation to keep the number of network prefixes low. Inter-AS paths are primarily chosen with respect to policy rules. For example, traffic is only forwarded to ISPs whose reachability information is trusted and that have enough capacity, or ISPs want to carry only the traffic from or to its customers. Therefore, the shortest path metric is not appropriate for interdomain routing purposes and not feasible either because the intradomain costs for the traversal of transit ASs are not comparable.

The de facto standard for inter-AS routing is the Border Gateway Protocol (BGP) version 4 [41, 42, 43]. Every AS has one BGP speaker that exchanges information about reachable networks with the BGP speakers of neighboring ASs over a reliable TCP connection. Thus, the abstract graph consists of nodes that represent networks and edges that result from provider-customer or peering relationships. If an AS has several BGP speakers, special care is required to maintain consistency. To support policy-based routing decisions, inter-AS routers announce for each possibly aggregated network address a list of attributes like interdomain routers and AS numbers on the respective path. Hence, BGP is a path vector protocol that works similarly to a distance vector protocol. In contrast to the presented intradomain protocols, the information is not sent periodically and only updates like route changes or route withdrawals are propagated. If a route fails, it may take tens of minutes until the convergence of the protocol reaches a consistent view of all routing tables [44, 45]. Therefore, outages of transit ASs should be avoided to prevent such scenarios.

BGP provides only the mechanism to exchange the relevant information but it does not determine the routing policy for which we give some examples. To avoid loops, a router must not forward traffic to a neighbor that announces a route containing its own ASN. If more than one path is acceptable, the one with the smallest number of ASs is usually taken by an AS and further announced to its customers. A multi-homed stub-AS announces to its providers that it has no path outside its domain to prevent being misused as a transit AS. Most of the routing policies are unknown because the contracts between peering ASs are

mostly confidential.

Inter-AS routing poses two protocol challenges. It requires firstly that BGP speakers of neighboring AS exchange reachability information and secondly that this information is distributed among all intradomain routers. The first task is performed by the Exterior BGP (E-BGP). If the amount of inter-AS routing information is small, e.g. in stub-ASs, the second duty can be supported by the normal intradomain routing protocol. However, the amount of inter-AS routing information in backbone networks is large, therefore, it cannot be distributed at regular intervals. The Interior BGP (I-BGP) helps to distribute the reachability information from the BGP speakers to the internal routers. The closest border router from each internal router towards a certain destination can be figured out by means of the intradomain routing protocol.

2.2 Multiprotocol Label Switching

MPLS stands for “Multiprotocol” Label Switching. Multiprotocol because its techniques are applicable to any network layer protocol [46]. The mechanism resides between the link layer and the network layer. A connection, a so-called label-switched path (LSP, not to confuse with link state packages of link state routing protocols) between two distant computers is set up and the packets are forwarded in the routers based on label switching instead of packet routing, which simplifies the forwarding process. The participating routers are called label switching routers (LSRs). Figure 2.5 illustrates that the LSP ingress LSR equips an IP packet with a label of 4 bytes – a so-called shim header – and sends it to the next LSR. The intermediate LSRs classify a packet according to its incoming interface and label. Based on this information and the incoming label map (ILM), label swapping is performed and the packet is forwarded to the respective outgoing interface. The LSP egress LSR just removes the label from the IP packet header and routes it to the next hop. In practice, modern routers are capable of processing both IP and MPLS packets. Hence, the label swapping process

requires entries for every LSP in the management information base (MIB) of an LSR, so MPLS is a stateful technology.

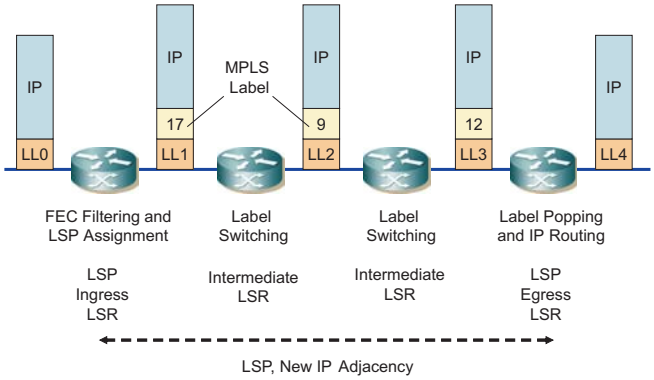


Figure 2.5: *An LSP creates a new IP adjacency.*

There are two major protocol alternatives for establishing an LSP. The RSVP Tunneling Extensions (RSVP-TE) [47] are modifications to RSVP that enable it to distribute labels. The Constraint-Based Label Distribution Protocol (CR-LDP) [48] has been designed particularly for that goal but the IETF now seems to adhere to RSVP-TE. An LSP may be established and associated with bandwidth reservations, e.g., using the primitives of RSVP. Thus, the LSP represents a virtual link that borrows its resources from the links connecting its LSRs. The more general Label Distribution Protocol (LDP) is not able to make reservations [49].

The label distribution and the label switching paradigm allow for explicit route pinning which gives a finer control on packet forwarding than routing. This is especially useful for traffic engineering [50, 51, 52, 53, 54]. As MPLS implements the connection concept, it is often viewed as a modified version of the Asynchronous Transfer Mode (ATM) [55, 56] with variable cell size. But there is a profound difference: ATM enables a two-fold aggregation with its virtual con-

nection and virtual path concept while MPLS allows for many-fold aggregation using multiple label stacking [51], i.e., an LSP may be transported over other LSPs. This feature helps to build scalable network structures, so-called LSP hierarchies [57, 58, 59, 60, 61].

2.3 QoS Issues

Due to economical aspects, a convergence of conventional communication systems such as telephony and Asynchronous Transfer Mode (ATM) networks, and data networks such as the Internet into an NGN architecture is desired. Traditional telecommunication networks have two revenue generating properties:

- They offer Quality of Service (QoS) in terms of limited packet loss and delay. Jitter, i.e. the delay variation among the packets of a flow, is kept to a minimum. These so-called premium services support interactive real-time communication such as telephony or remote control. They are especially required for demanding multimedia applications such as video conference [11] or mission-critical telematic applications. Note that our definition of QoS is only the technical-transmission-related subset of the definition given by the International Telecommunication Union (ITU) [62, 63].

- They provide high reliability which is required for business-critical applications such as Virtual Private Networks (VPNs) or carrier grade networks. Business customers require 99.999% service availability and are not ready to hazard the consequences of a local network outage. Reliability is also a component of the QoS definition of the ITU.

A router forwards the packets received from its input interfaces to its output interfaces. In between, the packets are switched from the input ports to the respective output ports where they are queued until they can be sent through the output interface. This is depicted in Figure 2.6. As the queues have limited capacity, they can overflow in which case packets are discarded. Thus, packet loss

occurs at the IP level. If the queues are very long, packet delay occurs with a duration that is also determined by the link bandwidth. Packet loss and delay can be avoided if routers have sufficient resources to carry the traffic or if the traffic rate is low enough for the respective router.

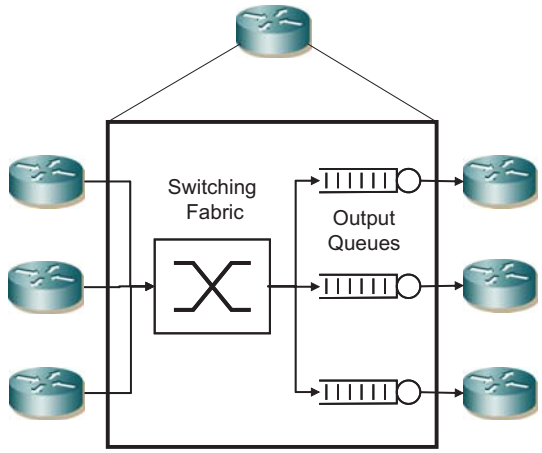


Figure 2.6: *The switching fabric and the output queues are essential components of a router.*

The availability of networks is compromised by internal outages. For example, links may fail due to physical damage or routers can fail due to software bugs, hardware crashes, or bad configuration. As a consequence, some network regions may be no longer reachable. When routing tables are constructed automatically by routing protocols, an alternative path can be found if such a path is feasible by the network topology. So far, this takes in the order of a minute to provide the deviation path if the timers of the routing protocols are set to default values.

Future networks will be packet-switched to support the successful connectionless IP communication model but they also have to provide QoS and high

reliability. Capacity overprovisioning, service differentiation, and admission control are approaches to introduce QoS in packet-switched networks. Network reliability is mainly achieved by massive redundancy in communication systems.

2.3.1 Overprovisioning

One technology-extensive solution to provide QoS is bandwidth overprovisioning [64, 65, 66], i.e., the network is equipped with sufficient bandwidth such that no congestion occurs.

Small networks have physically limited ingress lines which allows for the estimation of the maximum traffic rate on internal network links. In large networks, the traffic peaks on internal links are limited due to the high traffic aggregation and stochastic arguments. The traffic intensity can be measured and capacity provisioning can be based on forecasts. Such forecasts also have to take into account sudden load changes due to BGP updates. This and other unplanned events make overprovisioning a hard task. Since no modifications of today's IP world are required, this method is quite appealing. However, there is little evidence how much overprovisioning is required to have a sufficiently high probability that network congestion does not occur. Therefore, the resource efficiency of overprovisioning is still unknown which is a critical question for economical considerations.

2.3.2 Service Differentiation

Packet loss and delay can be avoided by capacity overprovisioning. However, accidental increases of the traffic rates lead to congestion in the routers if they exceed the estimates that were the base for capacity dimensioning because there is no technical possibility to at least improve the QoS for high-priority traffic. Different traffic classes are defined for service differentiation and high-priority packets are served preferentially to reduce their loss and delay in overload situations. For example, high-priority packets may overtake low-priority packets in the output queue of the router and low-priority packets are discarded with a

larger probability to leave the buffer space for high-priority packets. However, such mechanisms only mitigate the effects of congestion on high-priority traffic and cannot prevent that sufficiently large overload leads to QoS degradation.

In the following, we give an introduction to the Differentiated Services framework which implements preferential treatment on the packet level. Buffer management and packet scheduling disciplines in routers can balance the packet loss and delay among different traffic classes.

Differentiated Services

The Differentiated Services (DiffServ) framework [67] introduces different traffic classes [68, 69]. A corresponding Per-Hop Behavior (PHB) defines how packets of these classes are forwarded by the routers. Therefore, the terms traffic class and PHBs are equivalent in the DiffServ context. The DiffServ Code Point (DSCP) carries the PHB in the ToS field of the IP header [70] and the packets are labelled with the corresponding value either by the host or by an access router.

As outlined above, PHB mechanisms achieve service differentiation but they cannot avoid congestion in general. Therefore, traffic conditioners limit the rate of the traffic entering the network. A meter monitors the PHB-specific rate of the ingress traffic. Depending on the policy different actions may be performed:

- A marker marks the packets as in- or out-of-profile according to a traffic conditioning agreement (TCA) in the service level agreement (SLA). This is done on an aggregate basis, i.e., packets are treated unaware of the flows they belong to. One possibility is to discard packets that are marked out-of-profile.

- A second policy is downgrading the traffic to the normal best effort class.

- Another option is to carry the excess traffic according to its PHB type and to discard the marked packets only if overload occurs. This is called policing.

- The traffic conditioner may also work as a spacer, i.e., it may delay packets until they are in-profile according to the TCA. The packets are only discarded if the spacer buffer overflows. Spacers are often implemented in hardware [71].

The concept of service differentiation on the packet level scales well as the routers must be aware of only a few PHBs. The original stateless IP approach is marginally modified because the DSCP is recorded in the ToS field. However, service differentiation on the packet level impairs the QoS of all flows of a single PHB in the same way [72, 73, 74, 75]. For demanding applications it makes more sense to block some flows entirely in overload situations and to provide high QoS for the others. This mechanism is called admission control (AC) and will be the focus of Chapter 3.

Buffer Management and Packet Scheduling

The implementation of PHBs takes into account both buffer management and packet scheduling algorithms [76, 77].

Buffer management algorithms decide whether a router should store a received packet in its buffer if the forwarding unit is busy. Packets are usually discarded in case of buffer overflow. This simple buffer management policy is called Drop Tail. Random Early Detection (RED) gateways [78, 79, 80, 81, 82] discard packets based on a PHB-specific probability that depends on the buffer occupation.

Packet scheduling is an online algorithm that determines the order in which already queued packets are played out. The normal proceeding is First-In-First-Out (FIFO) scheduling which does not differentiate between traffic classes. Static Priority (SP) forwards packets of higher priority classes exhaustively in a FIFO manner and delays packets of lower priority classes until no high-priority packets are waiting. Generalized Processor Sharing (GPS) [83] or Weighted Fair Queuing (WFQ) [84, 85] serves packets of different traffic classes with a predefined fraction of the forwarding capacity and Weighted Round Robin (WRR) [86] can

be viewed as an easy to implement approximation of WFQ. Earliest Deadline First (EDF) [87, 88, 89] requires deadlines in the packet headers and chooses the packet with the earliest deadline for transmission which requires searching or sorting in real-time. The Modified Earliest Deadline First (MEDF) works on self-assigned deadlines and is easier to implement than EDF. Essentially, it achieves a delay advantage for high-priority packets [90, 76].

2.3.3 Admission Control

The above approaches try to avoid congestion by providing enough capacity and by preferring high-priority traffic in the routers. However, they do not limit the amount of high-priority traffic in the network, which is the actual cause for packet loss and delay. This is done by admission control (AC), i.e., flows need to be explicitly admitted for the transmission of a declared rate of high-priority packets. Hence, the QoS of admitted flows can be guaranteed at the expense of flow blocking. The transmission rate of the flows is also controlled by a traffic conditioner, i.e. spacer or policer, like above. Note that the capacity of a link depends on the underlying physical communication infrastructure and that the QoS features of the network layer must be supported by the link layer. This is however out of the scope of this work.

In circuit-switched networks like the telephone system, the existence of connections is coupled with the corresponding physical resources that are exclusively dedicated to them. Thus, congestion cannot occur. In packet-switched but still connection-oriented architectures like ATM, resources are explicitly reserved together with the setup of a virtual channel connection (VCC). The IP technology is even unaware of the connection concept which makes its management simple. However, this complicates the establishment of reservations because they must be associated with a packet stream.

Figure 2.7 gives a schematic overview of the relation between AC and the reservation process for a flow [18] in IP networks. Usually, a reservation request is signalled by end systems to the reservation process of a router by the means of

a resource reservation protocol. The request contains the QoS requirements (e.g. delay constraints or the traffic class), traffic descriptors (e.g. the data rate) [91], and the flow specifiers that characterize the packets of the flow. The reservation process first checks the authorization of the flow using the policy control module. Then, AC decides, based on the flow specifiers, whether the new reservation can be supported without violating the QoS requirements of the already admitted flows and the new flow. The flow specifiers are propagated to the packet classifier in the router. The traffic conditioner receives the traffic descriptors and the packet scheduler is notified about the QoS requirements. If the reservation is established, the data packets are associated with the corresponding reservations by the classifier. The traffic conditioner checks whether the data flow behaves according to the traffic descriptors and takes appropriate actions to avoid congestion. Finally, the packet scheduler gives preferential treatment to packets with reservations.

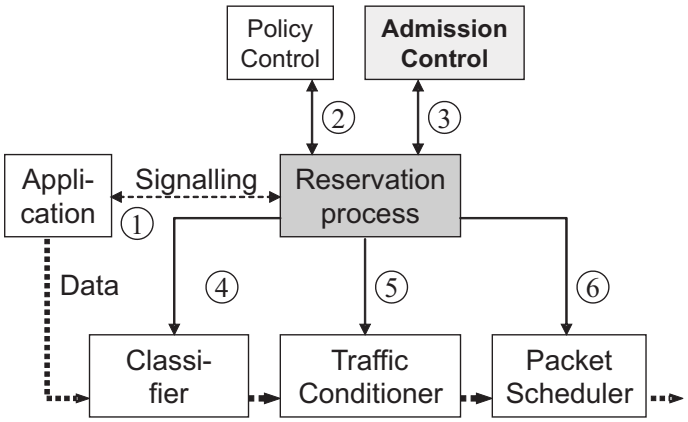


Figure 2.7: *Admission Control is part of the reservation process.*

The key function for the reservation process is AC. It can be implemented

by a wide variety of different approaches and there are no comprehensive studies that classify and compare them in a uniform way. The flow information must be stored at each router along its path which constitutes a reservation state. Thus, reservations introduce the connection concept into the IP world.

The Integrated Services [92, 93] concept uses the Resource Reservation Protocol (RSVP) [18] to signal the resource requirement along a path on a hop-by-hop basis. The intermediate nodes treat individual flows separately with regard to classification, policing, and scheduling, which is quite a heavy overhead.

DiffServ can be enhanced by AC, e.g., if AC is performed only at the border routers and the traffic conditioner marks the packets with the corresponding DSCP [94]. Then, core routers can continue with their simple PHB-dependent operation and remain unaware of individual flows [95].

2.3.4 Network Reliability

The fault-tolerance of a network to local outages is called resilience. It means that data flows reach their destination although a failure happened in the network that has affected the physical communication medium of the communication path. In traditional telephone systems most vital parts are laid out redundantly [96, 97], e.g., power supply, switching fabrics, processors, and communication lines. If one item fails, the corresponding backup item takes over the functionality. This strategy made these conventional telecommunication system very reliable and led to a high target availability of up to 99.999% which corresponds to 5 minutes downtime per year. This feature is know as the “five nines” [98]. In contrast, the current Internet infrastructure provides an availability of only 99% which corresponds to 15 minutes downtime per day [99]. The five nines are a requirement for business customers because communication is a vital component of their work flow. Companies are interconnected with their branch offices and with customers. A communication failure impedes the daily operation or even makes it impossible which translates directly into a loss of money. As a consequence, business users depend on reliable communication and they are willing to pay for it. In fact, the

major impairment of speech quality in the tier-1 IP backbone network of SPRINT in 2001 resulted from link failures and not from queuing delay [100].

In recent year, router vendors have addressed this need and offer highly reliable equipment in their portfolio. They increase the availability of their hardware devices by a redundant provisioning of essential components like in conventional switches [101, 102, 103, 104, 105].

However, there is an essential difference between traditional connection-oriented telephone technology and connectionless IP networks. Connection-oriented technologies are based on states for each individual flow in each switching node. To increase the reliability, the reliability of the nodes must be increased and backup communication lines must be provided. As an alternative, a backup connection for each individual flow can be set up over a disjoint path to provide a hot standby but this is a considerable overhead. In any case, 100% extra line capacity must be provided.

As current IP technology does not depend on states in the network, traffic can be simply deviated around the failure location if the routing adapts to the modified working topology. Since the line capacity is not bound to any connection, the extra capacity can be shared by different flows in different outage scenarios. This saves costs and makes rerouting an attractive means to increase the resilience of a network.

Although MPLS is also a stateful technology, it does not introduce a per flow state in the routers since it works on an aggregate basis. In addition, an LSP usually survives the lifetime of individual flows. Since MPLS is a powerful means for traffic engineering, we use it for rerouting purposes in Chapter 4.

2.3.5 Prototype Implementations of NGN Architectures

The need for NGNs has pushed several pilot projects to engineer and to test potential NGN architectures. All of them enhance today's Internet infrastructure by QoS mechanisms. The most famous one is the Internet2 initiative [106] that

interconnects north-american universities. The European Union promotes information society technologies (IST) and offers funding for projects in the so-called framework programme (FWP). The TEQUILA project [107] from the 5. FWP has concentrated on the definition of service level specifications and the setup of a QoS capable network that is based on loop control mechanisms using traffic measurements. The AQUILA project [108] has also been funded within the 5. FWP and focused on the definition of different traffic classes and their respective admission control algorithms.

This work is settled in the context of the KING project [109] which stands for "Key Components for the Internet of the Next Generation". The project duration is from 1 October 2001 until 30 September 2004 and it is funded by the Bundesministerium für Bildung und Forschung (BMBF) of the Federal Republic of Germany and the Siemens AG. The project is led by Siemens AG and 7 German research institutes are participating. The institutions are

- Siemens AG, Information and Communication Networks
- Fraunhofer-Gesellschaft, Institute for Communication Systems, Munich
- Fraunhofer-Gesellschaft, Institute for Open Communication Systems, Berlin
- University of Duisburg-Essen, Institute for Experimental Mathematics, Computer Network Technology Group
- University of Karlsruhe (TH), Institute of Telematics
- Technical University of Munich, Institute of Communication Networks
- University of Stuttgart, Institute of Communication Networks and Computer Engineering, and
- University of Würzburg, Institute of Computer Science, Department of Distributed Systems

KING is the first research project regarding NGNs that combines the QoS aspects and reliability issues, and suggests a comprehensive concept for resilient QoS networks. Basically, the network is operated is a DiffServ-like manner and AC limits the traffic volume to avoid overload. To keep the core network simple, traffic conditioners control the profile of the flows only at the network edge and mark the packets with the corresponding DSCP. Therefore, reconfiguration of policers on the backup path of a flow is not necessary if traffic rerouting occurs due to a network failure. In addition, AC limit the traffic to such a level that rerouting in protected failure scenarios does not lead to congestion on backup paths.

3 Network Admission Control

In this chapter, we give an overview of different admission control (AC) methods and present a new classification, where budget-based network AC (BNAC) plays an important role. We present a taxonomy for BNAC with respect to resource allocation and explain the different NAC types in detail. In the recent years, many different protocols and systems for BNAC have been proposed but no comprehensive comparison of these methods has been made so far. We suggest enhanced algorithms for link dimensioning and illustrate the concept of economy of scale. Then, we extend the link dimensioning approach towards BNAC-specific network dimensioning. We use the ratio of offered traffic and required capacity as performance measure for extensive comparisons among the different BNAC types regarding resource efficiency. Thereby, we focus on network-relevant parameters like the traffic matrix, topology, and routing. The combination of AC and network resilience is a novel issue and has never been addressed before in literature, although it is important for reliable communication supporting QoS. Therefore, we extend the capacity dimensioning framework to resilience requirements and compare their resource efficiency. Finally, we give algorithmic recommendations for the configuration of NAC budgets in real networks including fairness, efficiency, runtime, and resilience aspects.

3.1 Overview of Admission Control

We can divide AC into different categories. They differ in the quality of their QoS guarantees, in their scope and operation. First, we distinguish the scope. Link AC (LAC) gives answer to the question: how much traffic can be supported on a single link without violating the QoS requirements? Network AC (NAC) needs to protect more than one link with an admission decision and limits the number of flows such that their QoS requirements can still be supported by a network. Thus, NAC is a distributed problem and takes the path of a flow into account [110].

3.1.1 Link Admission Control

QoS criteria are usually packet loss and delay constraints and they are often formulated in a probabilistic way. The packet loss probability must be lower than a predefined objective and a certain percentile of the packet delay distribution must be lower than a given threshold. Bursty traffic requires more bandwidth for transmission than its mean rate to keep the queuing delay low. LAC takes the queuing characteristics of the traffic into account and determines the required bandwidth to carry flows over a single link without violating the QoS constraints.

Effective Bandwidth

The effective bandwidth of a flow is an additive amount of bandwidth for bandwidth accounting on a link. It is large enough to assure that the QoS requirements of all flows are met in the interplay with other admitted flows. The computation of the effective bandwidth can be based on declared or measured traffic parameters, e.g. mean rate or maximum burst size. The effective bandwidth can depend on the considered link capacity as it takes statistical multiplexing gain into account. A good overview on effective bandwidth methods can be found in [111, 112, 113]. However, the concept is not limited to special formulae. We describe some simple examples for bandwidth accounting. They assume certain traffic models and can be viewed as a realization of the effective bandwidth concept.

- With peak rate allocation, each flow declares its maximum rate. The AC makes sure that the sum of all peak rates is not larger than the link bandwidth. To achieve that objective, the AC entity records the traffic descriptors of individual flows to increase and decrease the reserved bandwidth when flows are admitted or terminate. This flow-related information is called an AC or reservation state. The peak rate allocation scheme requires only a small buffer to prevent packet loss and leads to little delay although delay is not explicitly taken into account.

- The $M/M/1$ queuing model [114] might be appropriate to compute the maximum load that a link can support without violating certain delay bounds when traffic flows have irregular inter-arrival and service-times, i.e. variable packet sizes. A traffic description for traffic with Poisson queuing properties or better is given in [115, 116] such that corresponding policers can be constructed.

- The $N \cdot D/D/1$ queuing model assumes that homogeneous flows with a deterministic packet inter-arrival and service time, i.e. constant packet sizes, are multiplexed onto a single link. Simple queuing formulae enable the computation of delay percentiles. This model is suitable for constant bitrate real-time traffic flows. An application of the formula can be found in [117].

- Many other methods, e.g. rate envelope multiplexing (REM), are discussed in [1], which is a good summary of research efforts regarding effective bandwidth in the context of ATM in the 1990s.

The suitability of these effective bandwidth methods depends on the required QoS. Hence, different approaches may be used to implement different traffic classes [110, 118], e.g. interactive real-time traffic requires stricter delay requirements than non-interactive traffic streaming. LAC can be further subdivided into parameter-based LAC (PLAC), measurement-based AC (MBAC), and

experience-based AC (EBAC), which is a new concept that combines both approaches.

Parameter-Based LAC

With PLAC, the effective bandwidth calculation for a flow is based on its traffic descriptor (e.g. peak rate and burstiness) that is declared by the application or a proxy. The adherence of the source is part of the traffic contract and it is essential for the correctness of the AC decision. Therefore, the traffic descriptors are usually controlled by policers that discard excess packets. In general, for every concise traffic description, a policer can be constructed. To avoid losses at the policer, traffic descriptor are declared larger than sufficient. As a consequence, the respective effective bandwidth is also larger than required and when blocking occurs, the link bandwidth is not yet utilized to a critical degree. This shortcoming is addressed by MBAC and EBAC.

Usually, PLAC is only applied to non-elastic real-time traffic but the authors of [119] adapt this method for TCP streams to guarantee a certain goodput.

Measurement-Based AC

With MBAC, the effective bandwidth calculation for a flow is based on a traffic descriptor that is derived from instantaneous measurements of that flow. As mentioned above, flows often reserve a larger rate than required to get enough bandwidth when needed or the traffic descriptors are unknown. Thus, MBAC schemes are more effective than PLAC schemes without violating the QoS requirements of the traffic because the measured traffic profile is a tighter description than a traffic descriptor declared by applications. Since meaningful measurements of a specific flow are only available after a certain measurement time [120], the effective bandwidth of a flow is initially computed based on a declared traffic descriptor, when it is required for the AC decision. Some MBAC methods require the measurements of individual flows but most of them are based on aggregate measurements of the admitted traffic [121].

Like with PLAC, flows are accepted or rejected based on worst case traffic descriptors and the amount of already admitted traffic. However, these worst case traffic descriptors are substituted after some time by some more economic flow characterizations.

MBAC with Flow-Specific Measurements MBAC with flow-specific measurements (F-MBAC) measures the traffic descriptors of each flow individually. As soon as the confidence in the measured results is sufficiently large, the effective bandwidth, that has been initially calculated for that flow based on declared traffic descriptors, is substituted by an update, which is computed based on the measured traffic descriptor. Examples of flow-specific measurement methods are given in [122, 121, 123].

MBAC with Aggregate Measurements The effective bandwidths of admitted flows are only required to determine the free bandwidth on the link. For this purpose, the rate of the admitted aggregate is sufficient and most MBAC methods measure properties of the admitted traffic aggregate instead of individual flows. MBAC with aggregate measurements (A-MBAC) has two advantages. The measurement is simpler as no per flow measurement states have to be kept and the statistical properties of a stationary aggregate are more stable. On the other hand, new flows are admitted and others terminate which makes the aggregate a non-stationary process which must be carefully observed [124, 125]. Comparisons of different A-MBAC approaches can be found in [126, 127, 128, 129, 130, 131, 132, 133].

Experience-Based AC

Experience-based AC (EBAC) [134] is a combination of PLAC and MBAC. It computes the effective bandwidths for flows based on declared traffic descriptors but it respects an overbooking factor for the AC decision which is computed based on past measurements. More precisely, EBAC measures and records both

traffic rates and the aggregate effective bandwidth and correlates them with respect to time. Based on these traces, the time-dependent reservation utilization is derived. The 99% percentile of the distribution function of the reservation utilization yields a value that is only rarely exceeded and, therefore, we take its reciprocal φ for overbooking. Overbooking means that the AC entity can allocate a φ multiple of the physical bandwidth. EBAC works only for sufficiently large links and traffic aggregates where the reservation utilization is fairly constant, for a large number of reservations, and it assumes that traffic properties change only slowly. Unlike MBAC, EBAC does not require instant measurements as the overbooking parameter is derived from traces. On the one hand, this is an important advantage of EBAC compared to MBAC because the EBAC approach has been successfully implemented with standard machines in the KING project while instant and sufficiently exact measurements for MBAC are not feasible by standard routers. On the other hand, EBAC cannot achieve as high resource utilizations like MBAC because it possesses less information about the actual current link utilization. It must respect a safety margin in terms of unallocated capacity since the utilization of individual reservations can vary over time. In addition, the control loop of EBAC is looser than the control loop of MBAC, which calls for another safety margin when traffic mixes change over time.

3.1.2 Network Admission Control

In contrast to LAC, NAC is the mechanism that admits a flow through an entire network and not only on a single link. Therefore, NAC takes the paths of the flows into account, i.e., it requires information about the routing and load balancing in the network. In addition, flows enter the network independently of each other at different ingress router. This makes NAC a distributed problem. To admit a flow, budget-based NAC (BNAC) methods apply LAC at different NAC instances in the network that have a virtual capacity budget instead of a link bandwidth. Feedback-based NAC (FNAC) methods use distributed instant measurements to decide whether a new flow can be accepted. We further describe these approaches

in the following.

Budget-Based NAC

BNAC methods are based on distributed budgets and we differentiate them according to their budget types. The budgets have virtual capacities that relate either to specific links, b2b aggregates, or combinations and sets thereof. They may be located at different NAC control points, e.g., in a central entity, only at the network border, or also at intermediate core routers. Figure 3.1 shows that each flow is associated with a set of distributed budgets and it is admitted by BNAC if AC decisions for all budgets of that set are successful. Theses AC decisions work according to LAC. The virtual capacity of the budgets must be assigned in such a way that the physical network resources are not unintentionally overbooked and that different b2b aggregates encounter fair flow blocking probabilities. We propose algorithms for that challenge in Section 3.8.

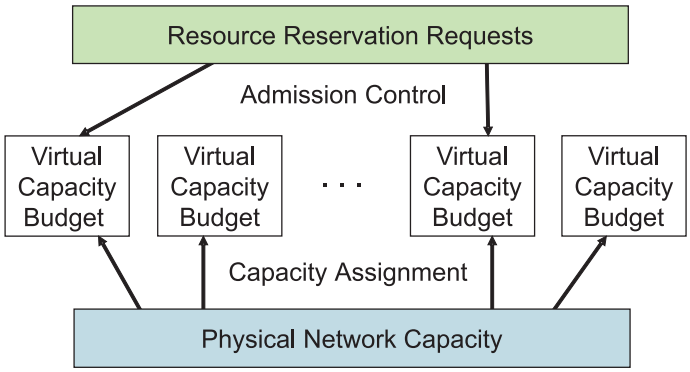


Figure 3.1: *BNAC methods differ in the number and location of their virtual capacity budgets and in the set of consulted budgets to admit a particular flow.*

The most intuitive NAC method is the link-by-link AC which uses one budget per link. If NAC is based on virtual tunnels, one b2b budget for each b2b aggregate is used. There are many protocols and systems whose operation can be viewed as BNAC from a resource allocation point of view, i.e., they accept or reject in principle the same flows. In Section 3.2 we identify four main groups according to their resource allocation strategy: link budget (LB) NAC, ingress and egress budget (IB/EB) NAC, border-to-border budget (BBB) NAC, and ingress and egress link budget (ILB/ELB) NAC [135]. We describe them in detail in and illustrate them with examples.

Feedback-Based NAC

Other approaches rely on a quality feedback of intermediate routers, so we call them feedback-based NAC (FNAC). The sender issues one or several probe messages to the destination and they are discarded intentionally by intermediate routers if the network is overloaded. The overload is diagnosed by local traffic measurements. If a certain proportion of the probes returns, the flow is admitted, otherwise it is rejected [136, 137, 138, 139, 140, 141]. This approach can be well combined with MBAC methods [142, 143]. The authors of [144, 145] renounce on the assistance of intermediate routers and perform the acceptance decision based on the normal packet loss ratio that is evaluated by probe messages. A similar implicit approach has been taken to perform AC for TCP traffic [146, 147, 148]. In this case, intermediate routers detect overload and block new TCP flows by discarding their initial SYN packets during their setup phase.

3.1.3 Overview of General AC Methods

Figure 3.2 gives an overview of AC methods. We distinguish primarily between LAC and NAC. We can further subdivide LAC into MBAC, PLAC, and combined approaches like EBAC. NAC differentiates between FNAC, which is related to MBAC, and BNAC, which is the logic extension of LAC to networks. BNAC

is the main focus of this work and we explain and compare the different BNAC approaches in this chapter.

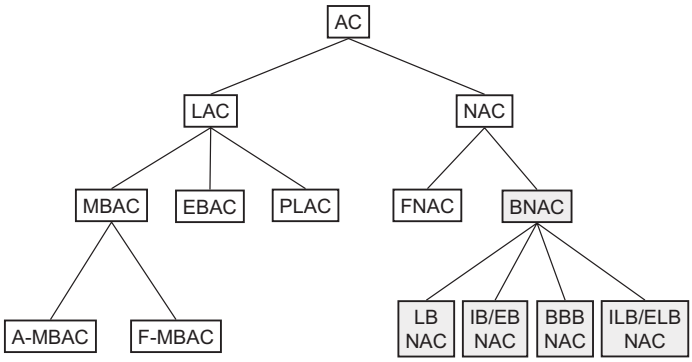


Figure 3.2: *A Taxonomy for admission control methods.*

Note that this classification does not claim to be complete or exclusive because protocols and system may be classified using different aspects [149, 150]. For example, the Next Steps in Signaling (NSIS) working group [151] of the IETF designs new protocols for signaling flow-related information along its path in the network. They also consider QoS signaling protocols [152] but they have their focus on signaling aspects and not on AC.

3.2 A Taxonomy for Budget-Based Network Admission Control

In this section we present four fundamentally different BNAC types that we categorize according to their resource allocation granularity. We start with some comments on the basic approach and notation. Then, we present the elementary

operations for each elementary BNAC concept and we give examples for standardized, implemented, or proposed protocols or systems that work according to that principle.

3.2.1 Basic Approach and Notation

Like illustrated in Figure 3.1, the various BNAC methods differ in the number and the location of the BNAC instances and budgets, and in the set of budgets that have to be consulted for the admission of a particular flow. Each BNAC entity records for each of its controlled budgets the effective bandwidth[1] of the admitted flows $\mathcal{F}$ in place. When a new flow arrives, it checks whether the relevant budgets have enough capacity to accommodate the new flow's effective bandwidth together with the effective bandwidth of the already established flows. If so, the flow is accepted, otherwise it is rejected by that BNAC entity. Several such budgets may be tested in different BNAC entities and all decisions must be positive for the b2b admission of a flow.

We use the following notation in this work. A networking scenario $\mathcal{N} = (\mathcal{V}, \mathcal{E}, u)$ is given by a set of routers $\mathcal{V}$, set of links $\mathcal{E}$, and a routing function u. To avoid special cases, we assume a pure transit network, i.e. traffic can enter and leave the network at all routers, hence, there are $|\mathcal{V}| \cdot (|\mathcal{V}| - 1)$ different b2b relationships[2]. The b2b traffic aggregate with ingress router v and egress router w is denoted by $g_{v,w}$ $(v, w \in \mathcal{V})$ and the set of all b2b traffic aggregates is $\mathcal{G}$. The routing function $u(l, g_{v,w})$ indicates the percentage of the traffic rate $c(g_{v,w})$ traversing link l. This is a very general approach because it can cover both single- and multi-path routing.

[1] The effective bandwidth of a flow depends in general both on its traffic characteristic and the size of the carrying link. More precisely, it converges to a lower limit for increasing bandwidth. As real-time flows can be efficiently multiplexed, this lower limit is reached already at moderately large bandwidth in the order of 10 *Mbit/s* [117]. This justifies the use of a single effective bandwidth for a flow for the transport over several links through a high-speed network.

[2] $|\mathcal{X}|$ is the cardinality of set $\mathcal{X}$.

3.2.2 Link Budget Network Admission Control

The link budget (LB) NAC is the link-by-link application of LAC methods. It is probably the most intuitive BNAC approach.

Basic Operations

The capacity $c(l)$ of each link l in the network is managed by a single link budget LB_l with size $c(LB_l)$. It may be located at the router sending over that link or in a centralized database. A new flow $f_{v,w}^{new}$ with ingress router v, egress router w, and bitrate $c(f_{v,w}^{new})$ must pass the AC procedure for the LBs of all links that are traversed in the network by $f_{v,w}^{new}$ (cf. Figure 3.3). The AC procedure will be successful if the following inequality holds

$$\begin{aligned} &\forall l \in \mathcal{E} : u(l, g_{v,w}) > 0 : \\ &c(f_{v,w}^{new}) \cdot u(l, g_{v,w}) + \sum_{f_{x,y} \in \mathcal{F}(LB_l)} c(f_{x,y}) \cdot u(l, g_{x,y}) \leq c(LB_l). \end{aligned} \tag{3.1}$$

Examples

The most significant criterion for the resource allocation of the LB NAC is that any new flow is accepted as long as the available capacity of a link suffices to accommodate it together with the existing reservations. It is met for many systems and protocols that we can group into four fundamentally different classes according to their key ideas.

Resource Reservation Protocols The simplest idea for resource guarantees along the path of a flow is to signal its demand from hop to hop and perform LAC for each link. The information about the admitted flows regarding their demand is stored locally in the routers controlling the outgoing links and the data records are called states. The signaling is achieved by resource reservation protocols and many different specifications and implementations have been proposed.

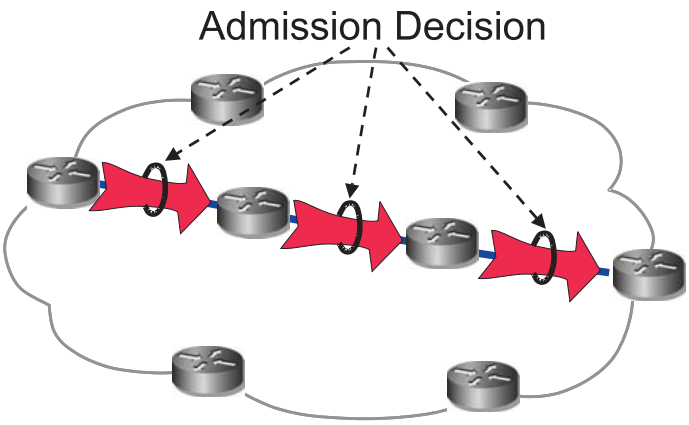


Figure 3.3: *Network admission control based on link budgets.*

- The Resource Reservation Protocol (RSVP) [18] has been proposed by the IETF as an accompanying protocol to IP without routing functionalities and whose signaling messages are carried like ICMP messages. It is not a transport protocol because the actual payload is carried by ordinary UDP or TCP streams. RSVP implements the Integrated Services paradigm [92, 153] that we have addressed in Section 2.3.3. Both unicast and multicast applications are supported and different reservation styles (e.g. shared reservations) are possible.

  To initiate a reservation with RSVP, the sending node issues a so-called PATH message that establishes a PATH state (PS) in the intermediate hops on the way to the destination machine. The destination responds with a RESV message that visits the intermediate routers in the reverse direction using the previous hop information of the PS. On that way, the RESV states (RS) are established and the routers usually reserve the required re-

sources for the requesting flow, which is an action outside of the scope of RSVP. This ensures that resources are reserved in each router in downstream direction. The first pass from the sender to the receiver (PATH msg.) collects advertising information that is delivered to the destination to enable it to make appropriate reservation requests. The actual reservation is made on the way back to the sender (RESV msg.). Hence, RSVP uses a two-pass signaling approach, also known as one-pass with advertising (OPWA). Explicit PATHERR and RESVERR messages indicate errors, and TEARDOWN messages tear down the connection and remove the states in the routers.

In case that applications quit without an explicit TEARDOWN message, RSVP uses a soft state approach for state cleanup, i.e., the states are removed automatically after a certain time. Therefore, the states need to be refreshed by additional update messages that are signalled periodically every 30 seconds. As this leads to a high signaling overhead, efficient implementations and protocol enhancements have been proposed [154, 155, 156, 157, 158]. The implementation of the RSVP engine itself has a big influence on the capacity of a router in terms of the number of manageable flows [159, 160].

- The Boomerang protocol [161] aims at reducing some part of the overhead which is induced by the generality of RSVP. RSVP is receiver initiated since it is conceived for multicast sessions, too. Therefore, it requires one pass to collect the path information from sender to receiver and another one to perform the actual reservation. With Boomerang, the sender generates a Boomerang message that is forwarded hop by hop to the receiver and the intermediate routers understanding Boomerang perform a reservation. As soon as the message arrives at the receiver, the full reservation is already in place. The receiver does not even need to process the message, it just bounces the message back to the sender to acknowledge the successful reservation setup. Optionally, the return channel of a bidirectional

session may be reserved on the way back. Note that a different path may be taken for that purpose. There are additional simplifications compared to RSVP, e.g., concerning the refresh message handling [162, 163].

- YESSIR (YEt another Sender Session Internet Reservation) [164] is a reservation protocol that is based on RTP [165, 19]. RTP is usually a wrapper for application data and adds sequence numbers, time stamps and other identifiers. Each session is controlled by the Real-time Transport Control Protocol (RTCP). Senders and receivers periodically send sender and receiver reports (SR, RR). SRs contain throughput and other information about the last report interval and allow, e.g., the derivation of the current round-trip time in the network. RRs indicate packet loss and delay statistics among others, which is useful for adaptive applications. YESSIR reservation messages are piggybacked at the end of RTCP SR or RR messages, possibly enhanced by additional YESSIR-specific data, carried in IP packets with router-alert option, i.e., they are intercepted by routers and processed by those supporting this option. As with Boomerang, reservations are triggered by the sender and both unicast and multicast is supported like in RSVP. If a router along the way is not able to provide the requested resources, the exact reasons for the reservation failure can be noticed. This helps the end systems to either drop the session or to decrease the requested bandwidth for the reservation. YESSIR also relies on the soft state approach.

- The Internet Stream Protocol version 2 (ST2) [166] was an experimental resource reservation protocol intended to provide end-to-end real-time guarantees over an internet. However, it is more than a pure resource reservation protocol because it replaces IP at the network layer. Both ST2 and IP apply the same addressing schemes to identify different hosts. ST2 and IP packets differ in the first four bits, which contain the internetwork protocol version number: number 5 is reserved for ST2 (IP itself has version number 4). As a network layer protocol, like IP, ST2 operates indepen-

dently of its underlying subnets. The ST2 protocol disappeared completely and IPv4 prevailed. It pursued a so-called hard state approach for reservation requests, i.e., the flow records have to be explicitly removed at the end of a session. Comparisons of RSVP and ST2 can be found in [167, 168].

- The connection setup in ATM (e.g. for CBR or VBR connections) reserves the network resources also like a resource reservation protocol [55, 56].

Resource Reservation Protocols with State Aggregation The Border Gateway Resource Reservation Protocol BGRP [169] has been conceived for inter-domain use and to work in cooperation with the Border Gateway Protocol (BGP) for routing. It is used for reservations between border routers only. BGRP addresses the scalability problem directly since it is designed to aggregate all inter-domain reservations with the same autonomous system (AS) gateway as destination into a single funnel reservation, no matter of their origin. The concept foresees a permanent BGRP reservation for each destination AS such that packet classification can be based on the network mask of the destination AS.

We explain briefly how BGRP signals a sink tree reservation (cf. Figure 3.4). A PROBE message is sent from a source border router to a destination border router and registers the visited border routers. Upon the reception of a PROBE message, the border routers check for available resources, and forwards the PROBE packet towards the destination. The destination border router terminates this process. It converts the PROBE message into a GRAFT message and inserts an ID that identifies its sink tree. The GRAFT message travels back on the collected path. The required reservation states are established and marked with the ID, or they are updated if they already exist. The PROBE and GRAFT messages contain only a relative reservation offset, therefore, the communication for GRAFT messages must be reliable (e.g. using TCP). BGRP is a soft state protocol, therefore, neighboring routers exchange explicit REFRESH messages to keep the reservation alive. Due to the different signaling, the reservation states of the individual reservations are aggregated and lead to state scalability in the

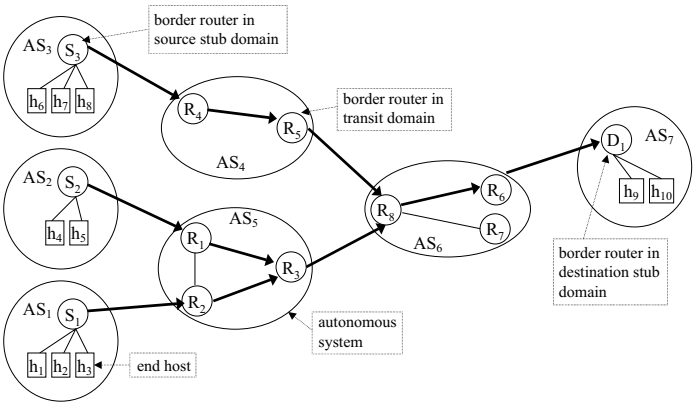


Figure 3.4: *BGRP Signaling.*

routers but from the resource allocation point of view the same result is obtained. A similar approach is presented in [170].

Bandwidth Brokers A bandwidth broker (BB) [171, 172, 173, 174, 175, 176, 177] is a central unit within an administrative domain that controls the access of high-priority flows. Therefore, an ingress routers has to redirect a reservation request to the BB and receives a positive or negative response which also triggers the configuration of the classification, marking, and policing parameters at the ingress router. The BB holds also per flow states, is has full information of the routing and of the link capacities within its domain. Therefore, it is able to perform an AC decision based on the same information like an RSVP engine. Thus, this systems looks completely different from an implementation point of view but is behaves like the LB NAC from a resource allocation point of view.

Token-Based Implementation The Edge Assisted Quality of Service (EQOS) protocol [178] has no single point of failure and avoids states in intermediate routers. Basically, the available link capacities are recorded in a token which is passed on among the ingress routers. Flows request a reservation at the ingress routers and are admitted if the token shows enough capacity on the links that will be used by the flow. If so, the request is admitted and the demand is subtracted from the resources recorded in the token. If a flow departs, the same amount is added. To make the system resilient against token loss and other inconsistencies, the reservation states are stored at the ingress routers such that a consistent token state can be restored within one token round trip time. Hence, this architecture is essentially a circulating bandwidth broker with a database stored in a distributed fashion. A drawback of this approach is that flow requests face an admission delay until the token is available. If the token round-trip time is sufficiently small, the admission delay can be neglected and the resource allocation is based on the same information like for the LB NAC.

Stateless Core Reservation Protocols The so-called stateless core reservation protocols avoid reservation states in the core network at the expense of measurements or increased response time. Reservation states are held only at ingress routers which send reservation tickets in regular time intervals from source to destination. The intermediate core routers count the overall rate of the reservation tickets within the last time interval. Thus, they can infer the reserved rate $c_{resv}(l)$ for each outgoing link l. Upon a new flow request, a packet with an unadmitted ticket is sent from source to destination. The core routers count those requests separately by $c_{new}(l)$ and if they dispose of available resources, i.e. $c_{resv}(l) + c_{new}(l) + c(f^{new}) \leq c(l)$, the ticket is forwarded and the destination returns it to the source and the flow is admitted. Otherwise, the ticket is discarded by the routers and the flow request is rejected at the ingress router if no positive feedback arrives after a certain time. The basic mechanism is simple and it is implemented by different protocols, however, all of them have manifold problems, e.g. regarding timing accuracy, and they require significant measure-

ment operations by core routers. The following protocols and systems operate according to that principle.

- The Scalable Resource Reservation Protocol for the Internet (SRP) [179, 180, 181] implements exactly the principle.
- The Stateless Core approach in [182, 183] combines AC according to that scheme with a distributed packet scheduling for real-time services.
- The Resource Management in Differentiated Services IP Networks framework (RMD) [184] aims at a lightweight signaling for QoS enabled access networks for wireless applications. However, it can be also used in other environments.

These methods differ from FNAC (cf. Section 3.1.2) by the fact that the traffic measurements in core routers are based on explicit reservation tickets and not on actual traffic.

Network Calculus With "Network Calculus", a maximum delay bound can be computed for a flow depending on its path through the network including the effects of spacers [185, 186, 187]. Its results can be used for NAC purposes to extend LAC based on simple peak rate allocation but in literature it is rather used for investigations. Recently, network calculus has been enhanced to provide statistical guarantees [188] instead of worst case bounds which leads to a better resource utilization.

State signaling in the core network with conventional resource reservation protocols, single points of failures with bandwidth brokers, increased response times with a token-based AC, or measurements in core routers with so-called stateless core reservations are problematic. The following three BNAC methods store all crucial AC information at the network edge, i.e., all budgets-related to a flow can be consulted at its ingress or its egress border router, so the above mentioned drawbacks are avoided.

3.2.3 Ingress and Egress Budget Network Admission Control

The ingress and egress budget (IB/EB) NAC is the simplest BNAC version with AC control only at the border routers.

Basic Operations

The IB/EB NAC defines for every ingress node $v \in \mathcal{V}$ an ingress budget IB_v and for every egress node $w \in \mathcal{V}$ an egress budget EB_w that must not be exceeded. A new flow $f_{v,w}^{new}$ must pass the AC procedure for IB_v and EB_w and it is only admitted if both requests are successful (cf. Figure 3.5). Hence, the following inequalities must hold

$$c(f_{v,w}^{new}) + \sum_{f \in \mathcal{F}(IB_v)} c(f) \quad \leq \quad c(IB_v) \tag{3.2}$$

$$c(f_{v,w}^{new}) + \sum_{f \in \mathcal{F}(EB_w)} c(f) \quad \leq \quad c(EB_w). \tag{3.3}$$

Flows are admitted at the ingress irrespectively of their egress router and at their egress router irrespectively of their ingress routers, i.e., both AC decisions are decoupled. This entails that the capacity managed by an IB or EB can be used in a very flexible manner. However, the network must be able to carry all – also pathological – traffic patterns that are admissible by the IBs and EBs with the required QoS. Hence, sufficient capacity must be allocated or the IBs and EBs must be set to small enough values.

If we leave the EBs aside, we get the simple IB NAC, so only Equation (3.2) must be met for the AC procedure.

Examples

The IB NAC idea originates from the DiffServ context [67] where traffic is admitted only at the ingress routers without looking at the destination address of

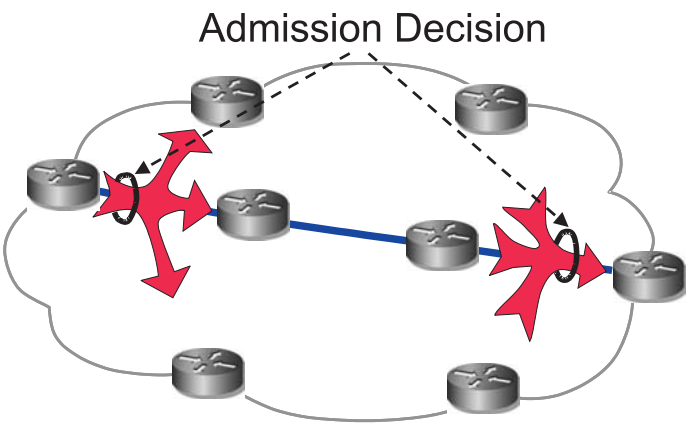


Figure 3.5: *Network admission control based on ingress and egress budgets.*

the flows. The QoS should be guaranteed by a sufficiently low utilization of the network resources by high quality traffic. To avoid any confusion: DiffServ is a mechanism for the forwarding differentiation of classified traffic while the IB NAC is just one concept among many others for the management of network resources which was mentioned in [189]. The IB/EB NAC has been implemented by the AQUILA project [190, 191, 192]. In addition, they introduce a layered approach to redistributed budget capacities in a scalable manner.

3.2.4 B2B Budget Network Admission Control

The b2b budget (BBB) NAC corresponds to AC on logical tunnels through a network with fixed bandwidth. Only the ingress and the egress of the tunnel are fixed but packets may be forwarded in the network over multiple paths.

Basic Operations

The BBB NAC takes both the ingress and the egress border router of a flow $f_{v,w}$ into account for the AC procedure, i.e., a b2b budget $BBB_{v,w}$ manages the capacity of a virtual tunnel between v and w. Figure 3.6 illustrates that a new flow $f_{v,w}^{new}$ passes only a single AC procedure for $BBB_{v,w}$. It is admitted if this request is successful, i.e., if the following inequality holds

$$c(f_{v,w}^{new}) + \sum_{f \in \mathcal{F}(BBB_{v,w})} c(f) \leq c(BBB_{v,w}). \tag{3.4}$$

The BBB NAC can also avoid per flow states within the network because the $BBB_{v,w}$ may be controlled at the ingress or egress router. In contrast to IB/EB NAC, the BBB NAC is able to exclude pathological traffic patterns but the capacity of a BBB is bounded to one specific b2b aggregate. However, this makes flexible resource allocation impossible since the capacity cannot be used for other traffic aggregates with different source or destination.

Examples

The implementations of the BBB NAC exist mostly as LAC for tunnel implementations that have both fixed capacity and a fixed path.

- The Virtual Path Connection (VPC) in ATM provides a tunnel through a network to accommodate several Virtual Channel Connections (VCCs) [55, 56].
- The ATM Adaptation Layer type 2 (AAL2) multiplexes several AAL2 connections into one VCC of fixed bandwidth [193, 194].
- The aggregation concept of RSVP for IPv4 and IPv6 reservations [195] sets up a virtual pipe of fixed bandwidth over several hops though which many RSVP protected flows can be tunnelled. This removes the reservation states of the individual flows along the tunnel and installs only a single state for the aggregate reservation.

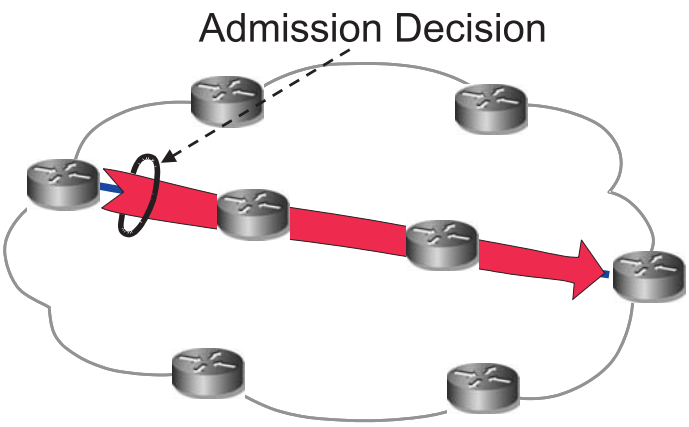


Figure 3.6: *The BBB NAC corresponds to a virtual tunnel.*

- In MPLS, LSPs with guaranteed bandwidth may be set up using the extensions to RSVP for LSP tunnels (RSVP-TE) [47]. Then, AC may be performed for individual flows entering the pipe.

- BGRP is usually applied to aggregates with fixed size multi-point-to-point reservations that are rarely updated. So, it can also serve as an example for scalable tunnel reservations.

- The admission control and the administration of the BBBs must be implemented by a network management system and also the budget configuration should be automated to avoid error prone human interaction. The KING project [2, 109] implements the BBB NAC. Since the concept is a NAC enhanced DiffServ solution, the budget capacity is only associated with b2b relations but not with explicit paths. This eases rerouting in outage scenarios. NAC entities at the border routers administer the budgets

and perform the AC decisions. A central entity, the so-called network control server (NCS), possesses network-wide information about the routing, the traffic matrix, the average traffic profile, additional QoS requirements like desired flow blocking probabilities. It calculates suitable budget sizes according to the algorithms in 3.8 and configures the NAC entities at the border routers. As soon as the budgets are set, the operation of the network is independent of the NCS. Unlike a bandwidth broker, the NCS is not directly involved in AC decisions but it is an automated network configuration center whose tasks must be performed in any network either by human operators or by network management software.

As the resource allocation is not flexible enough, the concept is often implemented in a more dynamic manner, such that the size of the BBBs can be rearranged [196, 197, 198, 199, 200]. Tunnels may also be used hierarchically [57, 201, 59].

3.2.5 Ingress and Egress Link Budget Network Admission Control

The ingress and egress link budget (ILB/ELB) NAC controls flows from the edge but takes their paths into account like the BBB NAC. Like the IB/EB NAC, it allows for resource sharing among flows with the same ingress or egress router, respectively.

Basic Operations

The ILB/ELB NAC defines ingress link budgets $ILB_{l,v}$ and egress link budgets $ELB_{l,w}$ to manage the capacity of each $l \in \mathcal{E}$. They are administered by border routers v and w, i.e., the link capacity is partitioned among up to $|\mathcal{V}|-1$ border routers. In case of single-path IP routing, the links $\{l : ILB_{l,v} > 0\}$, that are associated with v, constitute a logical source tree and the links $\{l : ELB_{l,w} > 0\}$, that are administered in w, form a logical sink tree (cf. Figure 3.7). A new flow

$f_{v,w}^{new}$ must pass the AC procedure for the $ILB_{l,v}$ and $ELB_{l,w}$ of all links l that are traversed in the network by $f_{v,w}^{new}$. The AC procedure will be successful if the following inequalities are fulfilled

$$\forall l \in \mathcal{E} : u(l, g_{v,w}) > 0 \quad : \\ c(f_{v,w}^{new}) \cdot u(l, g_{v,w}) + \sum_{f_{v,y} \in \mathcal{F}(ILB_{l,v})} c(f_{v,y}) \cdot u(l, g_{v,y}) \leq c(ILB_{l,v}), \text{ and} \quad (3.5)$$

$$\forall l \in \mathcal{E} : u(l, g_{v,w}) > 0 \quad : \\ c(f_{v,w}^{new}) \cdot u(l, g_{v,w}) + \sum_{f_{x,w} \in \mathcal{F}(ELB_{l,w})} c(f_{x,w}) \cdot u(l, g_{x,w}) \leq c(ELB_{l,w}). \quad (3.6)$$

There are several significant differences to the BBB NAC. A BBB covers only an aggregate of flows with the same source and destination while the ILBs (ELBs) cover flows with the same source (destination) but different destinations (sources). Therefore, the ILB/ELB NAC has more resource flexibility than the BBB NAC. The BBB NAC is simpler to implement because only one $BBB_{v,w}$ is checked while with ILB/ELB NAC, the number of budgets to be checked is twice the flow's path length in hops. In contrast to the LB NAC, these budgets are controlled only at the border routers. Like with the IB/EB NAC, there is the option to use only ILBs or ELBs by applying either only Equation (3.5) or only Equation (3.6). The concept of ILB/ELB or ILB NAC can be viewed as local bandwidth brokers at the border routers, possessing a fraction of the network capacity.

Examples

The hose model in [202, 203, 204] is a source tree where the sum of the children's link capacities may be larger than the link capacity of a parent. However, the capacity of the hose may be used by any flow from its root to one of its leaves.

An enhancement of the EQOS protocol leads to partitioning of the link capacities among the access routers [205, 206] which implements exactly the ILB NAC idea. Each of the access routers acts on its private bandwidth share like a

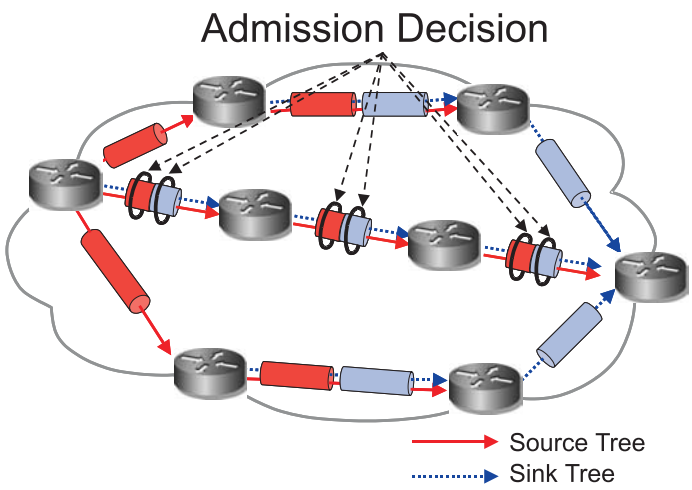


Figure 3.7: *Network admission control based on ingress and egress link budgets.*

local bandwidth broker. The token mechanism is only used to distribute and adapt the network resources in a suitable way. Note that BGRP does not match the ELB NAC because each access border router sees only a prereserved pipe towards a common destination which corresponds to a single BBB.

The ILB/ELB NAC is a new approach and has not yet been implemented by any resource management protocol.

3.3 Capacity Dimensioning for a Single Link

When AC is applied, flow requests can be blocked to prevent overload situations. Our goal is to asses the efficiency of NAC methods by comparing their required resources to achieve the same blocking probability. The blocking probability is determined by the provided resources and the traffic model. We review a multi-rate Poisson model for real-time traffic and the Kaufman & Roberts formula [1] which calculates the blocking probability for that traffic model. We present an efficient implementation of this formula and its inversion yields our capacity dimensioning algorithm. Then, we explain economy of scale and illustrate the sensitivity of capacity requirements to various parameters, e.g. the request size distribution, because this influences the resource efficiency of all AC schemes.

3.3.1 A Simple Model for Real-Time Traffic

The underlying traffic model has an essential impact both on flow blocking probabilities and on capacity dimensioning. We intend to investigate NAC for IP networks which operates on the session level. The inter-arrival time of sessions is exponentially distributed [207, 208]. Therefore, the Poisson model is appropriate for the description of session arrivals [209] which cause reservation requests. It is characterized by an exponentially distributed flow inter-arrival time with rate $\frac{1}{E(A)}$ and an independently and identically distributed call holding time with mean $E(B)$. The quotient $a = \frac{E(B)}{E(A)}$ is the offered load which equals the mean number of active flows in a system without flow blocking. It is a simple number and it is expressed in the pseudo unit Erlang [*Erl*].

As the request profile is multi-rate in a multi-service world like the Internet, we use a simplified multi-rate model. We have $n_r = 3$ different request types r_i, $0 \leq i < n_r$ with request sizes $c(r_i)$. The mean of the request-type-specific inter-arrival and the mean of the call holding time determine the request-type-

specific offered load $a(r_i) = \frac{E(B_i)}{E(A_i)}$. The overall load is $a = \sum_{0 \leq i < n_r} a(r_i)$. The random variable C_t indicates the request rate in case of a flow arrival and the request size probability is calculated by $P(C_t = c(r_i)) = \frac{E(A)}{E(A_i)}$.

The statistical properties of the request types are compiled in Table 3.1. They are chosen in such way that we get a constant mean of $E(C_t) = 256$ Kbit/s and a coefficient of variation of $c_{var}(C_t) = 2.291 \cdot t$ that depends linearly on t. So far, the absolute values $E(A_i)$ and $E(B_i)$ are not fixed but for illustration purposes $E(B_i) = 90$ *s* may be chosen.

Table 3.1: *Request type statistics.*

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

3.3.2 The Kaufman & Roberts Formula for the Computation of Blocking Probabilities

An algorithm for the computation of the blocking probabilities of a multi-rate Poisson model has been presented in [1] (18.1.1, p. 516) and [210, 211]. It is based on discrete units, so we discretize the link bandwidth C into C_u capacity units of size $u_c = 64$ *Kbit/s*. Analogously, $c_u(r_i)$ is the request rate in capacity units u_c. First, auxiliary variables $\tilde{w}[j]$ are calculated.

$$\tilde{w}[j] = \begin{cases} 0 & : j < 0, \\ 1 & : j = 0, \\ \frac{1}{j} \cdot \sum\limits_{0 \leq i < n_r} \tilde{w}[j - c_u(r_i)] \cdot c_u(r_i) \cdot a(r_i) & : 0 < j \leq C_u \end{cases} \tag{3.7}$$

A normalization derives the probability $w[j]$ for j used capacity units on the link.

$$w[j] \quad = \quad \tilde{w}[j] \cdot \Big(\sum_{k=0}^{C_u} \tilde{w}[k]\Big)^{-1} \tag{3.8}$$

The request-type-specific blocking probability $p(r_i)$ depends on the link capacity C_u.

$$p(r_i) \quad = \quad \sum_{j=C_u - c_u(r_i)+1}^{C_u} w[j] \tag{3.9}$$

So far, only flow level but no packet level dynamics have been taken into account. If these are also considered, request rates become subadditive and can be multiplexed more efficiently. This feature can be added by modifying the above equations. However, packet level dynamics introduce another degree of freedom and complexity. Since we are more interested in BNAC than in PLAC issues, we restrict ourselves to a simple peak or mean rate allocation model.

Options for Aggregate Blocking Probability Requests with a large demand have a larger blocking probability than those with smaller demand. As a single number is simpler for comparison purposes, an overall measure for blocking is required. We consider three basically different solutions for that problem.

The average flow blocking probability p_f is the mean blocking probabilities of all flows regardless of their request type. Hence, the request-type-specific blocking probability $p(r_i)$ is unconditioned by the request type probability:

$$p_f \quad = \quad \sum_{0 \le i < n_r} p(r_i) \cdot P(C_t = c(r_i)). \tag{3.10}$$

We will show in Section 3.3.6 that this approach is problematic for performance evaluation purposes as seldom occurring but large request types are hardly considered.

The average capacity blocking probability p_c takes the request sizes into account, i.e., the request-type-specific probability $P(C_t = c(r_i))$ is weighted with

the request size $c(r_i)$:

$$p_c = \sum_{0 \le i < n_r} p(r_i) \cdot \frac{c(r_i) \cdot P(C_t = c(r_i))}{E(C_t)}. \quad (3.11)$$

This calculation leads to more advantageous results (cf. Section 3.3.6) as the blocked traffic volume corresponds to the blocking probability. Nonetheless, the blocking probabilities of small request types still benefit at the expense of the blocking probabilities of the large request types.

Finally, we can take the largest blocking probability of all request types as a common measure:

$$p_m = \max_{0 \le i < n_r} p(r_i). \quad (3.12)$$

AC with Trunk Reservation So far, we have discussed AC only with *complete sharing* (CS) of resources as a blocking strategy, i.e., requests are admitted as long as resources are available. The *trunk reservation* (TR) [212, 213] policy for AC rejects a new flow when the available resources are not sufficient to guarantee QoS for the flows in place *and a new flow of maximum request size* $c_u^{max} = \max_{0 \le i < n_r}\big(c_u(r_i)\big)$. This change of the admission policy influences the distribution of the link occupation. The blocking probability can be computed by an adaptation of Equation (3.7) which is, however, not an exact solution but a sufficiently exact approximation [214, 215, 216, 217]. We substitute Equation (3.7) by

$$\tilde{w}[j] = \begin{cases} 0 & : j < 0, \\ 1 & : j = 0, \\ \frac{1}{j} \cdot \sum_{0 \le i < n_r} \tilde{w}[j - c_u(r_i)] \cdot c_u^{TR}(r_i, C_u) \cdot a(r_i) & : 0 < j \le C_u \end{cases} \quad (3.13)$$

with

$$c_u^{TR}(r_i, C_u) = \begin{cases} c_u(r_i) & : j \le C_u - c_u^{max} + c_u(r_i), \\ 0 & : \text{otherwise} \end{cases}$$

as well as the calculation of the request-type-specific blocking probability (Equation (3.9)) by

$$p(r_i) \quad = \quad \sum_{j=C_u - c_u^{max}+1}^{C_u} w[j]. \tag{3.14}$$

As a consequence, all request types have now the same specific blocking probability (cf. 3.3.6). AC with trunk reservation can lower the required capacity if a maximum request-type-specific blocking probability must be assured.

3.3.3 An Efficient Algorithm for the Calculation of Blocking Probabilities

A straightforward implementation of the above formulae is numerically problematic as the numerous variables $\tilde{w}[j]$ can contain very large values. Therefore, we propose a numerically stable and memory-efficient recipe for the calculation of blocking probabilities. The basic ideas for the numerical stability can be found in 7.4.1 of [218].

The recursions in Equation (3.7) and Equation (3.13) require only a storage of c_u^{max} values. Therefore, we can limit the physical storage for auxiliary variables $\tilde{w}[j]$ by a cyclic array of size $c_u^{max} + 1$. The utility function STORE($\tilde{w}, j, x$) stores value x associated with index position j in the array $\tilde{w}$, GET($\tilde{w}, j$) recalls the value from index position j, and DEVALUATE($\tilde{w}, d$) divides all values in the array by d.

Input: link capacity C_u, request type information

if $C_u \leq 0$ **then**
  for $0 \leq i < n_r$ **do** {set result}
    $p[r_i] := 1$
  end for
else
  $j := 0$ {initialization}
  STORE$(\tilde{w}, 0, 1)$ $\{\tilde{w}[0] := 1\}$
  for $0 < k \leq c_u^{max}$ **do** {initialization}
    STORE$(\tilde{w}, k, 0)$ $\{\tilde{w}[k] := 0\}$
  end for
  $T_{ctrl} := 1$ $\{T_{ctrl} := \sum_{k=0}^{j} \tilde{w}[k]\}$
  for $1 \leq j \leq C_u$ **do** {adapts state weights $\tilde{w}[j]$}
    if $T_{ctrl} > T_{max}$ **then** {scale $\tilde{w}[j]$ down if they become too large}
      DEVALUATE$(\tilde{w}, T_{ctrl})$; $T_{ctrl} := 1$
    end if
    $(\tilde{w}, T_{add})$:=STATEWEIGHTSCS$(j, \tilde{w})$
    $T_{ctrl} := T_{ctrl} + T_{add}$
  end for
  for $0 \leq i < n_r$ **do** {computes $p[r_i]$}
    $p[r_i]$:=BLOCKPROBSCS$(\tilde{w}, T_{ctrl}, C_u, \mathrm{I})$
  end for$p$:=BLOCKINGPROBABILITY$(p[])$
end if

Output: overall blocking probability p

Algorithm 1: BLOCKINGPROBABILITY: computation of the flow blocking probability.

Input: j, $\tilde{w}$, T_{ctrl}, request type information

$x := 0$ {computes $\tilde{w}[j]$ according to Equation (3.7)}
for $0 \leq i < n_r$ **do**
$x := x + \text{GET}(\tilde{w}, j - c_u(r_i)) \cdot c_u(r_i) \cdot a(r_i)$
end for
$x := \frac{x}{j}$; store($\tilde{w}, j, x$) $\{\tilde{w}[j] := x\}$

Output: state weights $\tilde{w}$, weight addition x

Algorithm 2: STATEWEIGHTSCS: computation of the state blocking weights for CS.

Input: j, $\tilde{w}$, request type information

$x := 0$
store($\tilde{w}, j, 0$) $\{\tilde{w}[j] := 0\}$ {updates $\tilde{w}$ according to Equation (3.13)}
for $0 \leq i < n_r$ **do**
$y := j - c_u^{max} + c_u(r_i)$
if $y > 0$ **then**
$tmp := \frac{\text{GET}(\tilde{w}, j - c_u^{max}) \cdot c_u(r_i) \cdot a(r_i)}{y}$
store($\tilde{w}, y, tmp + \text{GET}(\tilde{w}, y)$) $\{\tilde{w}[y] := \tilde{w}[y] + tmp\}$
$x := x + tmp$
end if
end for

Output: state weights $\tilde{w}$, weight addition x

Algorithm 3: STATEWEIGHTSTR: computation of the state blocking weights for trunk reservation.

The state probabilities $w[j]$ have values between 0 and 1 by definition but the values of the auxiliary variables $\tilde{w}[j]$ in Equations (3.7) and (3.13) can become very large and lead to numerical overflow problems. In Algorithm 1 we take advantage of the fact that Equation (3.8) is a fraction for which we can scale down both its counter and denominator without changing its value. To avoid large numbers, downscaling is performed when the control variable T_{ctrl} exceeds a threshold (e.g. $T_{max} = 10^6$).

Input: $\tilde{w}, T_{ctrl}, C_u, i$, request type information

$p[r_i] := 0$
for $C_u - \mathbf{c_u}(\mathbf{r_i}) + 1 \leq j \leq C_u$ **do** {computes $p[r_i]$, cf. Equation (3.9)}
$p[r_i] := p[r_i] + \frac{\text{GET}(\tilde{w},j)}{T_{ctrl}}$
end for

Output: request-type-specific blocking probabilities $p[r_i]$

Algorithm 4: BLOCKPROBSCS: computation of request-type-specific blocking probabilities for CS.

Input: $\tilde{w}, T_{ctrl}, C_u, i$, request type information

$p[r_i] := 0$
for $C_u - \mathbf{c_u^{max}} + 1 \leq j \leq C_u$ **do** {computes $p[r_i]$, cf. Equation (3.14)}
$p[r_i] := p[r_i] + \frac{\text{GET}(\tilde{w},j)}{T_{ctrl}}$
end for

Output: request-type-specific blocking probabilities $p[r_i]$

Algorithm 5: BLOCKPROBSTR: computation of request-type-specific blocking probabilities for trunk reservation.

Algorithm 1 computes the state weights $\tilde{w}[j]$ using Algorithms 2 and 3 according to Equations (3.7) and (3.13) for CS and TR as blocking policy, respectively. Algorithms 2 and 3 are designed such that they can also be applied in Algorithm 6. Then, the request-type-specific blocking probabilities are calculated with means of Algorithms 4 and 5 according to Equations (3.9) and (3.14) also for CS and TR, respectively. Based on these values, the blocking probability p_f, p_c, or p_m can be computed using the function BLOCKINGPROBABILITY according to Equations (3.10) – (3.12), which is not shown here.

3.3.4 An Efficient Algorithm for Capacity Dimensioning

For our framework we require capacity dimensioning which is the inversion of blocking probability calculation. Basically, one can increase the link capacity C_u and check the resulting blocking probability p_{rent}^{cur} until a target blocking probability p_{get}^{tar} is achieved. As this method is numerically expensive, we present Algorithm 6 which is faster for that objective. The key idea for the speedup is the introduction of blocking weights $\tilde{p}[r_i]$, which are auxiliary variables for request-type-specific blocking probabilities $p[r_i]$. The $\tilde{p}[r_i]$ can be adapted while the link capacity j is increased, an the request-type-specific blocking probabilities $p[r_i]$ can be calculated based on them.

Algorithm 6 computes the required capacity. It increases the capacity j until the achieved blocking probability p_{rent}^{cur} is lower than the target blocking probability p_{get}^{tar}. The state weights are again computed using Algorithms 2 and 3 for CS and TR, respectively. The increased state weight is required for the adaptation of the blocking weights $\tilde{p}$. These are updated by Algorithms 7 and 8 also for CS and TR, respectively. The fraction of the blocking weights $\tilde{p}$ and T_{ctrl} yields the request-type-specific blocking probabilities. Like above, BLOCKINGPROBABILITY calculates the overall blocking probability p_{rent}^{cur}, which is used as stop criterion for the capacity increase.

Input: target blocking probability p_{get}^{tar}, request type information

$j := 0$ {initialization}
if $\sum_{0<i\leq n_r} a(r_i) > 0$ **then**
  STORE$(\tilde{w}, 0, 1)$ $\{\tilde{w}[0] := 1\}$
  for $0 < k \leq c_u^{max}$ **do** {initialization}
    STORE$(\tilde{w}, k, 0)$ $\{\tilde{w}[k] := 0\}$
  end for
  for $0 \leq i < n_r$ **do** {initialization}
    $\tilde{p}[r_i] := 1$
  end for
  $p_{rent}^{cur} := 1; T_{ctrl} := 1$ $\{T_{ctrl} := \sum_{k=0}^{j} \tilde{w}[k]\}$
  while $p_{rent}^{cur} > p_{get}^{tar}$ **do** {until blocking probability is small enough}
    if $T_{ctrl} > T_{max}$ **then** {scale down if numbers become too large}
      for $0 \leq i < n_r$ **do**
        $\tilde{p}[r_i] := \frac{\tilde{p}[r_i]}{T_{ctrl}}$
      end for
      DEVALUATE$(\tilde{w}, T_{ctrl})$; $T_{ctrl} := 1$
    end if
    $j := j + 1$
    $(\tilde{w}, T_{add})$:=STATEWEIGHTSCS$(j, \tilde{w})$
    $T_{ctrl} := T_{ctrl} + T_{add}$
    $p_{rent}^{cur} := 0$ {p_{rent}^{cur} is updated}
    for $0 \leq i < n_r$ **do**
      $\tilde{p}[r_i]$:=BLOCKINGWEIGHTSCS$(\tilde{p}[r_i], i, \tilde{w}, j, T_{add})$
    end for
    $p[\,] := \frac{\tilde{p}[\,]}{T_{ctrl}}$
    p_{rent}^{cur} :=BLOCKINGPROBABILITIY$(p[\,])$
  end while
end if

Output: required capacity units j

Algorithm 6: CAPACITYDIMENSIONING: computation of the required capacity.

Input: $\tilde{p}[r_i], i, \tilde{w}, j, T_{add}$, request type information

$\tilde{p}[r_i] := \tilde{p}[r_i] - \text{GET}(\tilde{w}, j - \mathbf{c_u}(\mathbf{r_i})) + T_{add}$ {cf. Equation (3.9)}

Output: request-type-specific blocking weights $\tilde{p}[r_i]$

Algorithm 7: BLOCKINGWEIGHTSCS: computation of request-type-specific blocking weights for CS.

Input: $\tilde{p}, i, \tilde{w}, j, T_{add}$, request type information

$\tilde{p}[r_i] := \tilde{p}[r_i] - \text{GET}(\tilde{w}, j - \mathbf{c_u^{max}}) + T_{add}$ {cf. Equation (3.14)}

Output: request-type-specific blocking weights $\tilde{p}[r_i]$

Algorithm 8: BLOCKINGWEIGHTSTR: computation of request-type-specific blocking weights for trunk reservation.

3.3.5 Further Runtime Optimization

The budget assignment algorithms in Section 3.8 require the successive computation of blocking probabilities of Section 3.3.3, based on increasing link bandwidths. The computation of Algorithm 1 can be significantly accelerated by storing the array of state weights $\tilde{w}$ when a computation has finished, together with the corresponding offered load a, which is hidden in the request type information. If the blocking probability is calculated again based on a larger link bandwidth, the calculation can save many iteration steps by loading the stored values. In addition, the initial iteration parameter j must be set to the previous C_u and T_{ctrl} must be adapted.

3.3.6 Economy of Scale and its Sensitivity

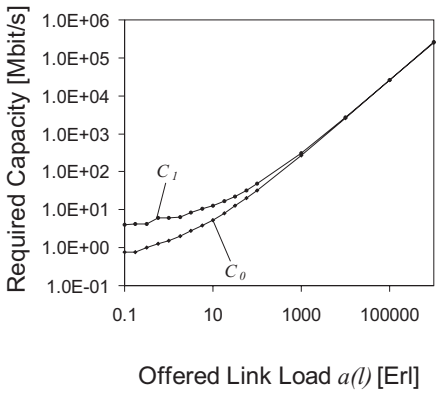
We apply the above formulae for different traffic characteristics under various conditions on a single link and illustrate the phenomenon economy of scale. As economy of scale is the key for understanding NAC performance, we show its sensitivity to different networking parameters.

Impact of Offered Load and Rate Variability

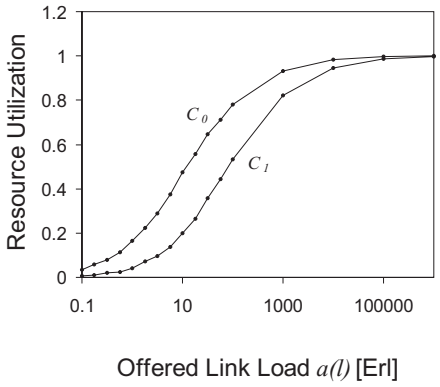
We dimension the required capacity for a single link l for a blocking probability of $p_c = 10^{-3}$ and vary the offered load $a(l)$. We also investigate the impact of the variability of the request rate C_t by setting its interpolation parameter to $t = 0$ and $t = 1$, respectively. Figure 3.8(a) shows the required link capacity $c(l)$ depending on the offered load $a(l)$ while Figure 3.8(b) illustrates the corresponding resource utilization $\rho(l) = \frac{c(l)}{a(l) \cdot E(C_t)}$.

The required link capacity is almost proportional to the offered link load, at least for an offered load of $a(l) = 1000$ Erlang or larger. We use the resource utilization as performance measure for most comparisons because it expresses efficiency in a natural way. The fact that little offered load leads to low utilization and that large offered load leads to high utilization is a non-linear functional dependency and it is called economy of scale or multiplexing gain.

Regarding the request size variability, the resource utilization makes the difference between system alternative C_0 and C_1 more visible than the required capacity. More variability increases the required bandwidth and decreases the resource efficiency but only to a limited extent which vanishes with increasing offered load. In the following investigations, we use rate distribution C_1 as default since we expect the traffic in the future Internet to be more variable than in the telephone network whose 64 Kbit/s connections in the Integrated Services Digital Network (ISDN) correspond rather to C_0.

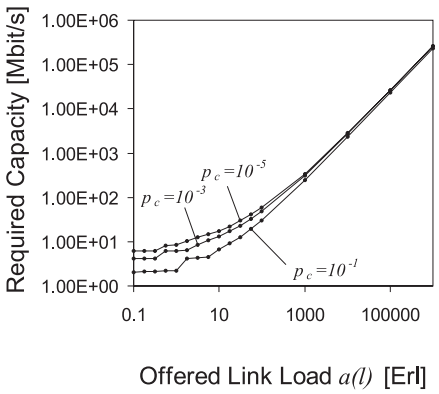


(a) Required capacity.

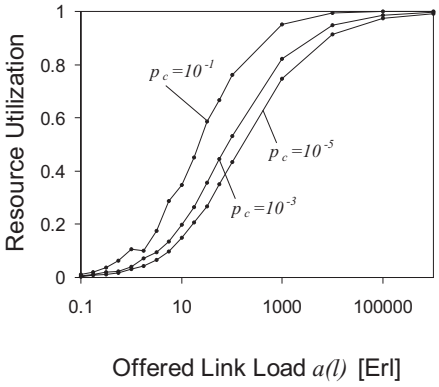


(b) Resource utilization.

Figure 3.8: *Impact of offered load and request rate variability on a single link.*



(a) Required capacity.



(b) Resource utilization.

Figure 3.9: *Impact of offered load and blocking probability on a single link.*

Impact of Blocking Probability

Figure 3.9(b) illustrates the influence of the blocking probability and the offered load on the resource utilization for C_1. We observe economy of scale for all curves but larger blocking probabilities allow for visibly more resource efficiency. However, this influence decreases for high offered load and the resource utilization converges for all blocking probabilities eventually to 100%. Regarding the capacity curves in Figure 3.9(a), the difference among the system alternatives is hardly visible. If not mentioned differently, we use in the following a blocking probability of 10^{-3}.

Impact of Request Rate Variability and Aggregate Blocking Probability Types

We have presented three different methods to compute aggregate blocking probabilities p_f, p_c, and p_m and apply them to the CS admission policy. This distinction has no impact with the TR admission policy because it yields homogeneous blocking probabilities for all request types. We compare the impact of these setting on the request-rate-specific blocking probability, the blocked traffic, the required capacity, and the resulting resource utilization.

Request-Rate-Specific Blocking Probability Figure 3.10 illustrates the request-type-specific blocking probabilities depending on the request rate variability. The link load is set to $a(l) = 100$ *Erl* and the link capacity is dimensioned to such a value that the aggregate blocking probabilities are $p_f = 10^{-3}$, $p_c = 10^{-3}$, and $p_m = 10^{-3}$, respectively.

There are three lines of a certain style. The uppermost corresponds to request type r_0 (2048 *Kbit/s*), the middle to r_1 (256 *Kbit/s*), and the lowermost to r_0 (64 *Kbit/s*). The aggregate blocking probability type p_f yields the highest request-type-specific blocking probabilities, followed by p_c and p_m. In all cases, the request-type-specific blocking probabilities $p(r_i)$ differentiate by about one order of magnitude for a given aggregate blocking probability type p_f, p_c, or p_m.

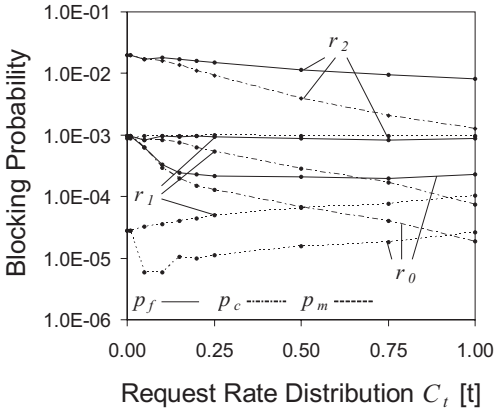


Figure 3.10: *Request-type-specific blocking probabilities for p_f, p_c, and p_m.*

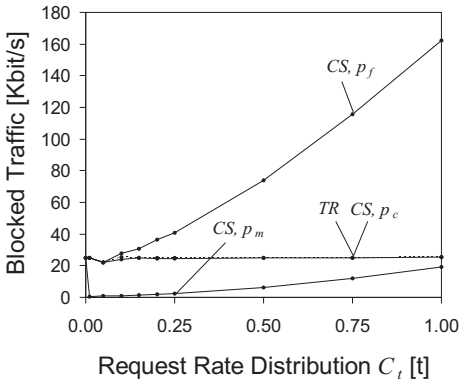
This holds for all request size distributions C_t. This phenomenon can be avoided with TR because due to its construction, all request-type-specific blocking probabilities are exactly 10^{-3}. For the sake of clarity, the corresponding curves are omitted in the figure.

If the average flow blocking probability p_f is used as target aggregate blocking probability for capacity dimensioning, the request-type-specific blocking probabilities are almost independent of the distribution of the request rate C_t. With p_c, all request-type-specific blocking probabilities decrease with increasing request size variability. For p_m, the request-type-specific blocking probability for r_2 is exactly 10^{-3} but the blocking probabilities for r_0 and r_1 are significantly smaller.

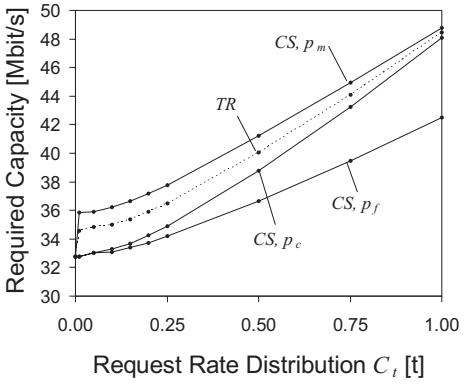
Blocked Traffic Figure 3.11(a) shows the amount of blocked traffic in *Kbit/s*. For p_c with CS and TR, the amount of blocked traffic is constant at $10^{-3} \cdot a(l) \cdot E(C_t)$ which is due to the construction of these mechanisms. The proportion of flows with a very large request size grows with increasing request rate variability. Since they encounter a larger blocking probability for p_f with CS, the amount of blocked traffic is tremendously enlarged for that option. With p_m and CS, the request type with the largest rate has a blocking probability of 10^{-3} and the others have a smaller one. Therefore, the blocked traffic is smaller than $10^{-3} \cdot a(l) \cdot E(C_t)$ in any case, and the blocked traffic increases also with an increasing portion of r_2 in the request rate distribution C_t.

Required Capacity Figure 3.11(b) shows the required link capacity for a target blocking probability of 10^{-3}. The required capacity grows with increasing request rate variability for all dimensioning examples. The methods p_f and p_c with CS need the least capacity but do not provide a maximum blocking probability of 10^{-3} for all request types. Admission Control with TR reaches that goal in a most economic way as it needs less capacity than p_m with CS. If the link is dimensioned for C_1, the resource requirements are about the same for p_c and p_m with CS and TR. Figure 3.12(a) illustrates that small blocking probabilities like 10^{-1} can lead to decreasing link bandwidth requirements for p_f with CS because a large amount of traffic in terms of *Kbit/s* may be lost. This is a counterintuitive result which is only obtained for p_f with CS. Therefore, dimensioning networks for that resource allocation method is not suitable for performance investigation.

Resource Utilization The resource utilization for a link dimensioned for a blocking probability of 10^{-3} is depicted in Figure 3.12(b). It looks symmetric to Figure 3.11(b) because it shows the transported traffic divided by the required capacity. Due to the small blocking probability, the blocked traffic is negligible such that the resource utilization is about indirectly proportional to the required capacity.

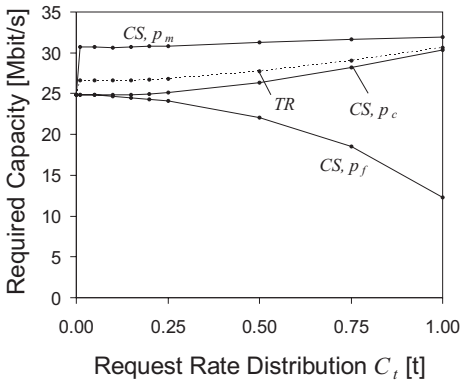


(a) Blocked traffic rate ($p = 10^{-3}$).

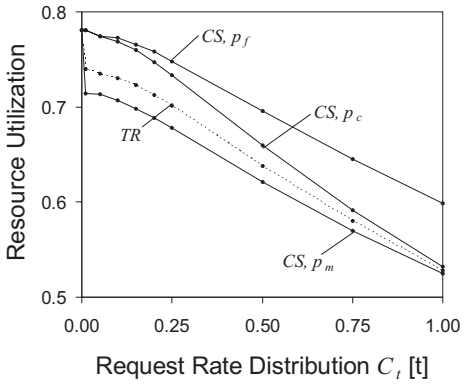


(b) Required capacity ($p = 10^{-3}$).

Figure 3.11: *Impact of request rate variability and blocking probability.*



(a) Required capacity ($p = 10^{-1}$).



(b) Resource utilization ($p = 10^{-3}$).

Figure 3.12: *Impact of request rate variability and blocking probability.*

Summary

We have illustrated the impact of various traffic and system parameter on the resource requirements on a single link to meet a certain blocking probability for a given offered traffic load. The required capacity and the achievable resource utilization depends strongly on the offered load. The target blocking probability and the request rate variability have a minor but also significant influence on the results. Although, TR is the most economic method to achieve a certain blocking probability for all request types, we do not apply it in our study. Usually, the largest request size is not known in advance and, therefore, TR is not implemented in practice, so we take CS as blocking policy. We choose p_c as aggregate blocking criterion because it allows an easy computation of the blocked traffic which is required for efficiency considerations. In our following NAC performance investigation, we will only modify the offered load because it has the major impact on economy of scale. We take C_1 since it reveals the largest request rate variability and we take $p_c = 10^{-3}$ as dimensioning target because similar values are used in the telephone system.

3.4 BNAC-Specific Capacity Dimensioning for Networks

In this section we derive formulae for the calculation of the required network capacity depending on the NAC method. We explain the basic approach and apply it to each NAC type. Finally, we define a performance measure for the comparison of NAC methods.

3.4.1 General Approach

To evaluate the required network capacity, we first determine a maximum blocking probability $p(b)$ for each budget b, then we determine the required budget capacity $c(b)$ based on its offered load $a(b)$ and $p(b)$. Finally, we use these budget capacities as constraints for a worst case analysis regarding the admitted traffic for each link l, which yields the minimum required link capacity $c(l)$.

Calculation of the Required Budget Blocking Probability

We consider a flow f that wants to pass the NAC of a network. The set $\mathcal{B}_g$ contains all budgets that need to be checked if the available capacity is sufficient whenever a flow of the b2b traffic aggregate g asks for admission. The flow blocking probability at an individual budget b is denoted by $p(b)$. Assuming that blocking at different budgets is rather positively correlated, an upper bound for the b2b flow blocking probability is given by

$$p(f) \quad = \quad 1 - \prod_{b \in \mathcal{B}(g)} (1 - p(b)). \tag{3.15}$$

We further simplify the model by postulating the same blocking probabilities p_b for all budgets involved which allows us to calculate p_b for a given target b2b flow blocking probability p_{b2b} by

$$p_b \quad = \quad 1 - \sqrt[m]{1 - p_{b2b}}. \tag{3.16}$$

with $m = |\mathcal{B}(g)|$.

Calculation of the Required Budget Capacity

We denote the offered load for a b2b traffic aggregate $g_{v,w}$ by $a(g_{v,w})$. The resulting matrix $A_{\mathcal{G}} = \left(a(g_{v,w})\right)_{v,w \in \mathcal{V}}$ is the traffic matrix. The offered load imposed by flows using a certain budget b depends on the NAC scheme and is denoted by $a(b)$. Like in the previous section, the required capacity $c(b)$ can be calculated based on $a(b)$ and $p(b)$.

Calculation of the Required Link Capacity

The admitted rate of an aggregate $g_{v,w}$ is given by $c(g_{v,w})$ and the matrix $C_{\mathcal{G}} = \left(c(g_{v,w})\right)_{v,w \in \mathcal{V}}$ describes the network-wide admitted traffic pattern. A possible traffic pattern $C_{\mathcal{G}} \in \mathbb{R}_0^{+\,|\mathcal{V}|^2}$ obeys the following formulae

$$\forall v, w \in \mathcal{V} \quad : \quad c(g_{v,w}) \geq 0 \tag{3.17}$$

$$\forall v \in \mathcal{V} \quad : \quad c(g_{v,v}) = 0. \tag{3.18}$$

If BNAC is applied to the network, the traffic patterns must in addition satisfy the constraints imposed by the NAC budgets. To determine the minimum required capacity $c(l)$ of each link $l \in \mathcal{E}$, we conduct worst case analyses. The mentioned linear equations serve as side conditions in the following rate maximization.

$$c(l) \quad \geq \quad \max_{C_{\mathcal{G}} \in \mathbb{R}_0^{+\,|\mathcal{V}|^2}} \sum_{g \in \mathcal{G}} c(g) \cdot u_l(g). \tag{3.19}$$

Since the aggregate rates have real values, the maximization can be performed by the Simplex algorithm [219] in polynomial time.

3.4.2 NAC-Specific Capacity Dimensioning

We adapt the above general link dimensioning approach to each NAC method individually. This yields the benefit that the rate maximization can be mostly

performed using quite trivial and efficient equations such that the time consuming Simplex algorithm can be bypassed.

LB NAC

The LB NAC requires transit flows to check a budget LB_l for every link l of their paths for admission. Therefore, the chosen path for a flow influences the cardinality of $\mathcal{B}(g)$. As a result, we get different required budget blocking probabilities $p(b)$ for the same budget b depending on the considered flows and the paths taken. In [220], we have investigated three different options to handle this problem and they yield the same results for practical networking scenarios. Thus, to calculate the required budget blocking probability, we take the maximum number of budgets $m(LB_l)$ of all budget sets $\mathcal{B}(g)$ that contain the budget LB_l. This number can be computed by

$$m(LB_l) \quad = \quad \max_{\{g \in \mathcal{G} : u_l(g) > 0\}} len^{max}_{path}(g, l) \tag{3.20}$$

whereby $len^{max}_{path}(g, l)$ is the maximum length of a path containing l used by g. Thus, $p(LB_l)$ can be determined. As the budget LB_l covers all flows traversing link l, its expected offered load is

$$a(LB_l) \quad = \quad \sum_{g \in \mathcal{G}} a(g) \cdot u_l(g). \tag{3.21}$$

This allows for the computation of $c(LB_l)$. Equation (3.1) yields the linear equation

$$\forall l \in \mathcal{E} : \sum_{g \in \mathcal{G}} c(g) \cdot u_l(g) \leq c(LB_l) \tag{3.22}$$

that must be respected by each traffic pattern, so the minimum required capacity $c(l)$ of link l is constrained by

$$c(l) \quad \geq \quad c(LB_l). \tag{3.23}$$

IB/EB NAC

With the IB/EB NAC, a flow is admitted by checking both the respective ingress and the egress budget. Thus, we get $m(IB_v) = m(EB_w) = 2$. The IB/EB NAC decides about all flows with the same ingress router v using IB_v and about flows with the same egress router w using EB_w. The offered load of the corresponding budgets is

$$\begin{aligned} a(IB_v) &= \sum_{w \in \mathcal{V}} a(g_{v,w}), \text{ and} \\ a(EB_w) &= \sum_{v \in \mathcal{V}} a(g_{v,w}). \end{aligned} \tag{3.24}$$

Here we use the inequalities from Equation (3.2) and Equation (3.3) as side conditions in the Simplex method for the computation of the capacity $c(l)$:

$$\forall v \in \mathcal{V} \quad : \quad \sum_{w \in \mathcal{V}} c(g_{v,w}) \leq c(IB_v), \text{ and} \tag{3.25}$$

$$\forall w \in \mathcal{V} \quad : \quad \sum_{v \in \mathcal{V}} c(g_{v,w}) \leq c(EB_w). \tag{3.26}$$

In case of the mere IB NAC we have $m(IB_v) = 1$. The IBs are computed in the same way like above, however, there is a computational shortcut to the Simplex method for the calculation of the required link capacity $c(l)$:

$$c(l) \quad \geq \quad \sum_{v \in \mathcal{V}} c(IB_v) \cdot \sum_{w \in \mathcal{V}} u(l, g_{v,w}). \tag{3.27}$$

BBB NAC

With the BBB NAC, only one budget is checked, therefore, we have $m(BBB_{v,w}) = 1$. The BBB NAC decides about all flows with ingress router v and egress router w using $BBB_{v,w}$. The offered load for $BBB_{v,w}$ is simply

$$a(BBB_{v,w}) \quad = \quad a(g_{v,w}). \tag{3.28}$$

Since Equation (3.4) is checked for admission,

$$\forall v, w \in \mathcal{V} : c(g_{v,w}) \leq c(BBB_{v,w}) \tag{3.29}$$

must be fulfilled and the minimum capacity $c(l)$ of link l is constrained by

$$c(l) \quad \geq \quad \sum_{v,w \in \mathcal{V}} c(BBB_{v,w}) \cdot u(l, g_{v,w}). \tag{3.30}$$

ILB/ELB NAC

The ILB/ELB NAC requires that a transit flow needs to ask for admission for every link as with the LB NAC. Therefore, we set

$$\begin{aligned} m(ILB_{l,v}) &= 2 \cdot \max_{\{w \in \mathcal{V} : u(l, g_{v,w}) > 0\}} len^{max}_{path}(g_{v,w}, l), \quad \text{and} \\ m(ELB_{l,w}) &= 2 \cdot \max_{\{v \in \mathcal{V} : u(l, g_{v,w}) > 0\}} len^{max}_{path}(g_{v,w}, l). \end{aligned} \tag{3.31}$$

The ILB/ELB NAC decides about flows with the same ingress router v on the link l using the $ILB_{l,v}$ and about all flows with the same egress router w on the link l using $ELB_{l,w}$. The offered load for the budgets is

$$\begin{aligned} a(ILB_{l,v}) &= \sum_{w \in \mathcal{V}} a(g_{v,w}) \cdot u(l, g_{v,w}), \text{ and} \\ a(ELB_{l,w}) &= \sum_{v \in \mathcal{V}} a(g_{v,w}) \cdot u(l, g_{v,w}). \end{aligned} \tag{3.32}$$

Due to Equation (3.5) and Equation (3.6), the side conditions

$$\forall v \in \mathcal{V} \quad : \quad \sum_{w \in \mathcal{V}} c(g_{v,w}) \cdot u(l, g_{v,w}) \leq c(ILB_{l,v}), \text{ and} \tag{3.33}$$

$$\forall w \in \mathcal{V} \quad : \quad \sum_{v \in \mathcal{V}} c(g_{v,w}) \cdot u(l, g_{v,w}) \leq c(ELB_{l,w}) \tag{3.34}$$

must be respected for the computation of the link capacities in Equation (3.19). In case of the mere ILB NAC, another shortcut can be applied to calculate the

required link capacity:

$$m(ILB_{l,v}) = \max_{\{w \in \mathcal{V}: u(l,g_{v,w})>0\}} len^{max}_{path}(g_{v,w}, l) \text{ and} \quad (3.35)$$

$$c(l) \geq \sum_{v \in \mathcal{V}} c(ILB_{l,v}) \quad (3.36)$$

After all, we can classify LBs, ILBs, and ELBs as path-aware budgets because they consider the routing for the calculation of their capacity. In contrast, the capacity of IBs, EBs, and BBBs is independent of the routing and we call them path-unaware.

3.4.3 Performance Measure for NAC Comparison

We compute the required link capacities for all NAC methods according to the equations above. The required network capacity $c(\mathcal{N})$ is the sum of all link capacities in the network. The overall transmitted traffic rate $\hat{c}(\mathcal{N})$ is the sum of the offered load of all b2b aggregates g weighted by their average path lengths $len^{avg}_{path}(g)$, their acceptance probability $(1-p_{b2b})$, and the mean request rate $E(C_t)$. We can neglect the fact that requests with a larger rate have a higher blocking probability due to the construction in Equation (3.11).

$$c(\mathcal{N}) = \sum_{l \in \mathcal{E}} c(l) \quad (3.37)$$

$$\hat{c}(\mathcal{N}) = (1-p_{b2b}) \cdot E(C_t) \cdot \sum_{\{g \in \mathcal{G}\}} a(g) \cdot len^{avg}_{path}(g) \quad (3.38)$$

$$\rho(\mathcal{N}) = \frac{\hat{c}(\mathcal{N})}{c(\mathcal{N})} \quad (3.39)$$

The overall resource utilization $\rho(\mathcal{N})$ is the fraction of the transmitted traffic rate and the overall network capacity. We use it in the next section as the performance measure for the comparison of NAC methods.

3.5 Performance Evaluation Framework for BNAC Methods

First, we present possible design options for NAC performance studies and discuss their advantages and shortcomings. Then, we present the test topologies and the traffic matrices that are mostly used in this work.

3.5.1 Design Options for NAC Performance Evaluation

The objective of a QoS network provider is the satisfaction of his customers at minimum expenses. However, in case of capacity shortage, the flow blocking probability is large due to NAC. As this also dissatisfies the user, enough capacity must be provided to cover the average transmission demand. This is characterized by an average load a and a corresponding request size distribution. As the required link capacities are either capital or operational expenses for the network provider, the potential resource utilization by NAC should be as large as possible to achieve best customer treatment at least cost.

There are various possibilities for the performance evaluation of NAC approaches that come from the relation among the system parameters offered load, blocking probability, and network capacity. Two of them condition the third term. We list the possible experiment designs in Table 3.2 and discuss them in the following.

Design Option 0

In design option 0 the network with all its link capacities is given and a given b2b blocking probability p_{b2b} must be met for all traffic aggregates. The offered load in the b2b traffic matrix is the variable parameter being part of the traffic model which determines, e.g., the average path length weighted by the offered

Table 3.2: *Design options for NAC performance investigation.*

influencing term	design option 0	design option 1	design option 2
offered load per b2b agg. $a(g_{v,w})$	variable	given	given
blocking prob. per b2b agg. $p(g_{v,w})$	given	variable	given
link capacities $c(l)$	given	given	variable

load $a(g_{v,w})$ of individual b2b aggregates. Since the traffic matrix has many degrees of freedom, its assignment is difficult. Furthermore, the structure of the traffic matrix influences the potential economy of scale that can be achieved by different NAC methods, and the achievable resource utilization. For these reasons, this design option leads to many difficulties and to an unfair NAC comparison. Apart from that, the offered load in real networks must be taken as it is. It cannot be chosen to convene the network properties to achieve a low flow blocking probability.

Design Option 1

Design option 1 provides the network with all its link capacities and the traffic matrix which determines the individual aggregate b2b blocking probabilities $p(g_{v,w})$. This can be observed in operational networks. However, appropriate settings for the fixed parameters are required to achieve reasonable b2b blocking probabilities, which complicates the investigation. Furthermore, an "appropriate" setting depends on the NAC mechanism itself such that the comparability of different NAC methods is not guaranteed by this design option, either. If different b2b aggregates experience different blocking probabilities, the comparison of different NAC methods becomes even more difficult. If a common minimum blocking probability must be found for all b2b relationships, some link capacities might be partly unused. Hence, there are many obstacles complicating the use of

this design option.

Design Option 2

In design option 2 the traffic matrix is given and the link capacities are determined to meet a required b2b aggregate blocking probability. With this approach, the above mentioned problems do not exist. Therefore, we use it as our methodology for NAC performance comparison. Figure 3.13 reviews the required calculation steps and the parameter flow.

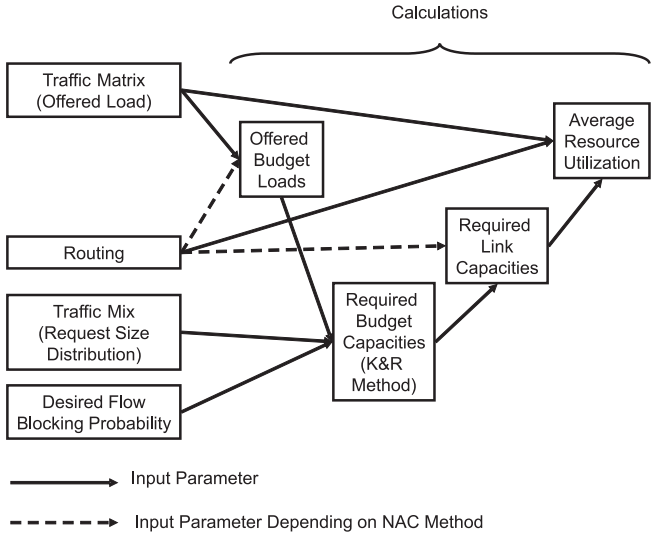


Figure 3.13: *Calculation steps in the NAC performance evaluation framework.*

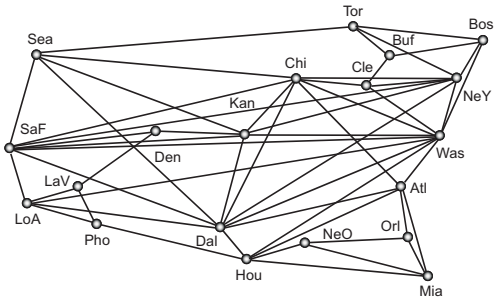
3.5.2 Networking Scenarios

We briefly illustrate the network topologies, the traffic matrices, and the routing strategy for our studies.

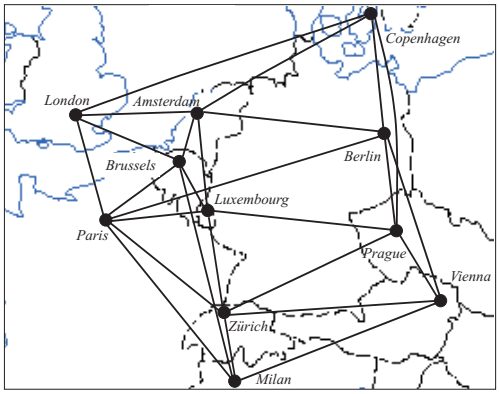
Test Networks

We want to evaluate the performance of NAC methods in the context of carrier grade networks. Therefore, we focus on core network structures by means of two topologies taken from operational networks. The network structure is described by graph notation, i.e., the topology is given by $\mathcal{N} = (\mathcal{V}, \mathcal{E})$ where the set of vertices $\mathcal{V}$ contains all routers and the set of edges $\mathcal{E}$ contains all uni-directional links. The number of unidirectional edges leaving a node v is called node degree $deg(v)$. The average node degree can be computed by $deg_{avg} = \frac{2 \cdot |\mathcal{E}|}{|\mathcal{V}|}$.

The Lab03 network in Figure 3.14(a) is taken from the testbed of the KING project [2]. It is a modification of the UUNET in 1994 where all nodes with a node degree of at most 2 are successively removed. The network in Figure 3.14(b) is the optical core of the infrastructure in the COST-279 project [221]. The project was part of the "European Co-operation in the Field of Scientific and Technical Research" and concentrated on ultra-high capacity optical transmission networks. We use both networks in our performance evaluation because they have different properties which are summarized in Table 3.3. Depending on the experiment, the effects can be better observed in the COST-239 or the Lab03 network. Both networks have such a topology that an alternative path exists for any b2b relationship in case of a single link or node failure. This is a prerequisite for resilient networking in general.



(a) Lab03 network.



(b) COST-239 core network.

Figure 3.14: *Network topologies for investigation of BNAC methods.*

Table 3.3: *Properties of the test networks.*

network properties	COST239	Lab03
number of nodes $\|\mathcal{V}\|$	11	20
number of links $\|\mathcal{V}\|$	52	106
minimum node degree deg_{min}	4	3
average node degree deg_{avg}	4.73	5.30
maximum node degree deg_{max}	6	10
coefficient of variation $c_{var}(\deg)$	0.17	0.29
parallel paths per b2b relation	4.38	3.50

Traffic Matrices

The average offered b2b load is given by parameter a_{b2b} and the overall offered load in the network is

$$a_{tot} = \sum_{g \in \mathcal{G}} a(g) = |\mathcal{V}| \cdot (|\mathcal{V}| - 1) \cdot a_{b2b}. \tag{3.40}$$

We use the average b2b load to scale the overall offered load a_{tot} and create a traffic matrix proportional to the populations $\pi(v)$ associated with the respective nodes v. The calculation is based on the populations given in Tables 3.4 and 3.5 and the following equation:

$$a(g_{v,w}) = \begin{cases} \frac{a_{tot} \cdot \pi(v) \cdot \pi(w)}{\sum_{x,y \in \mathcal{V}, x \neq y} \pi(x) \cdot \pi(y)} & \text{for } v \neq w, \\ 0 & \text{for } v = w. \end{cases} \tag{3.41}$$

Routing

We apply shortest path routing in the NAC investigation because it is the basis for most Interior Gateway Protocols (IGPs). We consider the options single- and multi-paths routing. If several equal cost paths exist from source to destination,

an arbitrary one is chosen once for all flows with single-path routing while for multi-path routing the traffic is distributed equally to all outgoing interfaces that have the same distance towards the destination. Both alternatives can be signalled using Open Shortest Path First (OSPF) [39] as routing protocol within an autonomous system. We use single shortest path routing as default if not mentioned differently.

Table 3.4: *Population of the cities and their surroundings for the Lab03 network.*

$name(v)$	$\pi(v)$ $[10^3]$	$name(v)$	$\pi(v)$ $[10^3]$
Atlanta	4112	Los Angeles	9519
Boston	3407	Miami	2253
Buffalo	1170	New Orleans	1338
Chicago	8273	New York	9314
Cleveland	2250	Orlando	1645
Dallas	3519	Phoenix	3252
Denver	2109	San Francisco	1731
Houston	4177	Seattle	2414
Kansas	1776	Toronto	4680
Las Vegas	1536	Washington	4923

Table 3.5: *Population of the respective countries for the COST-239 network.*

$name(v)$	$\pi(v)$ $[10^3]$	$name(v)$	$\pi(v)$ $[10^3]$
Amsterdam (NL)	16101	Paris (F)	59343
Berlin (D)	82360	Prague (CZ)	10300
Bruxelles (B)	10292	Rome (I)	58018
Copenhagen (DK)	5363	Vienna (A)	8141
London (UK)	60075	Zurich (CH)	7261
Luxembourg (L)	447		

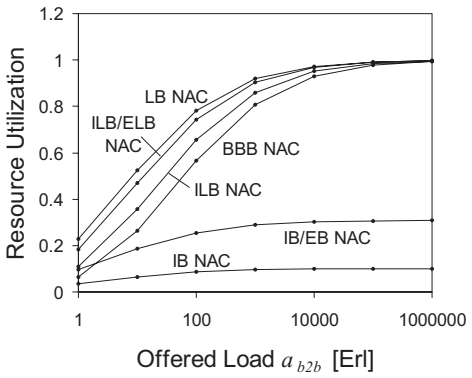
3.6 Performance Comparison of BNAC Methods

In this section, we take the resource utilization $\rho(\mathcal{N})$ as performance measure to compare different BNAC methods. We compare the performance of all basic NAC schemes depending on the offered load, the traffic matrix, the network topology, and the routing. The analysis of the experiments leads to a profound understanding of NAC performance. In the following experiments, the capacity for the example networks is dimensioned to meet a desired b2b flow blocking probability of $p_{b2b} = 10^{-3}$ in the presence of a given traffic matrix.

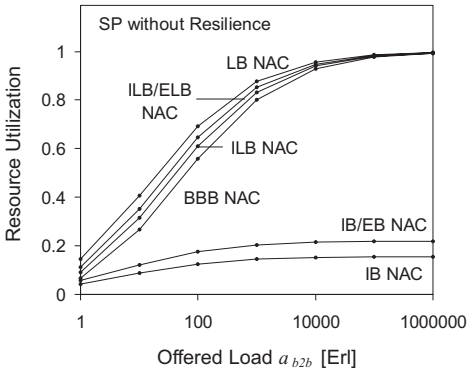
3.6.1 Influence of the Offered Load

Figures 3.15(a) and 3.15(b) show the resource utilization depending on the offered load a_{b2b} for all NAC methods in the Lab03 and in the COST-239 network. We observe the typical increase of the resource utilization with the offered b2b load a_{b2b} which is known as economy of scale. The differences among the NAC types result from their different ability to exploit it. The LB, ILB/ELB, ILB, and BBB NAC can achieve 100% resource utilization in the limit. The IB/EB NAC has a better performance than the IB NAC but they are both inefficient as their curves converge to network-topology-specific asymptotes of 16% and 22%.

The link budgets cover the largest amount of traffic (cf. Equation (3.21)), followed by ingress and egress link budgets (cf. Equations (3.32) and (3.32)), and by b2b budgets (cf. Equation (3.28)). A reduced traffic load per budget leads to a lower multiplexing gain and to a higher required overall network capacity $c(\mathcal{N})$. This explains the order of efficiency for the LB, ILB, and BBB NAC. For a sufficient offered load, the utilization of these NAC methods approaches 100% in the limit. Since the LB NAC induces states in the core or other complex mechanisms, our new ILB/ELB NAC methods is the most resource-efficient truly stateless-core NAC approach.



(a) Lab03 network.



(b) COST-239 network.

Figure 3.15: *The impact of the offered load on the resource utilization in the COST-239 network.*

The IB NAC is not economical. An ingress budget allocates its full capacity $c(IB_v)$ on the paths from v to all possible destinations in a network (cf. Equation (3.27)). If a b2b aggregate $g_{v,w}$ has the maximum capacity, i.e. $c(g_{v,w}) = c(IB_v)$, its traffic is carried only on the path from v to w. This leads to a low utilization because the allocated resources for all other paths from v to $x \in \mathcal{V} \setminus \{v, w\}$ cannot be fully used.

The reason for the abundant bandwidth provisioning is that enough capacity must be available for all traffic patterns that can be admitted by a NAC entity. The IB NAC is not sufficiently restrictive. Applying additional egress budgets excludes most unlikely traffic patterns, e.g., the scenario that all traffic from all ingress routers streams to the same egress router. Therefore, the IB/EB NAC limits the inefficiency to a certain extent but it does not solve the basic problem that not all allocated resources can be utilized simultaneously. The ILB/ELB NAC improves the performance of the ILB NAC in the same way.

Applying additional egress budgets excludes most unlikely traffic patterns, e.g., the scenario that all traffic from all ingress routers streams to the same egress router. Therefore, the IB/EB NAC limits the inefficiency to a certain extent but it does not solve the basic problem that not all allocated resources can be utilized simultaneously. The ILB/ELB NAC improves the performance of the ILB NAC in the same way.

Although both networks have substantially different topologies, the results look qualitatively similar. The quantitative differences result from a different traffic concentration on the links due to different network size and path length. Because of that, the BNAC methods based on path-aware budgets (LB, ILB/ELB, and ILB NAC) can unfold their strength in achieving more multiplexing gain only to a limited degree. This is also addressed in Section 3.6.4.

3.6.2 Influence of the Traffic Matrix

The offered load has a major impact on the resource efficiency [222]. Therefore, we also investigate the influence of its distribution over the network on the uti-

lization. We keep the overall load a_{tot} constant and distort the structure of the traffic matrix. We compute it still based on a_{tot} and the node populations π according to Equation (3.41) but we modify π using an exponential extrapolation with parameter t:

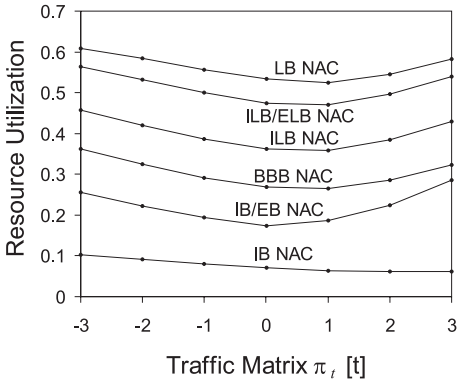
$$\pi_t(v) \quad = \quad |\mathcal{V}| \cdot \overline{\pi} \cdot \frac{\exp(\delta(v) \cdot t)}{\sum_{v \in \mathcal{V}} \exp(\delta(v) \cdot t)}, \tag{3.42}$$

with $\overline{\pi} = \sum_{v \in \mathcal{V}} \pi(v)$. The resulting traffic matrix is denoted by π_t. The value $\delta(v)$ is determined by $\pi_1(v) = \pi(v)$, i.e. $\delta(v) = \ln(\frac{\pi(v)}{\overline{\pi}})$. According to that construction, the original traffic matrix π and traffic matrix π_1 are equal.

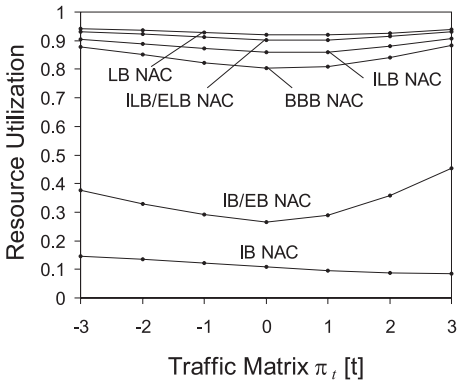
Table 3.6: *Properties of extrapolated city sizes in the Lab03 network.*

t	-3	-2	-1	0	1	2	3
$c_{var}[\pi_t(v)]$	7.88	2.62	0.78	0	0.69	2.02	5.29
len_{path}^{avg}	2.91	2.68	2.43	2.15	1.91	1.77	1.72

Table 3.6 describes the effect of the extrapolation on the city sizes $\pi_t(v)$. All city sizes are equal for $t = 0$. As a consequence, all b2b aggregates carry the same offered load. If a city is larger than the average city size ($\pi(v) > \overline{\pi}$), it is scaled down by a negative value of t and it is scaled up for a positive value of t. With increasing $|t|$, the number of cities below the average size increases and the number of cities above the average size decreases. Therefore, the coefficient of variation of the city sizes increases. As a consequence, most of the traffic flows among fewer cities, which impacts the coefficient of variation of the entries in the traffic matrix. We consider the average path length (len_{path}^{avg}) weighted by the corresponding offered load. Large cities (for $t = 1$) are usually connected closer among each other than smaller cities. If they grow in size, the hop distance among them dominates the average path length. Thus, the average path length decreases with increasing t.

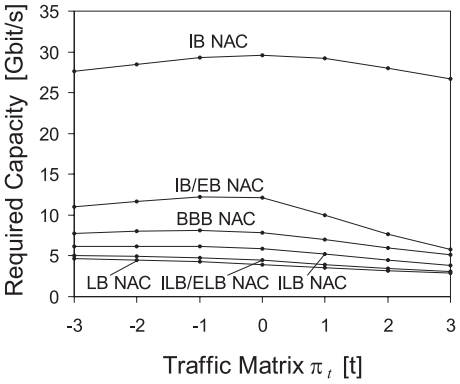


(a) $a_{b2b} = 10$ *Erl.*

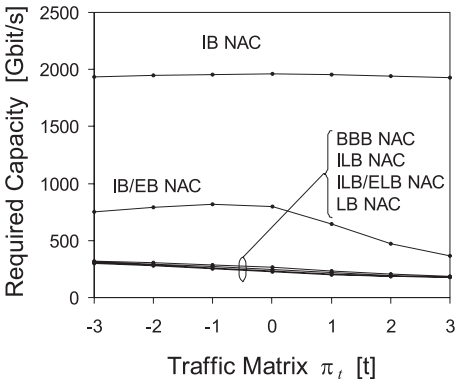


(b) $a_{b2b} = 1000$ *Erl.*

Figure 3.16: *Impact of traffic matrix variability on the resource utilization in the Lab03 network.*



(a) $a_{b2b} = 10$ *Erl.*



(b) $a_{b2b} = 1000$ *Erl.*

Figure 3.17: *Impact of traffic matrix variability on the required capacity in the Lab03 network.*

Figures 3.16(a) and 3.16(b) show the results of our experiments where we vary the extrapolation parameter and set the scaling factor for the offered load to $a_{b2b} = 10$ *Erl* and $a_{b2b} = 1000$ *Erl*, respectively. In Figure 3.16(a) we observe that the resource utilization increases for large absolute values of $|t|$. This is due to the fact that the paths between the large cities carry the major traffic portion and can utilize the bandwidth more efficiently than links with a medium-sized offered load in case of $t = 0$.

For the LB, ILB, ILB/ELB, and BBB NAC, the influence of the extrapolation parameter t depends on the offered load. It is reduced for $a_{b2b} = 1000$ *Erl* and vanishes entirely for very large a_{b2b} as the utilization tends towards 100% in all cases. Figures 3.17(a) and 3.17(b) show the required capacity. The curves for the LB, ILB, ILB/ELB, and BBB NAC are clearly correlated with the average path length. For $a_{b2b} = 1000$ *Erl* this phenomenon is better visible than for $a_{b2b} = 1000$ *Erl* because large offered load eliminates the effect of multiplexing gain for large $|t|$.

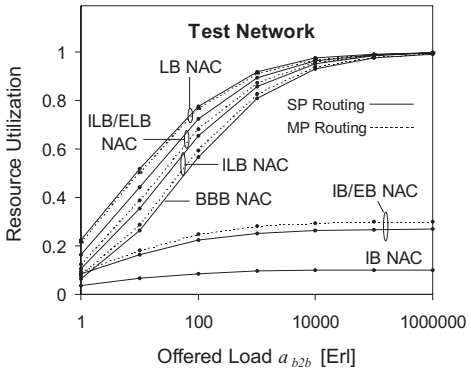
The IB NAC requires the allocation of $c(IB_v)$ bandwidth from v to all other nodes $w \in \mathcal{V}$. As all routers are also egress routers in our experiment, this capacity is reserved along a source tree with $|\mathcal{V}|-1$ links. Thus, the resource demand is roughly $\sum_{v \in \mathcal{V}} c(IB_v) \cdot (|\mathcal{V}|-1) = a_{tot} \cdot (|\mathcal{V}|-1) \cdot E(C_t)$, which is independent of the traffic matrix and the network structure. Figure 3.17(b) illustrates this very well while in Figure 3.17(a) the capacity savings due to the economy of scale are visible for large $|t|$. Since the network resources are about constant, the resource utilization is proportional to the average path length, i.e. $\rho(\mathcal{N}) = \frac{len_{path}^{avg}}{|\mathcal{V}|-1}$, which can be well observed by the limit for the IB NAC in Figures 3.15(a) and 3.15(b). Comparing Figures 3.16(a) and 3.16(b) with Table 3.6 shows that both the resource utilization and the average path length len_{path}^{avg} decrease with increasing t.

Additional egress budgets limit the variety of possible traffic patterns by $\sum_{v \in \mathcal{V}} c(g_{v,w}) \leq c(EB_w)$ and $\sum_{w \in \mathcal{V}} c(g_{v,w}) \leq c(IB_v)$. Compared to IB NAC, the achieved worst case scenarios with lower maximum rates for individual links in Equation (3.19). In Figures 3.17(a) and 3.17(a), this yields significantly reduced capacity requirements for the network and increases the resource utiliza-

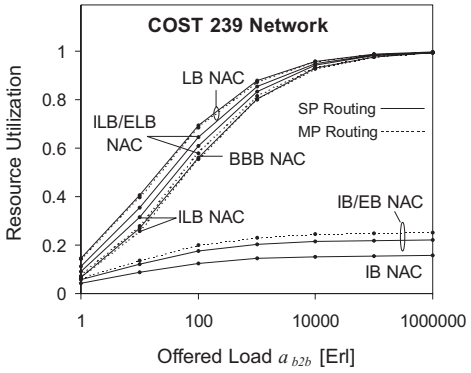
tion by a factor of 3. Moreover, the reduced network capacity for increased $|t|$ can be motivated by the following example. We consider a network with a homogeneous traffic matrix, i.e. $t = 0$. All ingress and egress budgets have the same capacity of $\frac{c^*}{|\mathcal{V}|}$. Then, the sum of all maximum b2b traffic aggregates is $(|\mathcal{V}|-1) \cdot c^*$. Now, let us consider an exponential extrapolation. We assume that $\frac{1}{3} \cdot |\mathcal{V}|$ nodes have ingress and egress budgets of $\frac{2 \cdot c^*}{|\mathcal{V}|}$ and that $\frac{2}{3} \cdot |\mathcal{V}|$ nodes have ingress and egress budgets of $\frac{c^*}{2 \cdot |\mathcal{V}|}$. The b2b traffic aggregate rates are limited to $c(g_{v,w}) \leq \min\big(c(IB_v), c(EB_w)\big)$. Therefore, the sum of all maximum b2b traffic aggregates is only $(\frac{2}{3} \cdot |\mathcal{V}|-1) \cdot c^*$. This motivates that heterogeneous traffic matrices reduce the required network capacity for IB/EB NAC. In Figures 3.17(a) and 3.17(b) the effect is not symmetric in t because the capacity reduction is superposed with increasing path lengths for increasing t.

3.6.3 Influence of the Routing

The traffic matrix influences the traffic distribution in the network. It can be also modified by different routing approaches. Shortest path routing with the single-path (SP) option is mostly applied for IGP routing, i.e., the data are transported on one shortest path from source to destination. A fundamental paradigm shift is multi-path (MP) routing because this decreases the size of b2b aggregates on specific links. For our analysis we choose Equal Cost Multi-Path (ECMP) routing which is an option in OSPF [39]. With ECMP, the router forwards the data equally over all outgoing interfaces that lead to the corresponding destination at minimum cost. Figures 3.18(a) and 3.18(b) show the influence of SP and MP routing on the NAC performance in the Lab03 and the COST-239 network based on the realistic traffic matrices ($t = 1$). The achievable resource utilization is illustrated by solid lines for SP routing and by dashed lines for MP routing. We discuss these results for each NAC type separately.



(a) Lab03 network.



(b) COST-239 network.

Figure 3.18: *Impact of SP and MP routing on the resource efficiency.*

There is only one solid line each for the IB NAC and the BBB NAC because the curves for SP and MP routing coincide in these cases. For both the IB and the BBB NAC, the calculation of the budget capacity (cf. Equation (3.24) and Equation (3.28)) is independent of the routing function. With BBB NAC, the budget capacity $c(BBB_{v,w})$ effects a resource allocation along the shortest path from v to w, possibly partitioned over several paths depending on the routing. However, the sum of the allocated link bandwidths does not depend on the routing as the shortest paths have the same length. With IB NAC this is similar. The budget capacity $c(BBB_{v,w})$ effects a resource allocation along the shortest source tree from v to any $w \in \mathcal{V}$. Also here, a partitioning of the resource allocation over different paths is possible and depends on the routing. The sum of the allocated link bandwidth is not influenced by the routing, either, because the paths of a shortest source tree have the same length. Therefore, ECMP does not affect the overall required network capacity for IB and BBB NAC.

The results show that the resource utilization increases for IB/EB NAC with MP routing by about 3 percent points in the limit in both networks. In other words, less link capacity is required to carry the same traffic. This is not due to the multiplexing gain since the routing option does not change the budget capacities, cf. Equations (3.24) and (3.24). MP routing spreads the traffic of an individual b2b aggregate out over more links in the network than SP routing. This induces less bandwidth requirements on each single link. As traffic patterns are linear combinations of b2b aggregates, the same holds mostly for traffic patterns, that are admissible by the IB/EB NAC. This reduces the maximum achievable rate on specific links for which capacity must be provided. because they are linear compositions of b2b traffic aggregates.

Figures 3.18(a) and 3.18(b) illustrate that the resource utilization suffers at a load of $a_{b2b} = 100$ *Erl* 4 and 6.5 percent points for the ILB/ELB NAC, and 6.0 and 5.5 percent points for the ILB NAC in the Lab03 and the COST-239 network, respectively. MP routing spreads out the the traffic of a b2b aggregate over more links than with SP routing. This leads to a lower traffic concentration for $a(ILB_{l,v})$ and $a(ELB_{l,w})$ (cf. Equations (3.32) and (3.32)) and to reduced

multiplexing gain. As a result, the budget capacities increase which explains the rise in required network capacity and the reduced resource utilization for MP routing compared to SP routing. This effect can be so strong that the ILB NAC is even less efficient than the BBB NAC in the COST-239 network. In contrast, the resource efficiency of the LB NAC is hardly reduced (about 1 percent point for $a_{b2b} = 100$ *Erl*). Like above, the superposition of the per link traffic load from all b2b aggregates (cf. Equation (3.21)) is also distributed more evenly for MP routing than for SP routing. The thereby reduced economy of scale explains the slightly decreased resource utilization for LB NAC but this effect is not so strong.

In a nutshell, MP routing decreases the traffic concentration for path-aware budgets in comparison to SP routing, such that they can achieve less multiplexing gain regarding their capacity. Path-unaware budgets remain unaffected. The reduction of capacity requirements for the IB/EB NAC results from lower maximum traffic rates on individual links due to the traffic distribution.

3.6.4 Influence of the Network Topology

The network topology is a limiting factor for the routing and influences thereby also the traffic concentration in the network which affects the resource efficiency [223].

Construction of Random Networks

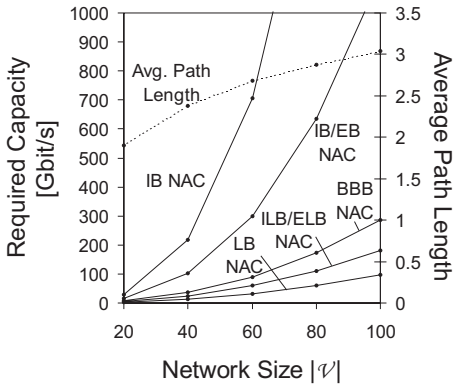
Salient features of a network are its size, its average node degree, and its internal structure. The authors of [31] propose algorithms for the random construction of inter-networks. However, we use our own construction methods (CMs) because we consider only a single autonomous system and we want to control the average and the maximum node degree quite rigidly, as well as the internal structure. Our CMs start by randomly connecting a spanning tree of $|\mathcal{V}|$ nodes. Then, edges are added while parallels and loops are avoided and the constraints on deg_{avg}, and deg_{max} are respected. We have implemented three different option for choosing a new edge.

- CM0 connects nodes with the largest distance within the graph in hops.
- CM1 connects nodes randomly.
- CM2 connects nodes with the shortest distance within the graph in hops.

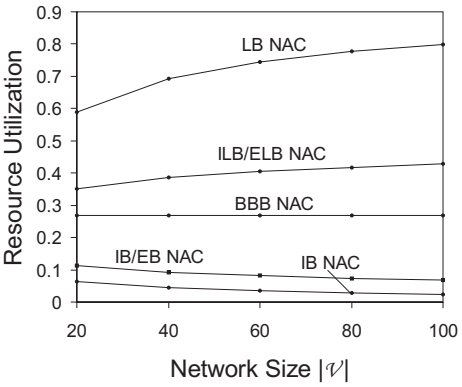
Since we want to have a decentralized network, we set the maximum node degree to $deg_{max} = deg_{avg} + 1$. If not mentioned differently, we construct random networks consist of 50 nodes with an average node degree of $deg_{avg} = 5$ using CM1 for our studies. For each data point we analyzed 10 different random networks to obtain small confidence intervals that are omitted in the figures. The traffic matrix is homogeneous with a small offered load of $a_{b2b} = 10$ *Erl*. This makes performance differences among the NAC types more visible due to their different ability to realize multiplexing gain.

Influence of the Network Size

Figure 3.19(a) illustrates that the required network capacity and the average path length increase with the network size $|\mathcal{V}|$. The growth is mainly due to our traffic model, i.e., the overall offered load scales about quadratically with the number of nodes ($a_{tot} = a_{b2b} \cdot |\mathcal{V}| \cdot (|\mathcal{V}| - 1)$). The number of links grows only linearly by $|\mathcal{E}| = \frac{|\mathcal{V}| \cdot deg_{avg}}{2}$. Hence, there is a linear growth of the offered load per link below the line, not yet taken into account that the average path length grows as well with increasing network size. Figure 3.19(b) reveals that only NAC methods based on path-aware budgets (LB, ILB, ILB/ELB NAC) can take advantage of this increased traffic concentration and achieve a larger resource utilization. For the sake of clarity, the curves for the ILB NAC are omitted in the figures. Their resource efficiency and capacity requirements lie between the ILB/ELB NAC and the BBB NAC. The resource utilization of the BBB NAC is independent of the network size since the offered load for one budget is exactly $a(BBB_{v,w}) = a(g_{v,w}) = a_{b2b}$. The performance of the IB NAC is low and decreases with increasing network size. The same holds for the IB/EB NAC but it outperforms the IB NAC significantly.

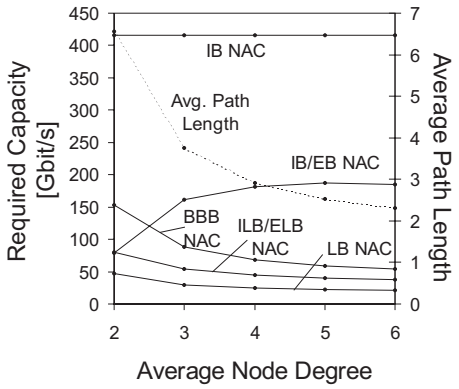


(a) Required network capacity

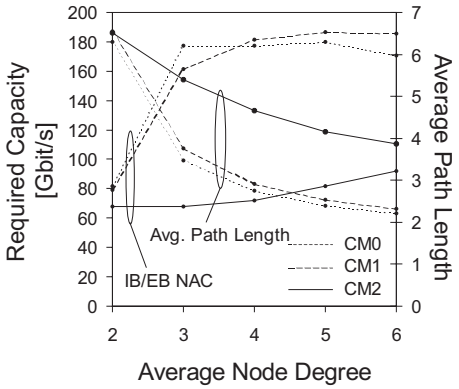


(b) Resource utilization.

Figure 3.19: *The sensitivity to the network size.*



(a) LB, ILB/ELB, and BBB NAC



(b) IB/EB NAC

Figure 3.20: *The sensitivity of the required network capacity to the average node degree.*

Influence of the Average Node Degree

Figure 3.20(a) shows the average path length and the required capacity for all BNAC types except for the ILB NAC. The required capacity for the LB, ILB/ELB, and LB NAC is clearly correlated with the the average path length because the overall traffic in the network depends on the average path length and the offered load which is constant in this experiment. The required capacity for the IB NAC is independent of the average node degree deg_{avg}. The shortest single-path routing tree seen by any source node is a spanning tree consisting of $(|\mathcal{V}| - 1)$ edges if all routers are border routers. Therefore, the required network capacity is $c(\mathcal{N}) = \sum_{v \in \mathcal{V}} c(IB_v) \cdot (|\mathcal{V}| - 1)$, which depends only on the network size. The IB/EB NAC restricts pathologic traffic patterns more efficiently than the IB NAC and requires less capacity. For a very small node degree, it can be even quite effective. However, it is remarkable that more capacity is required for an increasing average node degree although the average path length decreases. This phenomenon is explained by the next experiment.

Influence of Hierarchical Structures

Figure 3.20(b) illustrates that the average path length depends significantly on the average node degree and the construction method (CM). An average node degree of $deg_{avg} = 2$ yields almost a spanning tree network which is the basis for all CMs. Therefore, the average path length and the required capacity for the IB/EB NAC are about the same for all CMs in this case. An increasing average node degree shortens the average path length but enlarges the required network capacity. CM2 yields the longest paths and leads to most traffic in the network. However, it requires clearly less capacity than networks generated by CM0 and CM1. We explain these observations in the following.

We analyze these observations. In general, the average path length decreases with increasing deg_{avg} because more links allow in general for shorter paths. CM0 tries to add as many shortcuts as possible to the initial spanning tree which results in a relatively short path length. Randomly constructed networks

lead to approximately the same results. In contrast, CM2 avoids the installation of efficient shortcuts and yields a significantly larger average path length than CM0 and CM1. Therefore, the initial spanning tree structure dominates the CM2 topology and leads to a kind of traffic backbone since many shortest paths in the network use the links of the original spanning tree. Hence, CM2 networks reveal some hierarchical structure. To explain the reduced capacity requirements for CM2 for the IB/EB NAC, we consider a single link l. The set $\mathcal{X}$ are the routers that can send traffic over l and $\mathcal{Y}$ are the routers that can receive traffic over l. Thus, an upper bound for the link capacity is given by $c(l) \leq \min\left(\sum_{v \in \mathcal{X}} c(IB_v), \sum_{w \in \mathcal{Y}} c(EB_w)\right)$. If the traffic matrix is homogeneous, the link capacity can be effectively limited if $\mathcal{X}$ or $\mathcal{Y}$ are small. Hence, the performance of IB/EB NAC benefits from hierarchical networks with a backbone structure because they fulfill this condition. For example, the required capacity can be limited relatively well in networks with $deg_{avg} = 2$. An increasing average node degree increases the number of links, makes most nodes transit become nodes for multiple flows by providing shortcuts, and destroys the hierarchical structure of the network, which leads to larger capacity requirements. As CM2 provides less shortcuts than CM0 or CM1, the resource requirements for CM2 increase less. This experiment shows that the resource efficiency of the IB/EB NAC is very sensitive to the average node degree and to the internal network structure.

Summarizing, the different BNAC types differ significantly in their achievable resource utilization. This phenomenon is caused by two reasons. First, the BNAC methods have a different ability to obtain multiplexing gain for their budget capacities. Second, the BNAC-specific resource allocation practices lead to different resource efficiencies. As a consequence, the BNAC performance depends on many parameters. The impact of the offered load is most important while the effects of the traffic matrix and the routing are clearly weaker but still significant. As the traffic concentration increases with the network size, NAC methods based on path-aware budget gain in efficiency. Increasing the average node degree leads to shorter paths and decreases the required network capacity except for the IB/EB NAC which benefits from hierarchical network structures.

In conclusion, the LB NAC is most efficient but suffers from difficulties regarding its implementation. Among the truly stateless core BNAC methods, the ILB/ELB NAC is the most efficient one because its budgets are also path-aware.

3.7 Resilient BNAC

Resilience is the ability of a system to absorb failures without stopping or degrading an offered service. This means in the context of NAC that QoS must be guaranteed in case of network failures, too. Classical telephone systems offer traditionally very high reliability. They consist of oversupplied processors and switching fabrics, and they are operated in hot standby mode with backup machines. Hence, reliability is achieved by a high redundancy of hardware in the switching centers.

The KING project [2] pursues the idea of providing reliable QoS services for IP at cheaper costs by reducing the degree of redundancy for backup purposes. In case of a local network outage, e.g. a node or a link failure, many b2b aggregates may be affected and their service is interrupted. In such a case, reachability information is exchanged in IP networks by routing algorithms and routing tables are calculated anew. Due to this self-healing property the service is resumed when the routing tables have stabilized. However, QoS can only be maintained if sufficient resources are available on the deviation paths, otherwise, both deviated and resident flows possibly could suffer from congestion. Resilient NAC takes this aspect into account and limits the flow admissions to such a degree that no congestion is possible in the considered failure scenarios [224].

Under normal conditions where the network infrastructure is intact and the traffic rate conforms to the expected statistical behavior, overprovisioning requires a similar amount of bandwidth in the network like AC. There is just a tradeoff between QoS violation probability and blocking probability. However, AC is also a means to preserve QoS if unexpected events lead to congestion. These events can be new applications, BGP routing changes, or most probably link and node failures. Hence, AC is a kind of insurance against a shortage of bandwidth in situations for which a significantly increased blocking probability is rather accepted than a QoS degradation. However, traditional AC mechanisms fail in case of link outages where traffic is rerouted to backup paths. Firstly, the reservations of traditional AC schemes are usually bound to a specific path and

they do not protect the traffic on the backup route. Therefore, premium packets are possibly even discarded by a policer as they are classified out-of-profile due to a missing reservation context. Secondly, if they are not discarded, congestion occurs potentially on the backup link because the increased traffic rate has not been taken into account by the AC decision. Hence, traditional AC methods fail exactly in the case where they are needed most. Therefore, resilient NAC is one of the major contribution of this work.

3.7.1 Capacity Dimensioning for Resilient Networks

Appropriate capacity dimensioning is required for rerouted traffic in possible outage scenarios. First, the set $\mathcal{S}$ of protected failure scenarios $\mathcal{P}$ must be defined. Each $\mathcal{P} \in \mathcal{S}$ reflects a set of failed network elements $\mathcal{V}_{\mathcal{P}}^{F} \subseteq \mathcal{V}$ and $\mathcal{E}_{\mathcal{P}}^{F} \subseteq \mathcal{E}$. Since the set of working routers $\mathcal{V}_{\mathcal{P}}^{W} \subseteq \mathcal{V}$ and the set of working links $\mathcal{E}_{\mathcal{P}}^{W} \subseteq \mathcal{E}$ are different from $\mathcal{V}$ and $\mathcal{E}$, a new routing function $u^{\mathcal{P}}(l, g_{v,w})$ is provided. After all, we have a new networking scenario $\mathcal{N}_{\mathcal{P}}$ for every protected failure scenario $\mathcal{P} \in \mathcal{S}$. We denote the faultless networking scenario by $\mathcal{P}^{*}$ and define that it is always contained in $\mathcal{S}$ to facilitate the handling of the faultless case in the following.

The objective is to provide sufficient capacity $c(l)$ for each link $l \in \mathcal{E}$ that all admissible traffic can be carried in all failure scenarios $\mathcal{P} \in \mathcal{S}$. Hence, the required link capacity $c(l)$ can be calculated by

$$c(l) \quad \geq \quad \max_{\mathcal{P} \in \mathcal{S}} c_{\mathcal{P}}(l) \tag{3.43}$$

where $c_{\mathcal{P}}(l)$ is the required link capacity for the protected failure scenario $\mathcal{P} \in \mathcal{S}$. In the following we show how $c_{\mathcal{P}}(l)$ can be computed.

As outlined before, the NAC limits the traffic in the networks by Equations (3.1) – (3.6). They lead to the Inequalities (3.22), (3.25), (3.26), (3.29), (3.33), and (3.34) which can be used in a linear program to evaluate the required link capacities. In an outage scenario $\mathcal{P}$, the routing function $u(l, g_{v,w})$ is changed

to $u^{\mathcal{P}}(l, g_{v,w})$. This must be respected in the traffic maximization step in Equation (3.19). The BNAC remains unaware of the network outage in a failure case. Therefore, the constraints for the rate maximization process remain unchanged, i.e., the old routing function $u^{\mathcal{P}^*}(l, g_{v,w})$ is still applied for the side conditions.

Due to this change, the shortcuts for the calculation of the link capacities for the LB NAC in Equation (3.23) and for the ILB NAC in Equation (3.36) cannot be applied, and the time consuming Simplex method must be used like for the IB/EB and for the ILB/ELB NAC. The short solutions for the IB NAC in Equation (3.27) and for the BBB NAC in Equation (3.30) can be used if $u(l, g_{v,w})$ is substituted by $u^{\mathcal{P}}(l, g_{v,w})$.

The operation of the BBB NAC, ILB NAC, ILB/ELB NAC, IB NAC, and EB NAC does not change as their budgets are still controlled at the network edge. The LB NAC is more problematic because in most implementations the consulted LBs reside within the network and are bound to the path of a flow which is relocated due to rerouting in case of a failure. Therefore, the admission context is lost. Only the bandwidth broker implementation is a suitable LB NAC implementation that could support resilient NAC. For the sake of consistency in the bookkeeping of the reservations, it may further apply the original routing for AC decisions although the traffic follows a different path in the network.

3.7.2 BNAC Performance under Resilience Requirements

There are $\binom{|\mathcal{E}|}{n}$ possible failure scenarios with n different link failures in a network $\mathcal{N} = (\mathcal{V}, \mathcal{E})$. An outage with more link failures is less likely and its protection is more expensive. Therefore, we restrict the set of protected failure scenarios $\mathcal{S}$ to all single bi-directional link failure scenarios. We use conventional shortest path routing like in IS-IS or OSPF. The routing in a failure scenario $\mathcal{P}$ adapts to the new topology according to the shortest path algorithm and provides the rerouting function $u_{\mathcal{P}}$.

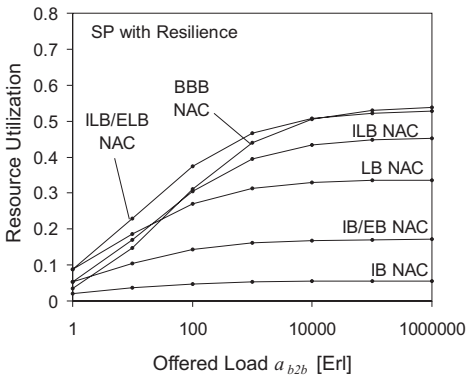
In the following, we compare the performance of all BNAC methods with

resilience requirements for the Lab03 and the COST-239 network topology with SP and MP routing.

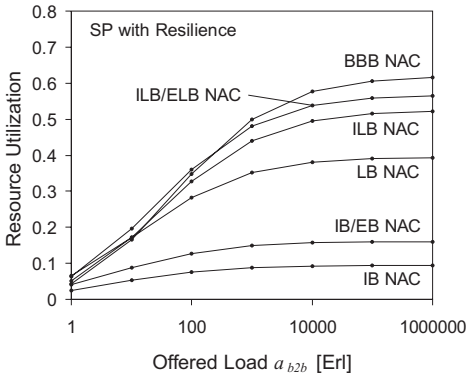
Impact of Resilience Requirements with Single-Path Routing

Figures 3.21(a) and 3.21(b) show the resource utilization for all NAC methods under resilience requirements depending on the average offered b2b load. It reveals a completely different performance behavior compared to the resource utilization without resilience requirements in Figures 3.15(a) and 3.15(b). A comparison of Figures 3.21(a) and 3.21(b) shows that all NAC types have different network-specific asymptotes for their resource utilization. They are also compiled in Table 3.7 for the sake of clarity. The BBB NAC outperforms the ILB/ELB NAC, the ILB NAC, and the LB NAC. Except for ILB NAC and ILB/ELB NAC, this is the reversed order from the scenario without resilience. The performance of the IB and IB/EB NAC is significantly worse.

With resilience requirements, the BBB NAC achieves only 60% in the COST-239 network (54% for Lab03) resource utilization in the limit instead of 100% without resource requirement. The reciprocal value $\frac{1}{0.6} \approx 1.67$ is the average degree of overdimensioning required for the survivability in outage scenarios and corresponds to 67% (85%) additional backup capacity (cf. Table 3.7). This finding confirms the idea that network resilience for QoS services can be provided at a cheaper cost than backing up all the resources which means 100% additional capacity. This is due to the fact that backup capacity can be shared among different rerouted b2b traffic aggregates in different failure scenarios.



(a) Lab03 network.



(b) COST-239 network.

Figure 3.21: *Resource utilization for single-path routing with resilience requirements.*

Table 3.7: *Limiting performance for SP routing.*

BNAC type	max. util. without res.	max. util. with res.	additional capacity
Lab03 network			
BBB NAC	100%	54 %	85 %
ILB/ELB NAC	100%	53 %	89 %
ILB NAC	100%	45 %	122 %
LB NAC	100%	34 %	194 %
IB/EB NAC	27 %	17 %	58 %
IB NAC	10 %	6 %	67 %
COST-239 network			
BBB NAC	100%	60 %	67 %
ILB NAC	100%	52 %	92 %
ILB/ELB NAC	100%	56 %	79 %
LB NAC	100%	40 %	150 %
IB/EB NAC	22 %	16 %	38 %
IB NAC	16 %	9 %	77 %

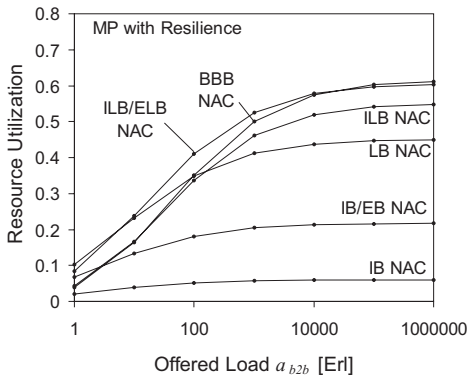
With resilience requirements, the resource utilization for the LB NAC decreases to 40% (34%) which corresponds to 150% (194%) additional costs for backup purposes. Hence, the LB NAC is clearly more expensive than the BBB, ILB/ELB, and ILB NAC from a resource point of view. In addition, it is not able to offer cheap resilience for QoS services because simple duplication of the link resources requires less capacity.

There is an explanation for that phenomenon. The LB NAC is more flexible than the BBB NAC with regard to the use of allocated link capacities, i.e., more traffic patterns can be supported with the same capacity. As less capacity suffices to obtain the same QoS level, the LB NAC has a better resource efficiency than the BBB NAC in the non-resilient case. Though, this flexibility is a drawback with resilience requirements since all admissible traffic patterns must

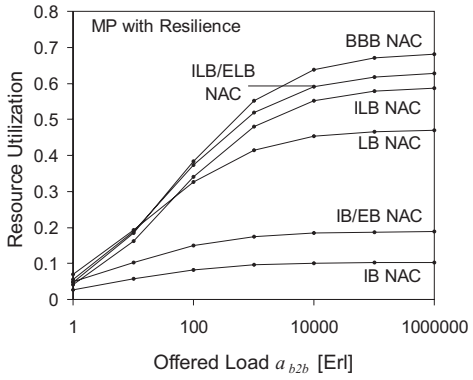
be protected. The traffic patterns that are accepted by the LB NAC but not by the BBB NAC are unrealistic. However, if they appear, their impact is more extreme. In contrast to the BBB NAC, but similar to the IB NAC, the LB NAC accepts the traffic pattern where one b2b traffic aggregate $g_{v,w}$ consumes almost all capacity along its path and other aggregates vanish. If a link of that path fails, a tremendous amount of traffic is deviated on a single backup path. The same example holds for any other b2b aggregate, which makes the capacity requirements for backup purposes very large. The BBB NAC prohibits this scenario by excluding such extreme traffic patterns from admission. Therefore, the LB NAC has a lower resource efficiency than the BBB NAC in case of resilience requirements. In a nutshell, a large NAC with regard to traffic patterns achieves a high resource utilization without resilience requirements but it requires much additional capacity for backup purposes which causes a low resource utilization with resilience requirements.

The resource efficiency of the ILB/ELB NAC is 56% (53%) in the limit, which corresponds to 79% (89%) additional network resources. The ILB/ELB NAC has a worse performance than the BBB NAC due to more resource flexibility. Yet due to this flexibility, the ILB/ELB NAC can lead to better results than the BBB NAC at low offered load in larger networks like the Lab03 network. However, the computation of the required capacity is very time consuming due to the mandatory application of the Simplex rate maximization in Equation (3.19). This and the fact that many budgets have to be checked for admission make the approach somewhat impractical. The same holds for the ILB NAC which requires 92% (122%) more capacity with a maximum resource utilization of 52% (45%).

Resilience requirements decrease the performance of the IB/EB NAC from 22% (27%) to 16% (17%). The additional expenses for backup purposes are only 37.5% (58%) but the absolute required network capacity exceeds the demand of the other NAC methods by far and leaves the IB/EB still unattractive. The same holds for the IB NAC which requires 77% (66%) more capacity for failure protection with a maximum resource utilization of 9% (6%) opposed to 16% (10%) without resilience requirements.



(a) Lab03 network.



(b) COST-239 network.

Figure 3.22: *Resource utilization for multi-path routing with resilience requirements.*

Impact of Resilience Requirements with Multi-Path Routing

We make the same experiments with MP Routing. Figures 3.21(a) and 3.21(b) show the load-dependent resource utilization and Table 3.8 compiles the utilization limits and the required backup capacity. We observe a significant performance improvement if we compare these data with Table 3.7.

Table 3.8: *Limiting performance for MP routing.*

BNAC type	max. util. without res.	max. util. with res.	additional capacity
Lab03 network			
BBB NAC	100%	61 %	64 %
ILB/ELB NAC	100%	60 %	67 %
ILB NAC	100%	55 %	82 %
LB NAC	100%	45 %	122 %
IB/EB NAC	30 %	22 %	36 %
IB NAC	10 %	6 %	67 %
COST-239 network			
BBB NAC	100%	68 %	47 %
ILB NAC	100%	63 %	58 %
ILB/ELB NAC	100%	59 %	69 %
LB NAC	100%	47 %	113 %
IB/EB NAC	25 %	19 %	32 %
IB NAC	16 %	10 %	60 %

We briefly explain why MP routing leads to less backup capacity. MP effects a better traffic distribution across the network and offers possibly more than one backup route in a failure case. If the traffic can be deviated in case of a link failure over more than one path, the maximum required backup resources on the backup routes are smaller. Thus, less backup capacity suffices per link which provides in

turn a partial protection for more, different outages. This observation applies to the performance of all BNAC methods. As this reduction of backup resources is due to routing, we recognize a potential for capacity savings that can be optimized and tackle this problem in Chapter 4.3.

3.8 Capacity Assignment to NAC Budgets

So far, we have identified different NAC categories and investigated their efficiency. We dimensioned the capacity of the NAC budgets such that a given traffic matrix could be supported with a given flow blocking probability. The NAC budgets and resilience requirements for a set of protected failure scenarios conditioned the link capacities. For the use of NAC in production environments, this process must be reversed. The link capacities are given and the NAC budgets must be set in such a way that they yield blocking probabilities as small as possible for all flows in the presence of a given traffic matrix. In addition, overbooking of the network resources must be avoided and all resilience requirements must be respected.

The solution to that problem reveals three challenges:

- Several budgets compete for the capacity of a single link.
- The maximum capacity of a single budget is limited by several links that are determined by the routing in the network.
- The routing depends on the active network elements in the protected failure scenarios.

In the following, we propose alternative solutions to these problems:

- link budget assignment (LBA) strategies,
- network budget assignment (NBA) strategies, and
- resilient budget assignment (RBA) strategies.

We investigate their ability to achieve low flow blocking probabilities for limited network resources.

3.8.1 Link Budget Assignment Strategies

We consider a single link l whose capacity must be shared by a set of budgets $\mathcal{B}(l)$. The offered traffic load $a(b)$ that is covered by each budget $b \in \mathcal{B}(l)$ may be different. As overbooking is not allowed, the link capacity must be partitioned among the budgets in $\mathcal{B}(l)$. We suggest two different LBA strategies.

Proportional LBA

A naive LBA strategy assigns the link capacity $c(l)$ to the budgets proportionally to their offered load. Hence, all budgets $b \in \mathcal{B}(l)$ have the same relative size

$$\xi(b) \quad = \quad \xi(l) = \frac{c(l)}{\sum_{b^* \in \mathcal{B}(l)} a(b^*)} \cdot u(l,b) \tag{3.44}$$

related to their offered load $a(b)$. The routing function $u(l,b)$ tells the percentage of the traffic covered by budget b that uses link l. This leads to an absolute budget size of

$$c(b) \quad = \quad a(b) \cdot \xi(l). \tag{3.45}$$

This proportional LBA (PLBA) strategy does not take economy of scale into account and entails unequal flow blocking probabilities $p(b)$ at different budgets b if they have a different offered load $a(b)$. We call $p(b)$ also the budget blocking probability. This consideration leads to a vague notion of unfairness. Fairness is given if all traffic aggregates face the same blocking probabilities on that link.

Fair LBA

The fair LBA (FLBA) strategy assigns the link capacity $c(l)$ to the budgets in such a way that the budget blocking probability $p(b)$ is about the same for all budgets $b \in \mathcal{B}(l)$. As this cannot be done in closed form, we solve that task by Algorithm 9. To be compliant with the algorithms in Section 3.3, a capacity c is measured in $c_u = \frac{c}{u_c}$ units of capacity u_c. After initializing the budget

capacities with zero, the link capacity is assigned iteratively by increasing the capacity of the budget with the largest blocking probability by a certain capacity increment c_u^{inc}. The function $c_u^{free}(l) = c_u(l) - \sum_{b \in \mathcal{B}(l)} c_u[b] \cdot u(l,b)$ calculates the remaining free capacity units on l. If several budgets have the same budget blocking probability, we take the one with the largest offered load because its blocking probability will be least decreased. Algorithm 9 may be slow if c_u is large as the capacity units are assigned one after another due to $c_u^{inc} = 1$. It can be accelerated by computing a suitable capacity increment.

Input: $l, \mathcal{B}(l)$

for all $b \in \mathcal{B}(l)$ **do** {initialize}
  $c_u[b] := 0$
end for
while $c_u^{free}(l) > 0$ **do**
  choose $b^* \in \mathcal{B}$ with largest blocking probability and take a budget with maximum offered load for tie breaking
  $c_u^{inc} := \min(1, c_u^{free}(l)$
  $c_u[b^*] := c_u[b^*] + c_u^{inc}$
end while

Output: assigned budget capacities $c_u[b], b \in \mathcal{B}(l)$

Algorithm 9: FAIRLBA: fair link budget assignment.

Simple and Fast Acceleration A simple acceleration is setting c_u^{inc} to a fraction of c_u proportional to its offered load. However, this yields the PLBA strategy which is not good because it yields large blocking probabilities for budgets with small offered load. Therefore, we relax it by taking only a fraction $\frac{1}{h}$ (usually $h = 2$) of the proportional assignment

$$c_u^{inc} = \max\Big(\min(1, c_u^{free}(l)), \min_{l \in \mathcal{B}(l)} \Big(\lfloor \frac{q \cdot a(b^*)}{h} \rfloor\Big)\Big) \tag{3.46}$$

with $q = \frac{c_u}{\sum_{b \in \mathcal{B}(l)} a(b)}$ and c_u being the remaining capacity in Algorithm 9. Additionally, we take budgets with the least offered load for breaking ties because those consume the least capacity and cause, therefore, the least unfairness. This approach is very fast but its correctness cannot be proven in the sense that some budgets may be penalized. If the completed capacity assignment result shows that some budgets with little offered load have comparatively large blocking probabilities, then Algorithm 9 must be run again with a larger h in Equation (3.46).

Safe Acceleration We new present a safe acceleration of Algorithm 9 which is based on the above idea but avoids the starvation of budgets with little offered load. Algorithm 10 computes safe capacity increments. A capacity increment is safe if it is so small that it decreases the candidate budget only to such an extent that any other budget $b \in \mathcal{B}(l)$ increased by its fair share can undergo the resulting blocking probability. The variable q^{dec} controls the granularity and the speed of the algorithm. This mechanism is considerably more computation intensive because the calculation of the blocking probability is quite time consuming.

Performance Comparison

Due to the different complexity of the PLBA and FLBA strategy it is important to know the impact of theses methods on the quality of the obtained results. As mentioned above, PLBA does not take economy of scale into account and leads to unfair results. We illustrate this suspicion in the following by considering the capacity assignment on a single link l. For simplicity reasons we conduct our experiments with only two budgets b_0 and b_1. We assume a fixed link load $a(b_0)+ a(b_1) = a(l) = 100$ *Erl*. Since there are only two competing budgets, the load distribution among them is characterized by the load fraction $lf(b_i) = \frac{a(b_i)}{a(l)}$. We dimension the budget capacities $c(b_i)$ for a desired blocking probability $p_c(b_i) = 10^{-3}$ and set the link capacity to $c(l) = c(b_0) + c(b_1)$.

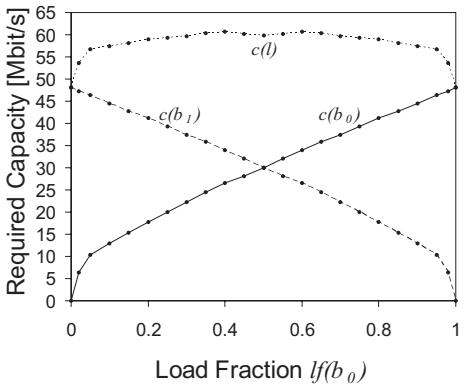
Input: unassigned capacity $c_u^{free}(l)$, $\mathcal{B}(l)$ with already assigned capacities

$q := \frac{c_u^{free}(l)}{\sum_{b \in \mathcal{B}} a(b)}$
choose $b^* \in \mathcal{B} > 0$ with largest blocking probability and use budget with smallest offered load for tie breaking
$c_u^* := \lfloor q \cdot a(b^*) \rfloor$
$p^* := p\big(a(b^*), c_u[b^*] + c_u^*\big)$
for all $b^+ \in \mathcal{B}$ **do**
  $c_u^+ := \lfloor q \cdot a(b^+) \rfloor$
  $p^+ := p\big(a(b^+), c_u[b^+] + c_u^+\big)$
  while $p^* < p^+$ **do**
    $c_u^* := \lfloor q^{dec} \cdot c_u^* \rfloor$
    $p^* := p\big(a(b^*), c_u[b^*] + c_u^*\big)$
  end while
end for

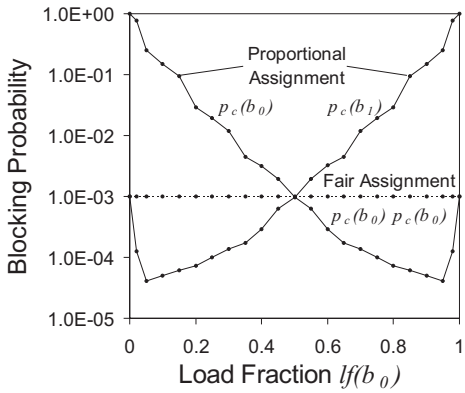
Output: safe capacity increment c_u^*

Algorithm 10: SAFEACC: calculation of a safe capacity increment c_u^{inc}.

Impact of the Load Fraction Figure 3.23(a) shows the budget sizes $c(b_i)$ and the required link capacity $c(l)$ for different load fractions $lf(b_0)$. The least capacity is required for $lf(b_0) = 0$ or $lf(b_0) = 1$ because then, b_1 or b_0 can be dimensioned most efficiently due to economy of scale. As a next step, we reassign the obtained link capacity $c(l)$ to the budget capacities $c(b_i)$ according to the proportional and to the fair LBA strategy. Figure 3.23(b) illustrates the resulting budget blocking probabilities. Due to the construction of the experiment, the blocking probabilities for FLBA are exactly 10^{-3}. For PLBA, the blocking probabilities depend on the load fraction. The value of $p_c(b_i)$ is larger than 10^{-3} if $lf(b_i) < 0.5$ and smaller, otherwise. This is clearly unfair. For example, we get values of $p_c(b_0) \approx 10^{-4}$ and $p_c(b_1) \approx 10^{-1}$ for $lf(b_0) = 0.2$.

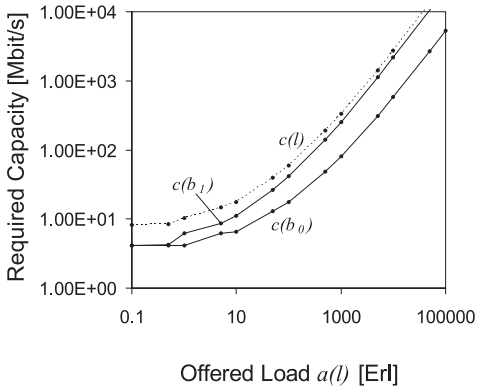


(a) Required budget and link capacities for $p_c = 10^{-3}$.

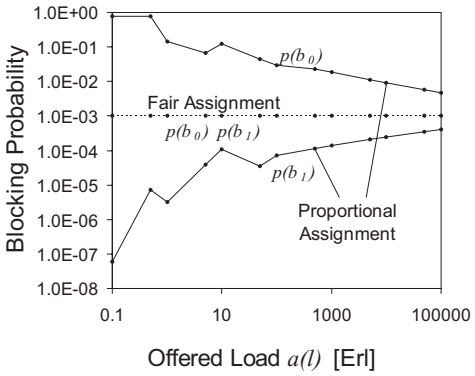


(b) Resulting budget blocking probabilities after capacity reassignment.

Figure 3.23: *Impact of the link load distribution among budgets.*



(a) Required budget and link capacities for $p_c = 10^{-3}$.



(b) Resulting budget blocking probabilities after capacity reassignment.

Figure 3.24: *Impact of the offered link load.*

Impact of the Offered Load We conduct the same experiment for a fixed load fraction $lf(b_0) = 0.2$ and vary the offered link load $a(l)$. Figure 3.24(a) reveals that for a very low link load $a(l)$ both b_0 and b_1 require a capacity of about 2 Mbit/s which corresponds to the maximum request size $c(r_2)$. For large values of $a(l)$, the required capacities for both budgets rise about linearly with the offered link load. According to Figure 3.24(b) the blocking probability $p_c(b_0)$ is about 10^{-2} even for large offered link load and the blocking probability $p_c(b_1)$ does not exceed 10^{-4}. Hence, PLBA is clearly unfair for all link loads, too.

3.8.2 Definition of Unfairness

So far, we only have a definition for fair resource assignment but we do not have a measure for unfairness, yet. In our experiments, some of the PLBA budget blocking probabilities $p^{PLBA}(b)$ are larger and some others are smaller than the values $p^{FLBA}(b)$ for FLBA. We use the positive difference in the graphs as a metric for unfairness because the FLBA curves show how budget blocking probabilities could be. For further use, we define unfairness formally by

$$\Delta = \frac{\sum_{b \in \mathcal{B}} \max(\log(p^{PLBA}(b)) - \log(p^{FLBA}(b)), 0)}{|\mathcal{B}|}. \quad (3.47)$$

This idea can be extended to an entire network if fair reference probabilities corresponding to $p^{FLBA}(b)$ are defined.

3.8.3 Network Budget Assignment Strategies

In this section, we consider the dimensioning of NAC budgets in the context of a network and not only of a single link. We respect the constraints arising from the different budget types.

LBs, ILBs, and ELBs are *link-specific budget types* as they pertain only to a single link l. The capacity of that link can be partitioned among the corresponding budgets according to the algorithms in the previous section.

IBs, EBs, and BBBs are *non-link-specific budget types* as they admit traffic to flow over several links, i.e., their scope is not limited to the AC of flows for a specific link. In turn, when capacity is assigned to a non-link-specific budget b, the set of all links for which the budget admits flows must be respected. The set of these links $\mathcal{L}(b)$ is given by

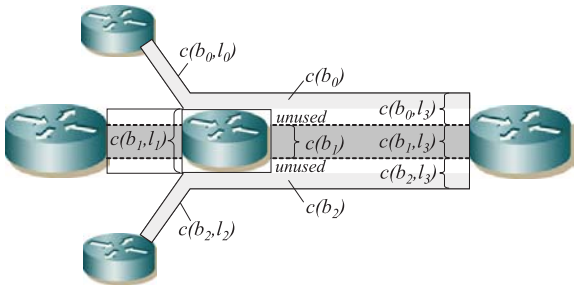
$$\mathcal{L}(b) = \begin{cases} \{l : l \in \mathcal{E} \wedge \sum_{w \in \mathcal{V}} a(b) \cdot u(l, g_{v,w}) > 0\} & \text{for } b = IB_v \\ \{l : l \in \mathcal{E} \wedge \sum_{v \in \mathcal{V}} a(b) \cdot u(l, g_{v,w}) > 0\} & \text{for } b = EB_w \\ \{l : l \in \mathcal{E} \wedge a(b) \cdot u(l, g_{v,w}) > 0\} & \text{for } b = BBB_{v,w} \end{cases} \quad (3.48)$$

Independent NBA

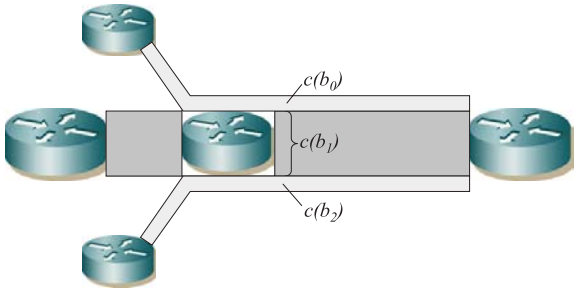
We propose first a simple and intuitive algorithm for the capacity assignment to non-link-specific NAC budgets. For each link $l \in \mathcal{E}$ link-specific capacities $c(b, l)$ are assigned to non-link-specific budgets b according to an LBA strategy in Section 3.8.1 whereby only $u(l, b) \cdot c[b]$ capacity. The actual capacity $c(b)$ of a budget b is then calculated as the minimum of all link-specific budget capacities $c(b, l)$ by

$$c(b) = \min_{l \in \mathcal{L}(b)} c(b, l). \quad (3.49)$$

As the budget capacity is first assigned independently on any link, we call this method the independent NBA (INBA). Figure 3.25(a) shows that this method leaves $c(b) - c(b, l)$ capacity unused for each budget b on each link l which can be considerable if the network is dimensioned in such a way that the budget blocking probabilities are substantially different. However, this unused capacity can be assigned to other budgets to further reduce their blocking probability like in Figure 3.25(b). This is proposed by the concurrent NBA.



(a) Some of the capacity on l_3 remains unused with INBA.



(b) The CNBA reassigns unused capacity to other budgets.

Figure 3.25: *An example for capacity utilization with INBA and CNBA.*

Concurrent NBA

Knowing about the inefficiency of INBA, we describe Algorithm 11 (CNBA) to avoid this problem. The presented algorithm is correct for BBB NAC and pure IB NAC. It must be enhanced for IB/EB NAC. We denote the set of all budgets in the network by $\mathcal{B}$. At the beginning, all budgets are initialized to zero and the set of unassigned budgets is $\mathcal{B}_{hot} = \mathcal{B}$. The free capacity of a link l is $c_u^{free}(l) = c_u(l) - \sum_{b \in \mathcal{B}} u(l, g_{b,}) \cdot c_u[b]$. To increase the budgets successively, a budget b^* with the currently largest blocking probability $p(b)$ is chosen and in case of ambiguity, a budget among them with a maximum offered load is taken. If there is enough capacity on all links supporting budget b^* ($\forall l \in \mathcal{E} : c_u^{free}(l) \geq c_u^{inc} \cdot u(l, b)$), the budget capacity is enlarged by c_u^{inc}. Otherwise, the budget is removed from $\mathcal{B}_{hot}$. We used efficient data structures to speed up the algorithm but we do not discuss them here for clarity reasons. Optionally, this procedure may stop if the blocking probability $p(b)$ of the unassigned budgets $b \in \mathcal{B}_{hot}$ falls below a predefined threshold p_{min}. This would possibly leave some spare capacity in the network. Algorithm 11 implements the FLBA strategy but it can also be adapted to PLBA by using $\xi(b)$ instead of $p(b)$ [225]. It can also be enhanced by the above explained acceleration mechanisms [226].

Performance Comparison

A complete budget assignment method (BAM) consists of a link and a network budget assign strategy. The first one impacts the fairness and the second one the efficiency. For easier reference to the BAMs, we abbreviate the combinations of (PLBA, FLBA) $\times$ (INBA, CNBA) by PL&IN, PL&CN, FL&IN, and FL&CN.

Impact of Unequal Load Distribution on the Unfairness We illustrate the impact of the unequal load distribution on the unfairness of the BAMs in the Lab03 network of Figure 3.14(a). We choose budgets for the BBB NAC, set the b2b offered load to $a_{b2b} = 10$ *Erl*, and apply the exponential extrapola-

Input: (implicitly: topology, routing, budgets)

for all $b \in \mathcal{B}$ **do** {initialize}
$c_u[b] := 0$
end for
$\mathcal{B}_{hot} := \mathcal{B}$
while $\mathcal{B}_{hot} \neq \emptyset$ **do**
choose $b^* \in \mathcal{B}_{hot}$ with largest blocking probability and take a budget with maximum offered load for tie breaking
$c_u^{inc} := 1$
if $\left(\forall l \in \mathcal{E} : c_u^{free}(l) \geq c_u^{inc} \cdot u(l, b^*)\right)$ **then**
$c_u[b^*] := c_u[b^*] + c_u^{inc}$
else
$\mathcal{B}_{hot} := \mathcal{B}_{hot} \setminus b^*$
end if
end while

Output: assigned budget capacities $c_u[b], b \in \mathcal{B}$

Algorithm 11: CNBA: concurrent network budget assignment.

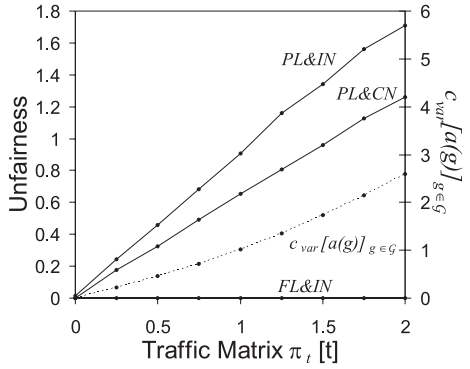
tion according to Equation (3.42). We first dimension the network size for a b2b flow blocking probability of $p = 10^{-3}$ and then reassign the capacity to the budgets according to the four different BAMS. Since FL&CN is fair and efficient by definition, we use it as a reference to assess the unfairness of the other BAMs.

Figure 3.26(a) shows the mean unfairness Δ per aggregate of the different BAMs. The coefficient of variation of the traffic aggregate sizes $c_{var}(a(g)_{g \in \mathcal{G}})$ shows that the variability of the entries in the traffic matrix increases with an increasing extrapolation parameter t. Since the network capacity has been dimensioned for a blocking probability p_{b2b} for all budgets, FLBA succeeds to assign link budget capacities $c(b, l)$ with a blocking probability of exactly 10^{-3}. As they have the same size on any link $l \in \mathcal{L}(b)$, every $c(b)$ reaches the minimum

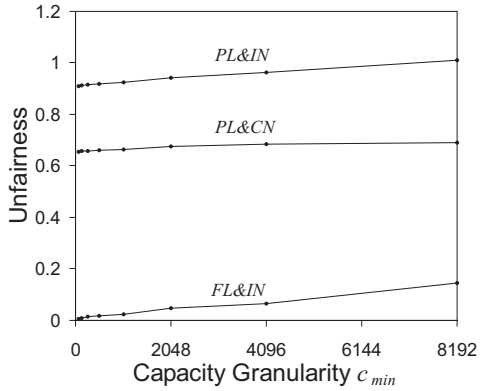
blocking probability. Therefore, both FL&CN and FL&IN are equally fair.

For $t = 0$, PL&IN and PL&CN also yield fair results because all aggregates have the same offered load $a(g_{v,w}) = a_{b2b}$, so the proportional and the fair LBA yield the same budget capacities on any link. For increasing t, the load distribution becomes more heterogeneous such that the unfairness due to proportional LBA increases with the variation of the traffic matrix. PL&IN is more unfair than PL&CN because some capacity remains in addition, due to the independent NBA strategy.

Impact of Capacity Granularity on the Unfairness In the previous experiments we have seen that the NBA strategy has no impact if the FLBA strategy is applied. Now, we assume the original traffic matrix, i.e. $t = 1$, and that the network is dimensioned for $p_{b2b} = 10^{-3}$. However, only multiples of a finest granularity c_{min} can be provided as bandwidth portions, i.e., the correctly dimensioned link capacities are rounded up. Figure 3.26(b) shows the impact of c_{min} on the unfairness. FL&IN is fair for a granularity of $c_{min} = 64$ Kbit/s. In this scenario, the capacity granularity is not a restriction since all request sizes in our model are a multiple of that quantity and the fairness of FL&IN can be explained like above. For increasing c_{min}, the unfairness of FL&IN, PL&CN and PL&CN slightly increases whereby PL&CN suffers the least from the coarser capacity granularity due to the CNBA strategy. However, the major impact on the unfairness in this experiment results also from the unbalanced load distribution from which the PLBA strategy suffers. Hence, if the link capacities in the network were properly dimensioned, the FLBA strategy is most important to achieve fair budgets and the NBA strategies play a minor role. This is different if networks are not properly designed.



(a) Unfairness due to differences in offered load.



(b) Unfairness due to quantized capacity assignment.

Figure 3.26: *Unfairness of budget assignment methods relative to* FL&CN.

3.8.4 Resilient Budget Assignment

If a local outage occurs in a network, the traffic must be rerouted. Therefore, sufficient capacity is required on the rerouted path or - in other words - the NAC must limit the admitted traffic to such a level that the capacity suffices [226]. The set $\mathcal{S}$ comprises all considered failure scenarios s which contain the remaining active network topology. Like above, the working scenario is included in $\mathcal{S}$. For each failure scenario, the routing for the traffic of a budget changes and we describe it by the enhanced routing function $u^s(l,b)$. In the following, we present a simple and an enhanced method to extend the presented capacity assignment algorithms for resilience requirements.

Independent RBA

A simple extension of the above algorithm is an independent capacity assignment $c_u[s,b]$ for all failure scenarios $s \in \mathcal{S}$ and a subsequent capacity minimization $c_u[b] = \min_{s\in\mathcal{S}} c_u[s,b]$. This independent RBA (IRBA) yields obviously safe values for all considered failure scenarios.

Concurrent RBA

The concurrent RBA (CRBA) performs faster and yields more efficient results than the preceding approach. Basically, the capacity of all budgets is increased concurrently in all failure scenarios until they are limited by a capacity bottleneck on some link in some failure scenario. We define failure scenario depending functions

$$c_u^{free}(s,l) = c_u(l) - \sum_{b\in\mathcal{B}} c_u[b]\cdot u^s(l,b) \tag{3.50}$$

$$a_{hot}(s,l) = \sum_{b\in\mathcal{B}_{hot}} a(b)\cdot u^s(l,b) \text{ and} \tag{3.51}$$

$$q(s,l) = \frac{c_u^{free}(s,l)}{a_{hot}(s,l)} \tag{3.52}$$

The adaptation of this algorithm is done by the reformulation of the condition in Algorithm 11 by $\left(\forall s \in \mathcal{S} \forall l \in \mathcal{E} : c_u^{free}(s,l) \geq c_u^{inc} \cdot u^s(l,b^*)\right)$. For the acceleration purposes, Equation (3.46) changes to

$$c_u^{inc} = \max\Big(\min_{l \in \mathcal{L}(b^*)}(1, c_u^{free}(l)), \min_{s \in \mathcal{S}, l \in \mathcal{E}: u^s(l,b)>0}\Big(\lfloor \frac{q(s,l) \cdot a(b)}{h} \rfloor\Big)\Big). \quad (3.53)$$

Performance Comparison

If networks are well designed for the offered load, the simple and the enhanced extension for resilience requirements lead almost to the same results. However, networks are static and traffic load changes such that they do not always fit together. In such a case, the enhanced extension method leads to more efficient budget assignments.

We take the network in Figure 3.27 and consider only a single failure scenario for resilience. We assume that the aggregate flows f_0 and f_1 have the same offered load. For the sake of simplicity, we indicate the budgets by their corresponding aggregate flows. The simple resilience extension calculates $c[s_0, f_0] = c[s_0, f_1] = 5$ *Mbit/s* for the case s_0 without any failure, and $c(s_1, f_0) = 7.5$ *Mbit/s*, $c(s_1, f_1) = 2.5$ *Mbit/s* for the case s_1 that the 5 *Mbit/s* link fails. Hence, the allowable budget capacities are $c[f_0] = 5$ *Mbit/s* and $c(f_1) = 2.5$ *Mbit/s*.

The enhanced resilience extension raises both budget capacities concurrently until $c[f_1] = 2.5$ *Mbit/s* is fixed due to the failure scenario s_1. Then, the other budget can take advantage of the full remaining capacity of the 10 *Mbit/s* link and it is finally set to $c(f_0) = 7.5$ *Mbit/s*.

This small example illustrates the operation of both algorithms and shows that the enhanced resilience algorithm leads to more efficient results than the simple version. To show that this phenomenon is not a pathological artefact, we validate this finding in the Lab03 network whose links are provisioned with 1 *Gbit/s*. We dimension the budgets with both resilience extension methods under consideration of all possible single link failures. To that aim, we scale the offered load in such a way that it yields blocking probabilities of at most 2%. We limit the maximum budget size by a minimum budget blocking probability of 10^{-6}.

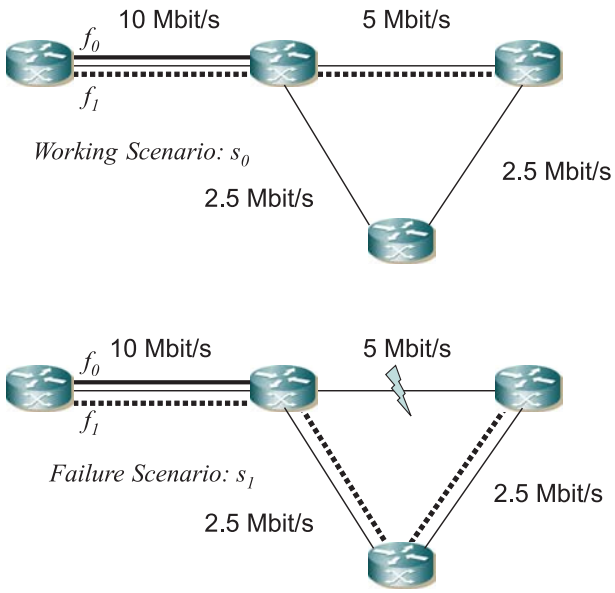


Figure 3.27: *Small networking scenarios.*

The budget sizes are significantly larger if they are calculated by the CRBA method instead by the IRBA method. Figures 3.29(a) and 3.29(b) present a distribution of the absolute and the relative capacity gain by CRBA compared to IRBA. More than half of the budgets remains unaffected and does not profit from CRBA. The additional budget capacity of the increased budgets differs considerably and the distribution for the absolute and the relative gain is different because the traffic matrix is heterogeneous. The average absolute gain is about 2 *Mbit/s* per budget and the average relative gain is about 6.6% per budget. Figure 3.28 shows the difference of the respective logarithmic blocking probabilities. The budget blocking probabilities obtained with CRBA are up to 4 orders of mag-

nitude smaller than those obtained with IRBA, and on average this advantage is 0.47 orders of magnitude. Hence, the benefit of the enhanced resilience extension is also clearly visible in large networks.

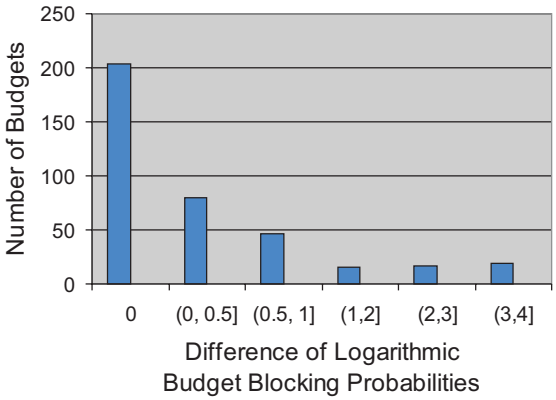
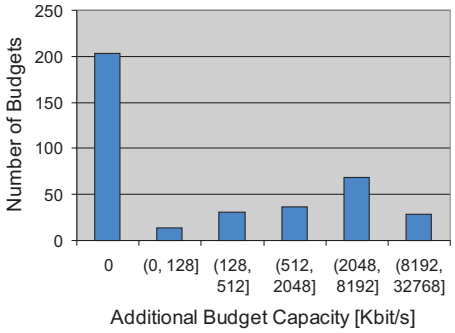
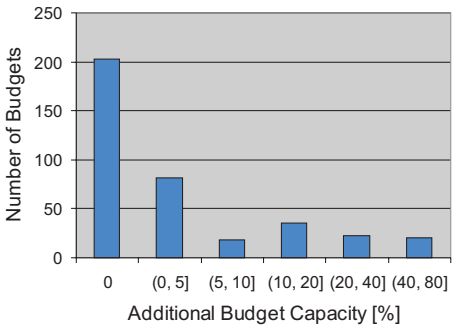


Figure 3.28: *Concurrent RBA obtains smaller budget blocking probabilities than independent RBA.*



(a) Absolute capacity gain per budget.



(b) Relative capacity gain per budget.

Figure 3.29: *Concurrent RBA assigns larger budget capacities than independent RBA.*

4 Routing Optimization for Resilient Networks

Companies depend on the reliability of their communication infrastructure and an outage translates immediately into a financial loss [227, 228]. Therefore, carrier grade networks are expected to provide an availability of “five nines” (99.999%) [229] in spite of the fact that network elements can fail. This challenge arises, e.g., for Virtual Private Networks (VPNs) or in the terrestrial radio access network (UTRAN) of the Universal Mobile Telecommunication System (UMTS). Today’s IP technology enables a global interconnection of remote hosts and servers on a best effort basis. It does not meet the requirements of carrier grade networks but the wide deployment and the simple operation of current IP networks call for Next Generation Networks (NGN) based on IP technology to substitute frame relay and ATM solutions.

Conventional telephone networks achieve the “five nines” reliability by massive redundancy of hardware provisioning. Also some of today’s IP networks are protected against potential link failures by backup lines or by SDH rings on the physical layer. However, these methods require 100% or more backup capacity because the backup line is only used in case that the primary line fails. The same holds for SDH rings because only one of both paths on the ring from a sender to a receiver is needed during faultless operation like in Figure 4.1. Then, the ring

is vulnerable until the failure is repaired.

Figure 4.1: *Failure protection by rings costs at least 100% backup capacity.*

With packet-switched networks, a similar reliability can be achieved by traffic deviation over alternate paths in case of local outages. However, the backup capacity can be shared among different traffic aggregates in different failure scenarios. Hence, backup capacity sharing offers the possibility to reduce the required backup capacity without compromising the failure resilience of the network. Therefore, resilience mechanisms at the network layer can achieve resilience at a cheaper cost than traditional physical layer protection mechanisms.

In this chapter we give an overview of resilience mechanisms and routing optimization in general. We discuss approaches from literature to optimize resilient routing and we derive side conditions for practical protection switching mechanisms. Based on these constraints, we develop several novel protection switching mechanisms. We optimize them to reduce the required backup capacity. Our performance evaluation shows the efficiency of the new approaches depending on load balancing, traffic distribution, resilience constraints, and the network topology.

4.1 Related Work

This work is about routing optimization and load balancing in a very broad sense. First, we point out the difference between destination-based forwarding

and connection-oriented forwarding. Then, we present a well-known categorization of resilient routing schemes. Finally, we give a short overview of routing optimization in general to classify our work.

4.1.1 Routing Paradigms

We review two major forwarding paradigms with respect to their traffic engineering capabilities: destination-based forwarding and connection-oriented forwarding.

Destination-Based Forwarding

In pure Internet Protocol (IP) technology, routers identify the corresponding output interface based on the destination address in the packet header according to their routing tables. The routes in IP forwarding are usually set up by means of routing protocols like the Open Shortest Path First (OSPF) protocol [39]. They exchange reachability information associated with link costs which are used for computing the output ports for the shortest paths to certain destinations. By manipulating the link costs, the routing can be influenced which gives room for traffic engineering. Load balancing over multiple paths is possible if several paths to the same destination have equal costs. This option is called Equal Cost Multi-Path (ECMP) and it is implemented, e.g., in OSPF.

Connection-Oriented Forwarding

MPLS is a connection-oriented switching technology, i.e., traffic is forwarded along virtual connections that build an overlay network (cf. 2.2). Packets matching a set of attributes in a router create a forwarding equivalent class (FEC). A so-called LSP ingress label switching router (LSR) identifies them and groups them together into a single traffic aggregate by assigning the packets a common label on top of their header. This traffic aggregate is forwarded along a label-switched path (LSP) to the LSP egress LSR that pops the label. The intermediate routers of

the LSP forward the packets by label swapping corresponding to the information in their label information base (LIB). The LIB holds a table of incoming LSPs that are identified by their ingress interface and their ingress label and maps them to their egress interface and their egress label. In contrast to routing tables, the information in the LIBs is provided at connection setup. At that occasion, the path of an LSP may be automatically determined by routing protocols or it may follow a pre-computed explicit route.

The routing granularity and the forwarding resolution in MPLS is much finer than in IP because the attributes of a FEC may be, e.g., source *and* destination addresses. Traffic to a same destination may be carried over different paths that have completely different costs by using explicit routes in MPLS. Explicit routing can be mimicked by source routing in IP technology but this is not advisable since it slows down the forwarding speed of routers considerably. In addition, routing along multiple paths is restricted to ECMP. Hence, connection-oriented forwarding technologies like MPLS have a finer control on the data path than destination-based forwarding.

4.1.2 Resilient Routing

In fault-tolerant networks, traffic is deviated over alternative paths in case of a local outage. There are basically two options for resilience mechanisms.

With path restoration in case of MPLS or with rerouting in case of IP technology, a deviation path or route is only established if a failure occurs [230]. Backup capacity can be shared because no resources are bound to any aggregate before a failure occurs. However, the reaction time of restoration mechanisms can be quite long. With path protection the outage is anticipated, i.e., a backup path is set up before a failure occurs. This is also called protection switching. A fast reaction time is one advantage of protection switching compared to restoration mechanisms [231].

The 1:1 protection switching approach sends the traffic over the backup path only if a failure occurs. Thus, the backup capacity can be shared among flows

that do not require the same resources in case of a specific link failure. Hence, the sharing possibilities are the same as with restoration schemes. It is possible to protect The 1+1 approach transmits the traffic simultaneously over a primary and a backup path and, therefore, backup capacity sharing is not possible.

The 1+1 protection switching has the shortest reaction time if a failure occurs because the destination recognizes if a path is down and takes the packet stream from the other path. The 1:1 protection switching requires that the source router detects the failure by the notification of lower layers or by missing "fast keep alive" message of a link management protocol [232, 233]. Then, the transmission of the traffic is redirected from the primary path to the backup path. The overall reaction time is within a few 100ms. The restoration scheme requires in addition the setup of the backup path and IP rerouting needs the flooding of link state messages which can be done within a short time [234]. In standard IP technology, a link failure is detected by missing Hello messages of the OSPF protocol and takes in the order of tens of seconds because the timers are set to a relatively high value [235]. However, modern routers have a higher processing capacity and can handle a larger signaling load, therefore, rerouting can be already achieved within the sub-second range [236, 237]. The shortest path computation is an online algorithm and is executed in the routers if topology changes are signalled and remains finally the bottleneck in IP if the timer problem is solved by notification of lower layer failure detection mechanisms. Multi-topology routing may also be an alternative for short failover times for IP routing [238].

In this work, we choose MPLS technology due to its more powerful routing capabilities. In addition, failover times longer than 200 *ms* are considered critical with respect to voice services [239]. To keep reaction times short, backup paths are established in advance. To save backup capacity, we concentrate on 1:1 protection switching mechanism.

There are several options concerning the scope of a backup path. A primary path can be restored or protected on an end-to-end basis, i.e., there is one backup path for each primary connection. Subpath protection holds a backup path from a possible failure location to the destination and local protection of single links

is another option [240]. Path protection on an end-to-end basis is however most effective regarding backup capacity requirement [241] because the connections that are are affected by a link or node failure can then be deviated far away from the outage location and avoid a concentration of backup traffic for which backup capacity has to be provisioned.

4.1.3 Routing Optimization

Routing optimization is a vast area with different aspects. We briefly give an insight into optimization issues with and without resilience requirements.

Routing Optimization without Resilience Requirements

A well investigated problem is routing optimization in the presence of limited link capacities to maximize the supportable traffic intensity whose b2b structure is given by a traffic matrix [242, 243]. This is a multi-commodity flow problem and its solution can be implemented, e.g., by LSPs [244]. For IP routing, a similar approach can be done by setting the link costs appropriately such that all traffic is transported through the network and that the mean and maximum link utilization is minimized [245, 246, 247, 248, 249, 250]. Pure IP and MPLS solutions may also be combined [251]. This is especially important if the capacity of some network links is increased in response to an increasing traffic demand [252, 253]. In [254] a stable closed loop solution based on multi-path routing is presented to equalize the link utilization for Internet traffic by load balancing mechanisms. Load balancing should be done on a per flow basis and not on a per packet basis to avoid packet reordering which has a detrimental effect on the TCP throughput. The hash-based algorithm in [255, 256] achieves that goal very well.

Routing Optimization with Resilience Requirements

The authors of [257] present an online solution for routing with resilience requirements. They try to minimize the blocking probability of successive path re-

quests using suitable single-paths as primary paths and backup paths. The backup bandwidth may be shared or dedicated. A distributed protocol solution is given in [258]. Another offline optimization algorithm [259] uses a Tabu search heuristic to minimize the overload in an IP network during transient link failures by setting suitable IGP link weights.

Routing with resilience requirements can also be considered under a network dimensioning aspect, i.e., the traffic matrix is given and the link capacities must be set. For example, the sum of link bandwidths in a network should be minimized while only technically available link capacities (e.g., OC3=STM1=155*Mbit/s*, OC12=STM4=622*Mbit/s*, OC48=STM4=2.488*Gbit/s*, OC192=STM64=9.953*Gbit/s*) can be used. Apart from that constraint, this problem is trivial without resilience requirements since a suitable bandwidth assignment for the shortest paths is already an optimum solution. It becomes an optimization problem if capacity sharing is allowed for backup paths. The path layout and the capacity assignment are designed such that primary paths and shared backup paths require minimal network capacity while the backup mechanisms provide full resilience for a given set of protected failure scenarios. This is fundamentally different from the above problem since both the routing and the link bandwidth are optimized simultaneously. Note that the results of such calculations depend on the capabilities of the applied restoration schemes.

The results of [260] can be well implemented since this work applies only single-paths for both primary and backup paths and relocates only affected primary paths. However, they do not make use of multi-path routing and load distribution for path restoration purposes. This is especially important in outage scenarios because traffic diverted over several different paths requires only a fraction of the backup capacity on detour links. If backup capacity sharing is allowed, this backup capacity may be used in different failure scenarios by different rerouted traffic aggregates, which leads to increased resource efficiency since less additional resources must be provisioned in the network. In [261, 241] multi-path routing is used. The required network resources are minimized by calculating the optimum path layout and routing independently for each failure scenario.

These backup solutions are too difficult for implementation but they present lower bounds for the required backup capacity.

4.2 Protection Switching Methods for Backup Capacity Reduction

In this section we derive restrictions for the path layout of protection switching mechanisms. Based on them, we present two novel approaches for backup routing that stand out by their simplicity. We briefly explain how their path layout and their load balancing functions may be computed. Finally, we describe some system constraints for the capacity dimensioning with resilience requirements.

4.2.1 Restrictions for Path Layout of Protection Switching Mechanisms

We explain why the results in [261, 241] cannot be implemented as restoration mechanisms and derive technical side constraints for feasible backup solutions. The path layout and the load balancing is calculated for the normal operation mode and for each failure scenario independently and general multi-path structures are allowed. In an outage case, broken paths must be rerouted but aggregates that are not affected by the failure might also need to be shifted to implement the solution with minimal resources.

Firstly, the knowledge of the exact location of the failure is required to choose the optimized path layout and load balancing. The mere information of a path outage which can be recognized by the sender is not enough. Therefore, the exact outage information must be propagated to all ingress routers to trigger protection switching for a specific outage scenario. This entails extensive signaling in a critical system state where the reachability is corrupted.

Secondly, the relocation of the paths cannot be done simultaneously. Deflect-

ing more paths than necessary might lead to a transient overload on some network elements. This leads to jitter and packet loss within this phase and can be avoided if only broken paths are redirected.

Thirdly, if each connection holds a backup path for each protected failure scenario, a large amount of paths must be pre-installed and administered. This makes the path configuration very complex and the large number of paths is a problem for the state maintenance of today's core network routers.

Finally, to keep the fault diagnostics and the reaction to failures simple, the ingress router should be able to detect a failure and to react locally by switching the traffic to another path. With general multi-path structures, paths may fork and join in transit routers. If a partial path fails, the entire multi-path loses some packets and cannot be used anymore. Implementing general multi-paths as a superposition of overlapping single-paths prevents that problem because only some partial paths may fail in case of a local outage. However, this increases the number of parallel LSPs and makes the state management more complex. Hence, only disjoint paths should be used to achieve simple fault diagnostics for multi-path forwarding.

Another restriction for path layout are Shared Risk Link Groups (SRLGs) [262, 263, 264] which group network elements together that may fail simultaneously with a high probability. For instance, all links originating at the same router fail if the router goes down. SRLGs are motivated by optical networking where a single optical fiber duct accommodates several logically separate links. In our work, we consider only the first scenario and the second one in a trivial way by excluding parallel links over the same duct. However, we do not take general SRLGs into account because our focus is the performance evaluation of the basic Self-Protecting Multi-Path (SPM) and not its adaptation to SRLGs. As the routing and the load balancing computations for our proposed protection switching mechanisms are independent of each other, more elaborate algorithms, that take general SRLGs into account, can be easily integrated to our framework.

4.2.2 Protection Switching Mechanisms Based on Multi-Path Structures

We present several novel protection switching mechanisms that incorporate the idea to reduce backup capacity by multi-path forwarding in failure cases. We can divide them into two distinct approaches: Path Protection (PP) methods and the Self-Protecting Multi-Path (SPM) [265].

Path Protection Mechanisms

Path Protection (PP) mechanisms transport the traffic usually on a primary single-path and use a multi-path structure as a backup path if the primary path fails. This is depicted in Figure 4.2. This is actually an intuitive approach that has been frequently taken in literature before [266]. The novelty is our extension to link and node disjoint multi-paths structures for backup purposes that make the approach easy to implement. Multi-path traffic forwarding in failure cases requires load balancing and gives room for optimization and minimization of required backup capacities.

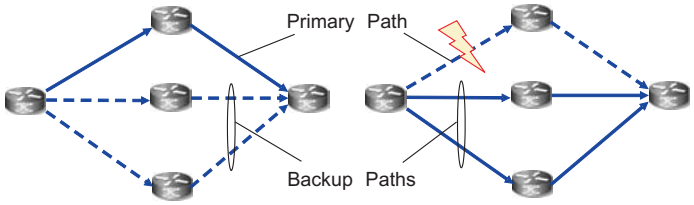


Figure 4.2: *Path Protection using a disjoint multi-path for backup.*

Comparison to P-Cycles The p-cycle approach [267, 268, 269] has been originally applied to the physical layer in WDM and Sonet transport networks but

has also been adapted to the network layer in IP networks using MPLS [270]. A so-called protection cycle is installed in the network like in Figure 4.3. It protects failures of links on the cycle by providing a detour in the counter-direction on the cycle. Hence, the p-cycle provides one disjoint backup path for each link. Paths that touch ("straddle") the p-cycle twice are also protected. Such a path cuts the p-cycle virtually into two parts. If the path fails, the traffic is deviated over these two parts to bridge the outage location. This also allows for load distribution in the failure case. P-Cycles can protect node failures, too, and they have been proven to be quite efficient [271, 272].

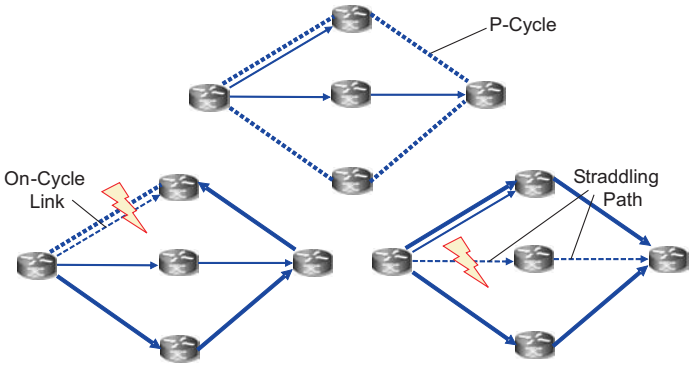


Figure 4.3: *P-Cycles protect on-cycle links and straddling paths.*

We compare the p-cycle approach with PP mechanisms on an abstract level. P-Cycles can be emulated by PP mechanisms with shared backup capacity. For each link on the p-cycle, a backup path on the remaining cycle is installed and for each path straddling a p-cycle twice, two corresponding backup paths are set up over the p-cycle. However, there are also differences. If links or paths are protected by several p-cycles, then the resulting backup paths for PP might not be disjoint. Dropping the requirement for disjoint paths makes PP methods slightly

more complex but does not change them fundamentally.

A drawback of p-cycles is that they have a fixed capacity along their length such that only 50%:50% load balancing can achieved by a single p-cycle. Advanced load balancing can be done by overlapping p-cycles. This makes the overall p-cycle layout more difficult. In addition, the capacity is bound quite strictly on the physical layer. For example, two p-cycles sharing a common link cannot share their backup capacity, and the capacity of failed primary paths cannot be reused for restoration purposes. As the capacity of p-cycles is bound to backup objectives only, the overlay network of p-cycles presents a mere restoration layer. There is a tradeoff depending on the length of a p-cycle. Straddling links are protected more efficiently and their number grows with the length of a p-cycle. However, the maximum required capacity must be provided along the whole cycle. Thus, the cycle length increases the total sum of reserved protection capacity [273].

Self-Protecting Multi-Path

The Self-Protecting Multi-Path (SPM) is a completely novel approach. Figure 4.4 shows that it consists of disjoint partial paths. In contrast to PP, the traffic is distributed onto all paths in the non-failure case, too. If a partial path fails, the traffic is redistributed onto the working paths by a *path failure* specific load distribution function. The SPM has more degrees of freedom than PP solutions. It can emulate PP mechanisms by applying suitable load distribution functions and it has, therefore, more optimization potential than PP methods.

4.2.3 Computation of Path Layout and Load Balancing

There are many options for the path layout and the load balancing for PP mechanisms and the SPM. We give a short overview in this section before we present detailed heuristics and optimization algorithms in the next section.

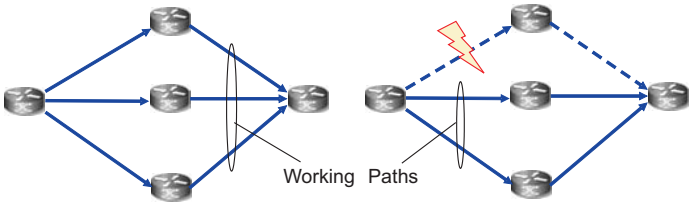


Figure 4.4: *The Self-Protecting Multi-Path uses always all working partial paths.*

Design of PP Mechanisms

In the next section, we derive a description of an integrated solution for an optimum path layout and load balancing of PP mechanisms. This contains, however, quadratical equations that require integer solutions. Since this is not computationally feasible, we propose heuristics for primary and backup path computation as well as for load balancing in outage scenarios.

Primary Path Calculation The paths for each source – destination pair are found independently of each other. We propose two different solutions to calculate the primary path.

- The kDSP algorithm [274, 275] computes up to k disjoint shortest paths in a network and we extend that algorithm to link and node disjointness. We take the shortest of k disjoint shortest paths as primary path. This guarantees that $k - 1$ link and node disjoint backup paths can be found afterwards if they are topologically feasible.
- Another routing approach is minimum traffic (MT) routing, i.e., the primary paths are chosen in such a way that the maximum traffic rate traversing a node is kept small. If a router fails, only a moderate traffic volume must be deviated from the outage location. This requires the solution of a linear program (LP).

Backup Path Calculation We propose also two different methods to calculate the path layout for the backup structure.

- The fixed primary path reduces the quadratical conditions for calculation of the optimum path layout to linear constraints. Therefore, an optimum multi-path backup structure (OPT) can be computed together with a load balancing function for each node contained in the multi-path by a general, computation intensive linear program (LP). Since this solution yields still non-disjoint multi-path backup structures, it is suitable for comparison purposes to assess the optimality of other approaches but it is not recommended for implementation because partial paths may fork and join.

- The preferred solution for practical applications is taking the paths from a $(k-1)$DSP calculation like above because this yields a multi-path consisting of disjoint partial paths. The computation is based on the original network which is reduced by the intermediate elements of the primary path. If the primary path has also been calculated by a kDSP approach, the maximum possible number of disjoint backup paths can be found. Primary path selection by MT routing can prohibit the existence of a disjoint backup path although a pair of disjoint primary and backup paths are topologically feasible.

Calculation of the Load Balancing Function We compute different load balancing functions for traffic forwarding over disjoint multi-paths in the failure case.

- Equal (E) load balancing is very simple. The traffic of a b2b aggregate is equally distributed over all working paths to distribute the traffic.

- A simple optimization approach is the assignment of a large portion of the b2b traffic aggregate to short partial paths and of a small portion to long partial paths. Mathematically speaking, we distribute the rate of a traffic

aggregate onto the working paths reciprocally (R) to the lengths of these paths.

- The load balancing functions can also be exactly optimized (O) by taking all b2b traffic aggregate and their routing into account. This is again done by a non-integer LP.

Design of SPM mechanisms

As all partial paths of an SPM are equal, we construct its path of up to k paths with a kDSP computation. The load distribution functions can be computed like for PP mechanisms according to the options E, R, and O.

4.2.4 System Constraints for Capacity Dimensioning with Resilience Requirements

Network resilience is a soft expression as it means fault tolerance against a set of faulty networking scenarios that adhere to some assumptions. We present them in the following.

Protected Failure Scenarios

The optimization of protection switching mechanisms requires a set of protected failure scenarios $\mathcal{S}$ which contains by default the working scenario. We consider three different options. “Link protection” takes only all single bidirectional link failures into account, “router protection” respects only single router failures, and we call the consideration of both single bidirectional link and router failures “full protection”.

Traffic Reduction

During normal operation without any failure, all b2b aggregates are active. If ingress or egress routers fail, some traffic may disappear. We consider several options. If network nodes lose only their capability to transport transit flows but if they are still able to generate traffic, then we speak of “no traffic reduction”. If failed nodes stop sending traffic, we talk about “source traffic reduction”. With “full traffic reduction” the traffic is stalled if either its source or destination node does not work.

Bandwidth Reuse

In packet-switched networks, resources are not physically dedicated to any flows. If traffic is rerouted due to a local outage, the resources can be automatically reused for transporting other traffic. Hence, “bandwidth reuse” is possible. In optical networks, connections are bound to physical resources like fibers, wavelengths, or time slots. If a network element fails, there might not be enough time to free the resource of a redirected connection. This is the “no bandwidth reuse” option because network resources allocated by failed paths cannot be reused for backup purposes.

4.3 Optimization

Our objective is the design of the routing and the protection switching including load balancing in such a way that the required capacity is minimized. In this section we formulate the optimization problem by linear equations [276]. We describe the problem solutions as linear programs (LPs). It is a standard technique [277] that can be solved by software like *ILOG CPlex* [278] or the *GNU Linear Programming Kit* [279]. We adapt this formulation to various protection mechanisms.

4.3.1 Optimum Primary and Backup Path Solution

We explain some basic notation from linear algebra and choose it for the representation of links and nodes. We introduce the traffic matrix, and embed paths and flows into that framework. We rewrite the set of protected scenarios and suggest how to handle traffic reduction by failed border routers. The failure indication function decides whether a path is affected by the failure of a specific network element because only path failures can be diagnosed in a robust way. The backup path can be structured such that either only link failures or both link and node failures can be protected. The objective function calculates the required overall bandwidth in the network and respects capacity constraints regarding bandwidth reuse. It should be minimized. Finally, we summarize how the optimal solution is calculated and point out why it is not feasible.

Basic Notation

Let $\mathbb{X}$ be a set of elements, then $\mathbb{X}^n$ is the set of all n-dimensional vectors and $\mathbb{X}^{n \times m}$ the set of all $n \times m$-matrices with components taken from $\mathbb{X}$. Vectors $\mathbf{x} \in \mathbb{X}^n$ and matrices $\mathbf{X} \in \mathbb{X}^{n \times m}$ are written bold and their components are written as $\mathbf{x} = \begin{pmatrix} x_0 \\ \cdot \\ x_{n-1} \end{pmatrix}$ and $\mathbf{X} = \begin{pmatrix} x_{0,0} & \cdots & x_{0,m-1} \\ \cdot & & \cdot \\ x_{n-1,0} & \cdots & x_{n-1,m-1} \end{pmatrix}$. The scalar multiplication $c \cdot \mathbf{v}$ and the transpose operator $^\top$ are defined as usual. The scalar product of two n-dimensional vectors $\mathbf{u}$ and $\mathbf{v}$ is written with the help of a matrix multiplication

$\mathbf{u}^\top \mathbf{v} = \sum_{i=1}^{n} u_i \cdot v_i$. Binary operators $\circ \in \{+, -, \cdot\}$ are applied componentwise, i.e. $\mathbf{u} \circ \mathbf{v} = (u_0 \circ v_0, \ldots, u_{n-1} \circ v_{n-1})^\top$. The same holds for relational operators $\circ \in \{<, \leq, =, \geq, >\}$, i.e., $\mathbf{u} \circ \mathbf{v}$ equals $\forall\ 0 \leq i < n : u_i \circ v_i$. For reasons of simplicity, we define special vectors $\mathbf{0} = (0, \ldots, 0)^\top$ and $\mathbf{1} = (1, \ldots, 1)^\top$ with context-specific dimensions.

Links and Nodes

The network $\mathcal{N} = (\mathcal{V}, \mathcal{E})$ consists of $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ unidirectional links that are represented as unit vectors $\mathbf{v_i} \in \{0,1\}^n$ and $\mathbf{e_i} \in \{0,1\}^m$, i.e.

$$(v_i)_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad \text{for } 0 \leq i, j < n \quad \text{and} \tag{4.1}$$

$$(e_i)_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad \text{for } 0 \leq i, j < m. \tag{4.2}$$

The links are directed and the operators $\alpha(e_i)$ and $\omega(e_i)$ yield the sending and the receiving router of a link. The outgoing and incoming incidence matrices $\mathbf{A}_\alpha$ and $\mathbf{A}_\omega$ describe the network connectivity, i.e.

$$(a_\alpha)_{ij} = \begin{cases} 0 & \alpha(e_j) \neq v_i \\ 1 & \alpha(e_j) = v_i \end{cases} \quad \text{and} \tag{4.3}$$

$$(a_\omega)_{ij} = \begin{cases} 0 & \omega(e_j) \neq v_i \\ 1 & \omega(e_j) = v_i \end{cases}. \tag{4.4}$$

The incidence matrix $\mathbf{A} \in \{-1, 0, 1\}^{n \times m}$ is defined as $\mathbf{A} = \mathbf{A}_\omega - \mathbf{A}_\alpha$. The j-th column of $\mathbf{A}$ indicates the source and target of link e_j. The vector $\mathbf{A}\mathbf{e_j}$ yields a node vector. It has a –1 in the i-th row if the source node of e_j is v_i, it has a 1 in the i-th row if the target node of e_j is v_i, and there are zeroes in all other positions. The j-th row of $\mathbf{A}$ indicates the outgoing and incoming links of node v_j. The link vector $\mathbf{v_j}^\top \mathbf{A}$ has a –1 for all outgoing links, a 1 for all incoming links, and zeroes in all other positions. Loops cannot be expressed by this formalism.

Traffic Matrix, Paths, and Flows

We introduce the traffic matrix, define a path for a b2b aggregate and enhance it to a flow.

Traffic Matrix The aggregate of all flows from an ingress router $\mathbf{v_i}$ to an egress router $\mathbf{v_j}$ is denoted by the b2b aggregate g_{v_i,v_j}. All b2b aggregates compose the set $\mathcal{G}$. The associated traffic rate for a b2b aggregate $g \in \mathcal{G}$ is given by $c(g)$ and corresponds to an entry in the traffic matrix.

Paths A path p_g of an aggregate $g \in \mathcal{G}$ between distinct nodes v_α and v_ω is a set of contiguous links represented by a link vector $\mathbf{p_g} \in \{0,1\}^m$. This corresponds to a single-path. However, we usually apply the concept of a multi-path $\mathbf{p_g} \in [0,1]^m$, which is more general since the traffic may be split into several partial paths carrying a real fraction of the traffic. A path follows conservation rules, i.e., the amount of incoming traffic equals the amount of outgoing traffic in a node which is expressed by

$$\mathbf{A}\mathbf{p_g} = (\mathbf{v}_\omega - \mathbf{v}_\alpha). \tag{4.5}$$

While cycles containing only inner nodes can be easily removed, cycles containing the start or end node of a path are more problematic. Therefore, we formulate a condition preventing this case. The expressions $\mathbf{v}_\alpha{}^\top \mathbf{A}_\omega$ and $\mathbf{v}_\omega{}^\top \mathbf{A}_\alpha$ yield the incoming edges of start node v_α and all outgoing edges of end node v_ω of a path p_g. Hence, cycles containing the start or end node can be prevented if the following equations hold:

$$(\mathbf{v}_\alpha{}^\top \mathbf{A}_\omega)\mathbf{p_g} = 0 \quad \text{and} \quad (\mathbf{v}_\omega{}^\top \mathbf{A}_\alpha)\mathbf{p_g} = 0. \tag{4.6}$$

Flows The mere path of an aggregate $g \in \mathcal{G}$ is $\mathbf{p_g}$. We get the corresponding flow by a scalar multiplication $c(g) \cdot \mathbf{p_g}$ to take the rate of the aggregate into account.

Protected Scenarios

A protected failure scenario is given by a vector of failed nodes $\mathbf{s}_\mathcal{V} \in \{0,1\}^n$ and a vector of failed links $\mathbf{s}_\mathcal{E} \in \{0,1\}^m$. We denote a failure scenario shortly by $\mathbf{s} = \left(\begin{smallmatrix}\mathbf{s}_\mathcal{V}\\ \mathbf{s}_\mathcal{E}\end{smallmatrix}\right)$. The set $\mathcal{S}$ contains all protected outage scenarios including $\mathbf{s} = \mathbf{0}$, i.e. the no failure case.

Traffic Reduction

During normal operation without any failure, all aggregates $g \in \mathcal{G}$ are active. If routers fail, some may disappear. We consider several options.

No Traffic Reduction We assume that failed routers lose only their transport capability for transit flows but are still able to generate traffic. Therefore, we have $\mathcal{G}_\mathbf{s} = \mathcal{G}$.

Source Traffic Reduction An aggregate flow is removed from the traffic matrix if the source node v_i of aggregate g_{v_i,v_j} fails. If a failed node is the destination of a flow, "server push" traffic may still be transported through the network, hence

$$\mathcal{G}_\mathbf{s} = \mathcal{G} \setminus \{g_{v_i,v_j} : \mathbf{v_i}^\top \mathbf{s}_\mathcal{V} = 1, 1 \le j \le n, i \ne j\}. \tag{4.7}$$

Full Traffic Reduction In contrast to above we assume that the traffic with a failed destination is stalled. An aggregate flow is removed from the traffic matrix if a node fails which is either the source or the destination of a flow, hence

$$\begin{aligned}\mathcal{G}_\mathbf{s} = \mathcal{G} \setminus \big(&\{g_{v_i,v_j} : \mathbf{v_i}^\top \mathbf{s}_\mathcal{V} = 1, 1 \le j \le n, i \ne j\} \cup &(4.8)\\ &\{g_{v_i,v_j} : \mathbf{v_j}^\top \mathbf{s}_\mathcal{V} = 1, 1 \le i \le n, i \ne j\}\big). &(4.9)\end{aligned}$$

Failure Indication Function

The failure indication function $\phi(\mathbf{p}, \mathbf{s})$ indicates whether a path p is affected by a failure scenario $\mathbf{s}$ [280]. Path p is affected by a link failure scenario $\mathbf{s}_\mathcal{E}$ if ${\mathbf{s}_\mathcal{E}}^\top \mathbf{p} > 0$. To formulate this analogously for node failures we define traces. The α-trace is $\mathbf{tr}_\alpha(\mathbf{p_g}) = \mathbf{A}_\alpha \mathbf{p_g}$ and the ω-trace is $\mathbf{tr}_\omega(\mathbf{p_g}) = \mathbf{A}_\omega \mathbf{p_g}$, respectively. We obtain the interior trace $\mathbf{ti}$ by excluding the corresponding end or the start node of the α- or ω-trace, respectively, i.e. $\mathbf{ti}(\mathbf{p_g}) = \mathbf{A}_\alpha \mathbf{p_g} - \mathbf{v}_\alpha = \mathbf{A}_\omega \mathbf{p_g} - \mathbf{v}_\omega$. Path p is affected by a node failure scenario $\mathbf{s}_\mathcal{V}$ if ${\mathbf{s}_\mathcal{V}}^\top \mathbf{ti}(\mathbf{p}) > 0$. Finally, the failure indication function is

$$\phi(\mathbf{p}, \mathbf{s}) \quad = \quad \begin{cases} 1 & {\mathbf{s}_\mathcal{E}}^\top \mathbf{p} + {\mathbf{s}_\mathcal{V}}^\top \mathbf{ti}(\mathbf{p}) > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

Protection Alternatives

A path restoration scheme introduces a backup path q_g which is activated if the primary path fails. This backup path protects against link and/or node failures of each primary path p_g depending on the required type of resilience. A backup path q_g is link protecting if

$${\mathbf{q_g}}^\top \mathbf{p_g} = 0 \tag{4.11}$$

and it is both link and node protecting if the following holds

$$\mathbf{ti}(\mathbf{q_g})^\top \mathbf{ti}(\mathbf{p_g}) = 0. \tag{4.12}$$

Objective Function and Capacity Constraints

We describe the capacity of all links by a vector of edges $\mathbf{b} \in (\mathbb{R}_0^+)^m$. The overall capacity in the network is the objective function that is to be minimized. It can be computed by

$$\mathbf{w}^\top \mathbf{b} \to \min \tag{4.13}$$

where $\mathbf{w} \in (\mathbb{R}_0^+)^m$ is a vector of weights, that is normally set to $\mathbf{w} = \mathbf{1}$. If the connectivity is maintained by a backup path in case of a failure scenario $\mathbf{s} \in \mathcal{S}$, the following bandwidth constraints guarantee that enough capacity is available to carry the traffic generated by the aggregates $g \in \mathcal{G}_\mathbf{s}$.

Bandwidth Reuse In pure packet-switched networks, resources are not physically dedicated to any flows. If traffic is rerouted due to an outage, the resources can be automatically reused for transporting other traffic. Under this assumption, the capacity constraints are

$$\forall \mathbf{s} \in \mathcal{S} : \sum_{g \in \mathcal{G}_\mathbf{s}} c(g) \cdot ((1 - \phi(\mathbf{p_g}, \mathbf{s})) \cdot \mathbf{p_g} + \phi(\mathbf{p_g}, \mathbf{s}) \cdot \mathbf{q_g}) \leq \mathbf{b}. \tag{4.14}$$

No Bandwidth Reuse In optical networks, physical resources like fibers, wavelengths, or time slots are bound to connections. If a network element fails, there might not be enough time to free the resources of a redirected connection. This is respected by the following capacity constraints:

$$\forall \mathbf{s} \in \mathcal{S} : \sum_{g \in \mathcal{G}} c(g) \cdot \mathbf{p_g} + \sum_{g \in \mathcal{G}_\mathbf{s}} c(g) \cdot \phi(\mathbf{p_g}, \mathbf{s}) \cdot \mathbf{q_g} \leq \mathbf{b}. \tag{4.15}$$

Optimal Solution Summary

We summarize the above derived formalism. The free variables to be set by the optimization are

$$\mathbf{b} \in (\mathbb{R}_0^+)^m \text{ and } \forall g \in \mathcal{G} : \mathbf{p_g}, \mathbf{q_g} \in [0,1]^m. \tag{4.16}$$

Both the primary paths $\mathbf{p_g}$ and the backup paths $\mathbf{q_g}$ conform to the conservation rule Equation (4.5) and exclude start and end nodes explicitly from cycles by Equation (4.6). The protection of path $\mathbf{p_g}$ is achieved if the backup path $\mathbf{q_g}$ respects either Equation (4.11) or Equation (4.12) for link protection or for link and node protection, respectively. The capacity constraints have to be met either

with or without bandwidth reuse (Equation (4.14) and Equation (4.15)). The objective function in Equation (4.13) is to be minimized while all these constraints are taken into account.

Unfortunately, the path protection constraints (Equation (4.11) and Equation (4.12)) are quadratic with respect to the free variables. Therefore, this description cannot be solved by LP solvers. In addition, the failure indication function $\phi(\mathbf{p}, \mathbf{s})$ cannot be transformed into a linear mapping. Thus, we have no efficient algorithm to compute the desired structures $\mathbf{p_g}$ and $\mathbf{q_g}$. If the complexity of the primary and backup multi-paths is restricted, e.g. to single-paths, the computation becomes more difficult due to a required integer solution for $\mathbf{p_g}$ and $\mathbf{q_g}$. The modelling of disjoint multi-paths solutions is even more difficult. Therefore, we use heuristics in the following.

4.3.2 Heuristics for Path Calculation

Due to the computational problems and due to the difficulty of controlling the structure of multi-paths we propose to calculate first a suitable path layout and then to derive a suitable load balancing function. We propose to calculate a link and node disjoint multi-path structure by using an algorithm to compute the k disjoint shortest paths (kDSP). We propose another heuristic that tries to place the primary path in a preferred way for PP methods. If a primary path is given, the kDSP algorithm may be used for the computation of a link and node disjoint multi-path for backup purposes. Another option is the computation of an optimal path layout together with a load balancing function. This method yields a general multi-path and is, therefore, not suitable in practice.

The k Disjoint Shortest Path Algorithm

Both the PP method and the SPM approach require disjoint multi-paths for their path layout. A very simple solution to get a set of disjoint paths is taking the shortest path $\mathbf{p}$ which can be found by Dijkstra's algorithm [38], removing its interior nodes $\mathbf{ti}(\mathbf{p})$ and links $\mathbf{tr}(\mathbf{p})$ from the network and running Dijkstra's

algorithm again. However, this procedure does not always find k disjoint paths in the network although they might be topologically feasible [281]. Figure 4.5 shows how a first path can prohibit the existence of another link and node disjoint path in the network.

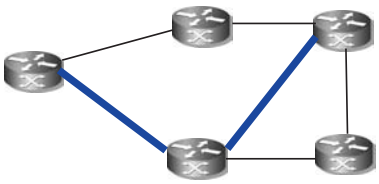


Figure 4.5: *The primary path prohibits the existence of a node and link disjoint backup path.*

In contrast to online solutions [282, 283], the k Disjoint Shortest Path (kDSP) offline algorithm [274, 275, 284, 285, 277] finds always up to k disjoint shortest path in a network if they exist These paths may be taken as the equal paths of an SPM. If they are taken for the layout of PP mechanisms, the shortest one of them should become the primary path and the other paths constitute the multi-path for backup purposes.

Primary Path Computation: Minimum Traffic (MT) Routing

With PP, the primary path plays a distinguished role. If a network element carries a large amount of traffic and fails, this traffic has to be redistributed and requires a lot of backup capacity near the outage location. Therefore, we construct a path layout that entails a minimum traffic load on each network element.

Minimum Traffic Constraints The overall traffic on all links is given by the auxiliary vector $\mathbf{a}^{\mathbf{E}} \in (\mathbb{R}_0^+)^m$ and the overall traffic in all nodes is given by

the auxiliary vector $\mathbf{a^V} \in (\mathbb{R}_0^+)^n$, respectively.

$$\mathbf{a^E} = \sum_{g \in \mathcal{G}} f(g) \cdot \mathbf{p_g} \qquad \text{and} \qquad \mathbf{a^V} = \sum_{g \in \mathcal{G}} f(g) \cdot \mathbf{ti}(\mathbf{p_g}) \tag{4.17}$$

$$\mathbf{a^E} \leq a^E_{max} \cdot \mathbf{1} \qquad \text{and} \qquad \mathbf{a^V} \leq a^V_{max} \cdot \mathbf{1}. \tag{4.18}$$

The value $f(g)$ may be set to 1 if only the number of aggregates is to be minimized or it may be set to $c(g)$ if their rate should be taken into account. We use $f(g) = c(g)$ in this study.

Objective Function Both the maximum traffic per network element (a^E_{max} or a^V_{max}) and the overall capacity ($\mathbf{1}^\top \mathbf{a^E}$ or $\mathbf{1}^\top \mathbf{a^V}$) should be minimized but they represent potentially conflicting goals. To avoid very long paths, the objective function takes also the overall required capacity $\mathbf{1}^\top \mathbf{a^Y}$ into account:

$$M^X \cdot a^X_{max} + \mathbf{1}^\top \mathbf{a^Y} \rightarrow \min. \tag{4.19}$$

The constants $M^E, M^V \in \mathbb{R}_0^+$ control the tradeoff between the conflicting goals. A small M^X favors little overall capacity while a large M^X favors little maximum traffic per network element. In our experiments, we set $X = V$ and $Y = V$.

Path Constraints Like above, the flow conservation rule (Equation (4.5)) and the exclusion of start and end nodes from cycles (Equation (4.6)) have to be respected. Since we are interested in single-path solutions, $\mathbf{p_g} \in \{0,1\}^m$ is required. This, however, leads to a mixed integer LP that takes a long computation time.

Therefore, we relax this condition to $\mathbf{p_g} \in [0,1]^m$ to get a non-integer LP. To obtain a desired single-path as primary path, we decompose the general multi-path into single-paths and a load balancing function. Then, we take the single-path of the calculated multi-path structure with the largest load balancing value. Note that this decomposition is not unique and various results can be obtained depending on the implementation. This is very similar to the computation of a single-shortest path.

Backup Path Computation with kDSP

A set of disjoint single-paths is required to build a backup path for a given primary path $\mathbf{p_g}$. They can be obtained using the kDSP algorithm. For that objective, we first reduce the links $\mathbf{tr}(\mathbf{p_g})$ contained in the primary path from the network. If the backup path should be both link and node disjoint with the primary path, we also remove the interior nodes $\mathbf{ti}(\mathbf{p_g})$. Then, we run the kDSP algorithm on the remaining network and the result provides the resulting structure of the backup path. If the primary path has not been found by the kDSP algorithm, a link and node disjoint backup path cannot always be found although two disjoint paths may exist in the network.

Computation of an Optimum Backup Path

If a primary path is given, the optimum backup path together with the corresponding load balancing function can be obtained by a slight modification of the LP formulation in Section 4.3.1. As $\mathbf{p_g}$ is already fixed, we remove it from the set of free variables. Then, the quadratic conditions in terms of free variables in Equations (4.11) and (4.12) disappear. In addition, the failure indication function $\phi(\mathbf{p_g}, \mathbf{s})$ is independent of any free variables. Therefore, this modification yields an LP formulation which can be solved efficiently. The so obtained backup path structure may have circles that do not increase the required capacity. When this path layout is configured in a real system, these circles must be removed. We omit the corresponding elementary graph-theoretical operations, which are simple because the source and destination nodes are prevented to be part of a circle (cf. Equation (4.6)). However, the structure of the resulting backup path is potentially still very complex since the partial b2b paths are not necessarily disjoint and, therefore, this method is rather intended for comparison purposes and not in practice.

Adaptation to SRLGs

For the computation of disjoint multi-paths we use the kDSP algorithm which is simple and efficient to compute. However, it does not take general SRLGs into account which is a different and NP hard problem. Basically, our kDSP heuristic can be substituted by any other routing scheme yielding disjoint multi-paths.

4.3.3 Computation of the Load Balancing Function

If the path layout for a SPM or a PP mechanism is given, a suitable load balancing function is required. We first present some basics for failure-dependent load balancing and then derive three different load balancing mechanisms for SPM. Finally, we adapt them to PP mechanisms.

Basics for Failure-Dependent Load Balancing

An SPM consists of k_g link and (not necessarily) node disjoint paths (except for source and destination) $\mathbf{p_g^i}$ for $0 \leq i < k_g$ that may be found, e.g., by a kDSP solution. It is represented by a vector of single-paths $\mathbf{P_g} = (\mathbf{p_g^0}, ..., \mathbf{p_g^{k_g-1}})^\top$. These paths are equal in the sense that they all may be active without any network failure.

Path Failure Pattern $\mathbf{f_g(s)}$ We define the path failure pattern $\mathbf{f_g(s)} \in \{0,1\}^{k_g}$ that indicates the failed partial paths of the SPM for g depending on the failure scenario $\mathbf{s}$. It is computed by

$$\mathbf{f_g(s)} = \left(\phi(\mathbf{p_g^0}, \mathbf{s}), ..., \phi(\mathbf{p_g^{k_g-1}}, \mathbf{s})\right)^\top. \tag{4.20}$$

With a path failure pattern of $\mathbf{f_g} = \mathbf{0}$ all paths are working while for $\mathbf{f_g} = \mathbf{1}$ connectivity cannot be maintained. The set of all different failures for SPM $\mathbf{P_g}$ is denoted by $\mathcal{F}_g = \{\mathbf{f_g(s)} : \mathbf{s} \in \mathcal{S}\}$.

Load Balancing Function $\mathbf{l_g(f)}$ For all aggregates $g \in \mathcal{G}$, a load balancing function $\mathbf{l_g(f)} \in (\mathbb{R}_0^+)^{k_g}$ must be found whose arguments are path failure patterns $\mathbf{f} \in \mathcal{F}_g$. They have to suffice the following restriction:

$$\mathbf{1}^\top \mathbf{l_g(f)} = 1. \tag{4.21}$$

Furthermore, failed paths must not be used, i.e.

$$\mathbf{f}^\top \mathbf{l_g(f)} = 0. \tag{4.22}$$

Finally, the vector indicating the transported traffic of aggregate g over all links is calculated by $\mathbf{P_g}^\top \mathbf{l_g(f)} \cdot c(g)$.

Equal Load Balancing

The traffic may be distributed equally over all working paths, i.e.

$$\mathbf{l_g(f)} \quad = \quad \frac{1}{\mathbf{1}^\top(\mathbf{1}-\mathbf{f})} \cdot (\mathbf{1}-\mathbf{f}). \tag{4.23}$$

Reciprocal Load Balancing

The load balancing factors may be indirectly proportional to the length of the partial paths ($\mathbf{1}^\top \mathbf{p}$). They can be computed for all partial paths.

$$(l_g(f))_i \quad = \quad \frac{\frac{1-f_i}{\mathbf{1}^\top(\mathbf{P_g})_\mathbf{i}}}{\sum_{0 \le j < k_g} \frac{1-f_j}{\mathbf{1}^\top(\mathbf{P_g})_\mathbf{j}}} \text{ for } 0 \le i < k_g \tag{4.24}$$

Optimized Load Balancing

Load balancing is optimal if the required capacity $\mathbf{b}$ to protect all aggregates $g \in \mathcal{G}$ in all protected failure scenarios $\mathbf{s} \in \mathcal{S}$ is minimal. We formulate a LP to describe the solution. The free variables are

$$\mathbf{b} \in (\mathbb{R}_0^+)^m, \quad \forall g \in \mathcal{G}\ \forall \mathbf{f} \in \mathcal{F}_g : \mathbf{l_g(f)} \in (\mathbb{R}_0^+)^{k_g}. \tag{4.25}$$

The objective function is given by Equation (4.13). The load balancing constraints in Equations (4.21) and (4.22) must be respected by all $\mathbf{l_g(f)}$ and the bandwidth constraints are newly formulated.

Bandwidth Constraints with Capacity Reuse The capacity vector $\mathbf{b}$ must be large enough to accommodate the traffic in all protected failure scenarios $\mathbf{s} \in \mathcal{S}$:

$$\forall \mathbf{s} \in \mathcal{S} : \sum_{g \in \mathcal{G}_\mathbf{s}} \mathbf{P_g}^\top \mathbf{l_g}(\mathbf{f_g}(\mathbf{s})) \cdot c(g) \leq \mathbf{b}. \tag{4.26}$$

Bandwidth Constraints without Capacity Reuse Releasing capacity unnecessarily leads to a waste of bandwidth if it cannot be reused by other connections. Therefore, load balancing factors $\mathbf{l_g(f)}$ of active paths must only increase in an outage scenario, except for failed paths for which they are zero. This quasi monotonicity can be expressed by

$$\forall \mathbf{f} \in \mathcal{F}_g : \mathbf{l_g(f)} + \mathbf{f} \geq \mathbf{l_g}(\mathbf{f_g}(\mathbf{0})), \tag{4.27}$$

where $\mathbf{l_g}(\mathbf{f_g}(\mathbf{0}))$ is the load balancing function without failures. The vector of required link capacities $\mathbf{b}$ must meet the bandwidth requirements in all protected failure scenarios $\mathbf{s} \in \mathcal{S}$. They consist of three summands: (1) the capacity used for active aggregates $\mathcal{G}_\mathbf{s}$, (2) the unreleased capacity of failed paths of active aggregates $\mathcal{G}_\mathbf{s}$, and (3) the unreleased capacity of path of aggregates $\mathcal{G} \setminus \mathcal{G}_\mathbf{s}$ that

are removed due to a router failure. Thus, the bandwidth constraints are

$$\forall \mathbf{s} \in \mathcal{S}: \quad \underbrace{\sum_{g \in \mathcal{G}_\mathbf{s}} c(g) \cdot \mathbf{P_g}^\top \mathbf{l_g}(\mathbf{f_g}(\mathbf{s}))}_{\text{(1) used capacity}} + \underbrace{\sum_{g \in \mathcal{G}_\mathbf{s}} c(g) \cdot \mathbf{P_g}^\top (\mathbf{f_g}(\mathbf{s}) \cdot \mathbf{l_g}(\mathbf{f_g}(\mathbf{0})))}_{\text{(2) inactive partial paths}} + \underbrace{\sum_{g \in \mathcal{G} \setminus \mathcal{G}_\mathbf{s}} c(g) \cdot \mathbf{P_g}^\top \mathbf{l_g}(\mathbf{f_g}(\mathbf{0}))}_{\text{(3) removed aggregates}} \leq \mathbf{b}. \tag{4.28}$$

Note that the term $\mathbf{f_g}(\mathbf{s}) \cdot \mathbf{l_g}(\mathbf{f}(\mathbf{0}))$ expresses an element-wise multiplication of two vectors. Hence, if bandwidth reuse is possible, Equation (4.26) is used as bandwidth constraint, otherwise Equations (4.27) and (4.28) must be respected. Neither protection constraints (Equations (4.11) and (4.12)) nor path constraints (Equations (4.5) and (4.6)) apply since the structure of the path is already fixed.

Adaptation to Path Protection

The adaptation of the above explained load balancing scheme to PP mechanisms is simple. We denote the primary paths p_g together with its disjoint backup single-paths as SPM $\mathbf{P_g}$ with $\mathbf{p_g} = (\mathbf{P_g})_0$. The essential difference between the path protection scheme and the SPM is the path failure pattern if the primary path is working. For path protection schemes, the path failure pattern $\mathbf{f_g^{PP}}(\mathbf{s})$ is described by

$$\mathbf{f_g^{PP}}(\mathbf{s}) = \begin{cases} \mathbf{u^0} & \phi(\mathbf{p_g}, \mathbf{s}) = 0 \\ \mathbf{f_g}(\mathbf{s}) & \phi(\mathbf{p_g}, \mathbf{s}) = 1 \end{cases} \tag{4.29}$$

with $\mathbf{u^0} = (0, 1, ..., 1)^\top$. By substituting the path failure pattern in Equation (4.20) by Equation (4.29), the load balancing optimization in Section 4.3.3 can be applied to PP schemes.

4.4 Performance Evaluation of Resilient Routing

In this chapter, we evaluate the performance of the above discussed protection switching methods. Our evaluation methodology is the following. We determine the required network capacity if shortest path routing is used based on the hop count metric. That is the sum of all link bandwidths that are required to accommodate the traffic matrix without any resilience requirements. We take it as a reference value since it is a lower bound for the required network capacity. Then we calculate the capacity for a given protection scheme that is required to meet the resilience requirements. The resulting extra capacity in percent is the performance measure in our studies. Note that this extra capacity is not always used for backup purposes only, because protection mechanisms sometimes require longer paths than the shortest one in normal operation. However, we use the term extra capacity and backup capacity exchangeably since the extra capacity is required to provide resilience with the respective protection mechanism.

First, we compare the performance of the different protection schemes that have been presented above. Then, we investigate the SPM in more detail since it proves to be the most viable solution for protection switching.

4.4.1 Performance Comparison of Different Protection Switching Mechanisms

In this section, we give first a summary of the various protection switching methods and introduce some abbreviations. Then, we evaluate the impact of multipath routing and load balancing on their required backup capacity. We study their backup performance in different example networks and investigate the influence of the traffic matrix. In the end, we compare the backup performance of these methods with the one of the p-cycles in the same experimental environment as in [273, 272].

Overview of Considered Protection Switching Mechanisms

In Section 4.2.2 we have presented novel protection switching mechanisms that we briefly recapitulate for the reader's convenience.

Path Protection With Path Protection (PP), a backup multi-path protects a primary single-path. The primary path may be determined by a k-disjoint shortest paths solution (kDSP) or by minimum traffic (MT) routing. The backup path can be calculated by a $(k-1)$DSP solution. In that case, a load balancing scheme is needed. The load may be balanced equally over all parallel paths (E), reciprocally to the length of the disjoint parallel paths (R), or according to an optimized solution computed by an LP (O). An optimum backup multi-path may be also computed jointly with an appropriate load balancing function by an LP optimization (OPT) which does not necessarily yield disjoint paths for the multi-path and is rather suited for comparison purposes.

Self-Protection Multi-Path We calculate the paths of an SPM by a kSPM solution. The kSPM can differentiate many path failure symptoms (including the normal operation) and requires for each of them a load balancing function. They may be also configured like above (E, R, O).

Abbreviations We abbreviate the PP mechanisms by the methods used to find their primary path, their backup path, and to determine their load balancing and the same holds analogously for the SPM. For example, 5DSP-4DSP-R means that the single primary path is chosen as the shortest from a 5DSP solution and the other (at most) 4 are taken for backup purposes. Load balancing is done reciprocally to the respective path lengths. With MT-OPT the primary path is found by an MT routing solution and the backup multi-path together with a load balancing scheme is computed by an LP. Finally, 5SPM-O signifies a self-protecting multi-path consisting of up to 5 disjoint paths. with equal load balancing. Load balancing is done in an optimal way by a non-integer LP. In the following, we mainly use these abbreviations to refer to specific protection mechanism.

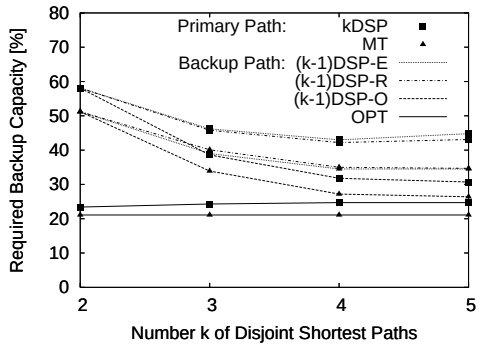
The calculations for the routing and the load balancing were carried out on a Pentium IV 1.5 GHz standard PC. The computation time for the kSPM-O and {MT,kDSP}-$(k-1)$DSP-O was in the rage of seconds for small and of minutes for large networks. The {MT,kDSP}-OPT computation is more complex and took up to several hours.

Impact of Multi-Path Routing and Load Balancing

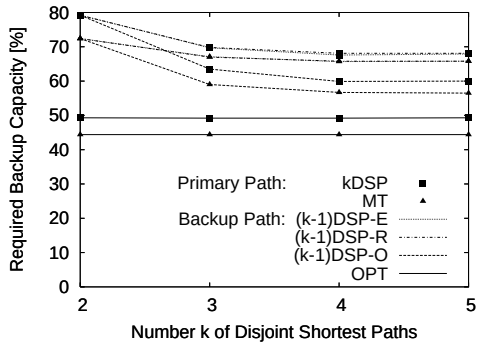
We investigate the impact of multi-path routing and load balancing on the backup performance. We first consider path protection schemes and then study the self-protecting multi-path. Our computations are based on the topology of the COST 239 core network and the Lab03 network from Figures 3.14(b) and 3.14(a), and on homogeneous traffic matrices. We use full traffic reduction and bandwidth reuse, and the protection of single router and link failures as default since 30% of all network failures are due to router failures and 70% of them are due to link failures [286].

PP Schemes Figures 4.6(a) and 4.6(b) show the required backup capacity in the COST239 and the Lab03 network for all path protection schemes ({kDSP,MT}-{$(k-1)$DSP-{E,R,0},OPT} with $2 \leq k \leq 5$). We discuss some observations.

The following holds both for primary paths found by MT and by kDSP. For $k = 2$, only one backup path is available. If a primary path fails, 100% of the traffic is transported over the remaining path, i.e., the performance of all load balancing alternatives (E, R, O) coincides. For larger k, more disjoint backup paths are available and the traffic can be better distributed in a failure case. Therefore, less extra capacity is required on the backup links. The most striking performance gain is achieved for taking $k = 3$ instead of $k = 2$. Due to the network topology, only 3 disjoint path can be found in most cases even for $k = 4$. This limits the reduction of the required backup capacity.



(a) COST-239 network.



(b) Lab03 network.

Figure 4.6: *Impact of multi-path routing and load balancing for path protection methods.*

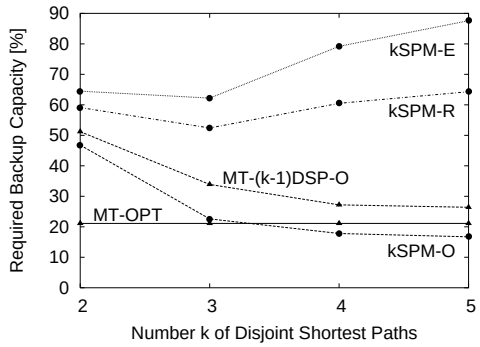
The layout of the backup path and the load balancing for the optimum backup solution (OPT) depends only on the primary path layout. PP mechanisms based on {MT,kDSP}-OPT solutions are most efficient because the backup path is not limited by k disjoint shortest paths. As a consequence, the performance of {MT,kDSP}-OPT is almost independent of k. However, complex multi-path structures are hard to deploy and to manage in failure cases. In addition, the backup path computation is very time consuming.

The layout of the primary path depends on the heuristic (MT or kDSP for a specific k). It has a significant influence on the required extra capacity. Throughout all experiments, the results for minimum traffic (MT) routing yields by 5-10 percent points better results than taking the shortest path of kDSP as primary path.

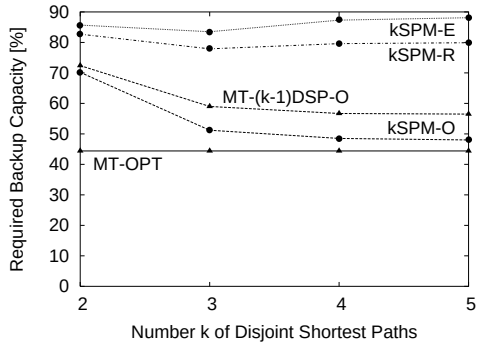
Equal and reciprocal load balancing on the backup multi-path lead approximately to the same results. The optimization of load balancing reduces the required extra capacity by about 10 percent points and the combined optimized backup path and load balancing computation yield another 5-10 percent points capacity reduction.

If a large k effects a longer primary path, it requires more capacity for normal operation without any failure. In contrast to load balancing options R and O, the load balancing option E cannot compensate the increased capacity requirements by load distribution because the load assignment is independent of the path length. As a result, slightly more capacity is required for 5DSP-4DSP-E than for 4DSP-3DSP-E in the COST-239 network.

SPM Figures 4.7(a) and 4.7(b) show the required backup capacity in the COST 239 and the Lab03 network for various SPMs (kSPM-{E,R,O}) in comparison with the best PP schemes (MT-($k-1$)DSP-O and MT-OPT).



(a) COST-239 network.



(b) Lab03 network.

Figure 4.7: *Impact of multi-path routing and load balancing for the Self-Protecting Multi-Path.*

In contrast to the PP methods, the load balancing function (E, R, O) has a greater impact on the backup performance of SPMs than for PP methods and their impact increases with larger k. Although a capacity reduction is expected due to increased path diversification in failure cases, the backup performance of kSPM-E and kSPM-R degrades considerably with increasing k in the COST-239 network. In the Lab03 network, it stays about constant. If k increases, longer paths join the SPM. The SPM with equal or reciprocal load balancing (kSPM-E or kSPM-R) cannot avoid their extensive use which leads to an increased required network capacity. Hence, SPM with simple load balancing schemes reveal only minor benefits.

Optimized load balancing reduces the required backup capacity of the SPM considerably and the potential savings increase with the path diversification. 5SPM-O is about 10 percent points superior to MT-4DSP-O in both networks, which has been proven to be the best feasible PP solution. In the COST 239 network, 5SPM-O is even better than MT-OPT. It requires only 17% additional capacity to protect the network against all link and router failures. We motivate the superiority of the SPM by the following explanation. In contrast to a single primary path, an SPM distributes the traffic from a single source through the network over several disjoint paths. In case of a link failure, the affected traffic stems from more different aggregates and only a fraction of each of their traffic $c(g)$ is carried over the failed link. The load of the failed link can be spread out over more backup paths and links because more aggregates are affected than with a single primary paths if PP mechanisms are used. As a consequence, less shareable backup capacity is required on the individual links.

Like above, there is only a single backup path for $k = 2$ in a failure case but the corresponding extra capacities for 2SPM-{E,R,O} do not coincide in the figure, i.e., load balancing does matter. The optimized load balancing distributes the traffic in such a way that strong traffic concentrations are prevented in any network element. This avoids that a large traffic rate must be redirected if this element fails. This idea is similar to the MT heuristic for finding suitable primary paths.

Impact of Network Topologies

Figure 4.8 shows the required backup capacity for various protection mechanisms in various example networks. A point in the figure indicates the required backup capacity for a certain network and protection mechanism. The x-axis indicates the average number of disjoint parallel paths k^* per b2b relation in the respective network and the y-axis indicates the required backup capacity. The studied protection switching mechanisms are simple OSPF rerouting, 5DSP-4DSP-O, MT-4DSP-O, 5DSP-OPT, MT-OPT, and 5SPM-O, and their corresponding required backup capacities are distinguished by the point shape. Symbols belonging to the same network are grouped together by a vertical line. The sequence of these vertical lines maps the sequence of the letters in the figure. Lowercase letters correspond to networks taken from [261] while uppercase letters correspond to these networks with the modification that nodes with a node degree of at most 2 are successively removed. We have assembled these topologies in the appendix. Therefore, they have a higher average node degree than their lowercase counterparts. Note that the MT-5DSP and MT-OPT protection mechanisms are missing for some networks because no backup path could be found due to the choice of the primary path.

In general, we observe that the required backup capacity decreases with increasing k^* for all protection mechanisms. The dashed line shows the least square interpolation of the results for 5SPM-0 according to an exponential function. Furthermore, the relative savings compared to OSPF rerouting increase with increasing k^*. The SPM is superior to all feasible PP schemes. That can be explained as follows. A kDSP-$k-1$DSP-O is structurally very similar to a kSPM because they use the same disjoint paths of a kDSP computation. But due to the limitation of Equation (4.29), the optimization of the load balancing for path protection methods has fewer degrees of freedom, so comparable SPMs require less backup capacity. The 5SPM-0 clearly outperforms mostly all other protection mechanisms, only the optimized backup paths 5DSP-OPT and MT-OPT lead sometimes to less backup capacity at the expense of a complex multi-path backup structure. Hence,

the SPM is the best of all feasible solutions in all investigated networks.

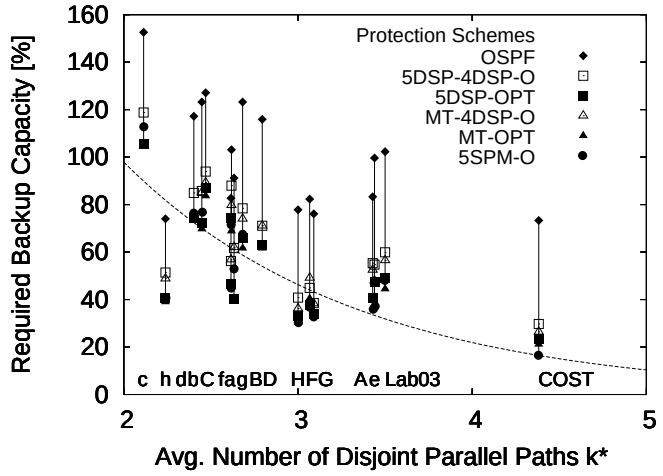
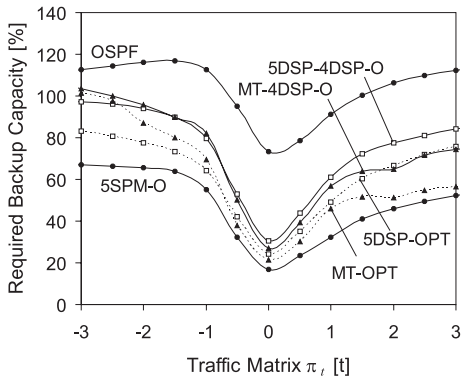


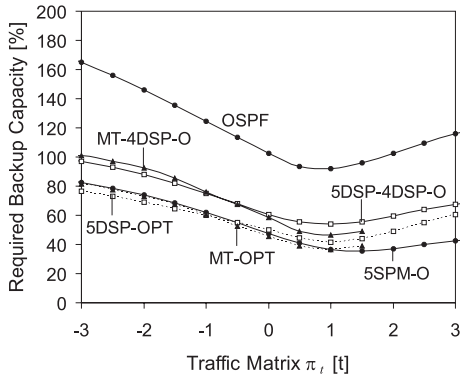
Figure 4.8: *Comparison of protection switching mechanisms in example networks.*

Impact of the Traffic Matrix

Figures 4.9(a) and 4.9(b) show the required backup capacity in the COST-239 and the Lab03 network for different traffic matrices π_t that are obtained as described by Equation (3.42) in Section 3.5 [287]. Hence, the x-axis shows the extrapolation parameter t. All previously discussed protection switching mechanisms are compared. Solid lines stand for well implementable solutions. The dashed lines refer to a general backup multi-path and are rather of theoretical interest.



(a) COST-239 network.



(b) Lab03 network.

Figure 4.9: *The required backup capacity depending on the traffic matrix.*

The curves in both figures differ significantly in their absolute shape but all have a minimum required backup capacity in common. The required backup capacity in the COST-239 network increases for all mechanisms clearly with the variation of the traffic matrix whereas the minimum capacity for the Lab03 network is obtained for $t = 1$ which corresponds to the most realistic traffic matrix. Hence, both the network topology and the traffic matrix have an impact on the required capacity.

The major difference in the required backup capacity results from the protection switching mechanisms. The difference in required backup capacity between OSPF and the protection switching mechanisms is evident for all traffic matrices. The SPM is by far the most efficient one of the well implementable solutions and outperforms often even MT-OPT and 5DSP-OPT. The curves for MT-OPT and MT-4DSP stop at $t = 1.5$ in Figure 4.9(b) because MT routing yields for larger values of t some primary paths that prohibit the existence of a disjoint backup path. Moreover, the difference between the required backup capacity of SPM and OSPF is almost constant, i.e., the absolute capacity savings of about 60% do not depend on the traffic matrix. Hence, the performance of the SPM is very attractive for all traffic matrices in our investigated networks and outperforms clearly other feasible mechanisms.

Comparison with P-Cycles

In [273, 272] the p-cycle concept has been investigated. An optimal p-cycle layout has been found to protect the network with the least capacity possible using a maximum cycle length as side constraint. The experiments were also conducted with the COST-239 network but with the original and partly asymmetric traffic matrix which is given in [221]. The most effective solution required 44% more backup-capacity-related to the capacity requirements for shortest path routing based on the hop count without resilience. For comparison reasons, we calculate the performance value for the 5SPMO and get an additional bandwidth of 23.4%.

4.4.2 Performance of the Self-Protection Multi-Path

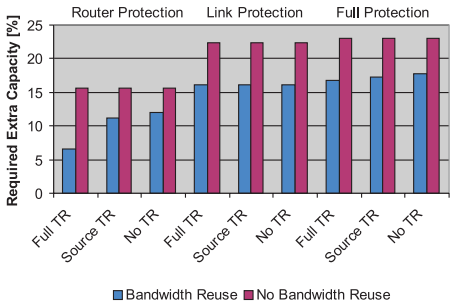
We focus now on the SPM and investigate the impact of resilience constraints on the required backup capacity. Furthermore, we study the influence of different network characteristics using random networks. We use again homogenous traffic matrices in all the experiments.

Impact of Resilience Constraints

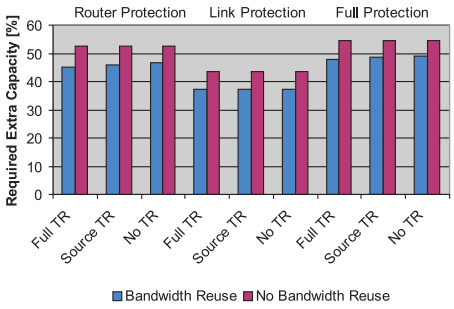
We investigate the impact of the traffic reduction options, the protection options, and the bandwidth reuse options on the required backup capacity.

Figures 4.10(a) and 4.10(b) show the required backup capacity for the 5SPM-O protection switching scheme in the COST-239 and in the Lab03 network. The traffic reduction has no effect if only link failures occur because no router outages are considered in the failure scenario. For full and router protection, it has hardly any effect except for router protection in the COST-239 network. Due to the small size of that network, the proportion of the reduced traffic is large, related to the overall traffic and, therefore, the impact of full traffic reduction is significantly larger than in the Lab03 network.

We consider the influence of the protection option. The most capacity is needed for full protection in any case. In the COST-239 network, router failure protection needs the least backup capacity while link failure protection needs the least backup capacity in the Lab03 network. The reason for that contradictory result is the network size and the average path length. The COST-239 network has a small average shortest path length of $len_{path}^{avg} = 1.56$ and only a few flows traverse transit routers. Only these flows are redirected if a router fails. Other flows originating or ending at a failed router do not impact the resource requirements because they are either removed or they stay unchanged depending on the considered traffic reduction option.



(a) COST-239 network.



(b) Lab03 network.

Figure 4.10: *Impact of protected failure scenarios, traffic reduction, and bandwidth reuse.*

The medium sized Lab03 network has a longer average path length of $len_{path}^{avg} = 2.15$. Since more aggregates are transit flows than in the COST-239 network, router failures cause more deviated traffic. As a consequence, router failure protection requires almost as much backup capacity as full protection in the Lab03 network and the mere link failure protection is about 10 percent points cheaper.

Throughout all experiments the "no bandwidth reuse" restriction leads to about 5 percent points more backup capacity compared to "bandwidth reuse" by backup paths.

Impact of Network Characteristics

To study the impact of the network topology on the required backup capacity in more detail, we conduct studies based on random networks and take 5SPM-O as protection switching mechanism. At first, we describe our algorithm for the construction of random networks. Then we illustrate the impact of the network topology on the backup performance of SPMs both in absolute values and in comparison to the backup performance of OSPF rerouting.

Construction of Random Networks We construct random networks and control some of their essential characteristics. One of them is the degree $deg(v)$ of a node v, which is the number of links v is connected with. As we lay importance on different aspects than in Section 3.6.4, we propose a new network construction method. It incorporates features of the well know Waxman model [31, 288] and we explain it briefly. It is an efficient algorithm that provides control over the minimum, the average, and the maximum node degree (deg_{min}, deg_{avg}, deg_{max}), and avoids loops and parallels.

The algorithms starts with an empty link set $\mathcal{E} = \emptyset$ and defines a single arbitrary node $v_{start} \in \mathcal{V}$ connected. Then, $\frac{|\mathcal{V}| \cdot deg_{avg}}{2}$ links are added successively to $\mathcal{E}$ by connecting suitable nodes v_α and v_ω. An arbitrary node v_α is chosen from a set of preferred nodes $\mathcal{V}_\alpha$ with the following properties. All $v \in \mathcal{V}_\alpha$ are con-

nected and have $deg(v) \leq deg_{max}$. If a node $v \in \mathcal{V}$ exists with $deg(v) < deg_{min}$, all $v \in \mathcal{V}_\alpha$ must have $deg(v) < deg_{min}$. The set of potential neighbor nodes $\mathcal{V}_\omega$ obeys the following requirements: Loops and parallels must be avoided, i.e. $v_\alpha \notin \mathcal{V}_\omega$ and $(v_\alpha, v_\omega) \notin \mathcal{E}$. Furthermore, if an unconnected node $v \in \mathcal{V}$ exists, all $v \in \mathcal{V}_\omega$ must be unconnected. The node $v_\omega \in \mathcal{V}_\omega$ is chosen according to a probability distribution which depends on v_α and $\mathcal{V}_\omega$. Here, the Waxman model comes into play. Each node has a position in the plane. The Euclidean distance $d(v, w)$ induces a weight $P(v, w) = a \cdot e^{-\frac{d(v,w)}{b \cdot d_{max}}}$ with $d_{max} = \max_{v,w \in \mathcal{V}} d(v, w)$, and $P(v, w)$ produces the probability distribution $p_{v_\alpha}(w) = \frac{P(v_\alpha, w)}{\sum_{v \in \mathcal{V}_\omega} P(v_\alpha, v)}$. Given a maximum node degree deviation deg_{dev}^{max}, the minimum node degree is set to $deg_{min} = \max(deg_{avg} - deg_{dev}^{max}, 2)$ and the maximum node degree is set to $deg_{max} = deg_{avg} + deg_{dev}^{max}$.

Absolute Backup Performance We investigate the required backup capacity for 240 random networks of different size, different average node degree deg_{avg}, and different maximum node degree deviation deg_{dev}^{max}. There are 5 random networks for each topology description. In Figure 4.11, the x-axis indicates the average number of disjoint parallel paths k^* that are found for all source–destination pairs in a network and the y-axis shows the required backup capacity. In general, we observe that the required backup capacity decreases with increasing k^*. We identify four clusters of networks that are marked by dashed lines which are least square interpolations among the points of these clusters according to an exponential function. It turns out that all networks of a cluster have the same average node degree deg_{avg}. The dashed lines make the clusters more visible. However, the extrapolation of those curves does not make sense since deg_{avg} is a trivial upper bound on k^*. Within a cluster, the network size n seems to be irrelevant. A small maximum deviation deg_{dev}^{max} of the node degree $deg(v)$ from the average node degree deg_{avg} seems to increase k^* and leads to more efficient backup solutions within a cluster. Therefore, resilience can be achieved at lower cost if the network topology is symmetric.

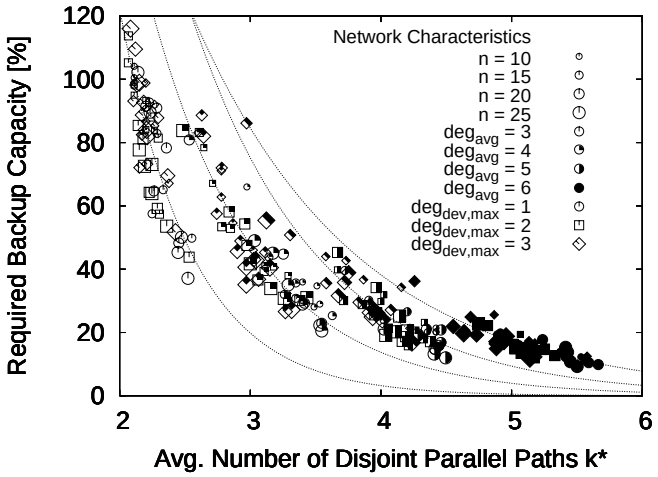


Figure 4.11: *Required extra capacity for SPM in random networks relative to unprotected network dimensioning.*

Relative Backup Performance Figure 4.12 shows the required backup capacity for the SPM and the same networks in relation to the required backup capacity for OSPF rerouting. The OSPF normalization dampens the influence of topological characteristics and shows clearly the benefits of the SPM approach in comparison with conventional rerouting. The figure shows for the 5 random networks with the same characteristics the mean of their average node degree k^* and the mean of their ratios of the SPM and OSPF rerouting backup capacity. The horizontal and vertical lines provide the 90% confidence intervals. The data are plotted on a logarithmic scale to make exponential trends better visible.

The dashed line is the least square interpolation of all experiments and the solid lines are the interpolations within a cluster of networks with the same average node degree deg_{avg}. The four clusters confirm the above observation that

deg_{avg} of a network is strongly correlated with k^*. Increasing the average node degree deg_{avg} shifts the exponential trend slightly towards larger backup capacity. Again, we observe an exponential decay with regard to an increasing k^*, i.e., the superiority of the SPM over OSPF rerouting increases with a larger average number of disjoint paths k^* because SPM reduces the required backup capacity by multi-path forwarding.

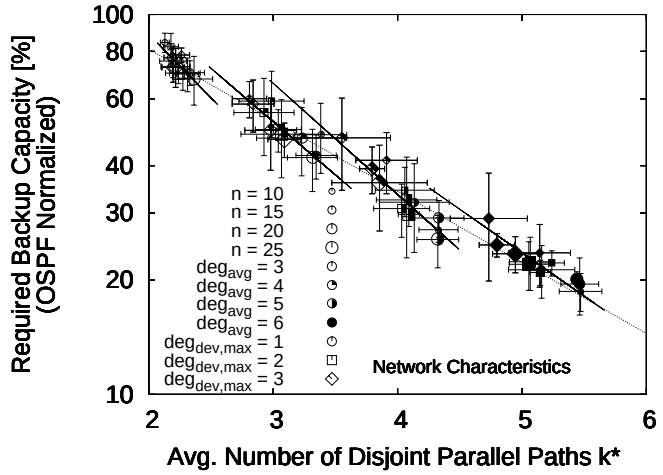


Figure 4.12: *Required extra capacity for SPM in random networks in relation to OSPF routing.*

5 Conclusion

This work was driven by the economic requirement to substitute the coexistence of best effort packet-switched data networks and circuit-switched voice networks by so-called next generation networks (NGNs). These NGNs are based on IP technology and must offer quality of service (QoS) and a high reliability as provided by traditional telephone systems. Real-time QoS in terms of packet loss and delay can be achieved by limiting the traffic rate in the network on the flow level by admission control (AC). Since traffic enters the network at many sites, AC is an inherently distributed problem with various solutions that we call network AC (NAC). Methods for NAC have not been classified and compared before. Resilience against network failures can be achieved by resource duplication or by traffic deviation around the outage location. The latter option is especially attractive because it requires less capacity due to shared protection but it corrupts the resource planning of conventional NAC schemes. Therefore, this work consists of two major parts: (1) the investigation of different NAC concepts and their adaptation to resilience requirements and (2) the investigation of new protection switching mechanisms for MPLS that allow for traffic deviation around local outages at low cost.

First, we classified different budget-based NAC (BNAC) methods. This led to the ILB/ELB NAC which is new and reveals to be the most efficient truly stateless core BNAC method. To compare the BNAC methods, our performance evaluation framework uses as performance measure the achievable resource utilization in a

network with appropriately provisioned link bandwidths. We suggested an efficient implementation for capacity dimensioning based on Kaufman and Robert's method for the calculation of blocking probabilities for multi-rate traffic. The fact that a higher resource utilization can be achieved for larger offered load is called economy of scale or multiplexing gain and it is the key for understanding NAC performance. We illustrated its dependence on various system parameters on a single link but none of them has a similarly strong effect like the offered load. We explained the BNAC-type-specific capacity dimensioning of the budgets, which relies on the proposed capacity dimensioning algorithm. Then, we derived the link capacities that are required to support all traffic patterns that are admissible by the NAC.

Our analytical results showed that the LB NAC achieves the highest resource utilization, followed by the ILB/ELB NAC, the ILB NAC, and the BBB NAC. All these methods can achieve almost 100% utilization for sufficiently large offered load ($a = 10000$ *Erl*). The IB/EB NAC and the IB NAC are least efficient and achieve a resource utilization of at most 25% depending on the network size. Their utilization is limited to a low value because a lot of capacity must be provided in the network to support many admissible traffic patterns that are rather unrealistic. We conducted experiments in different networking scenarios that deepened the understanding of the methods and confirmed these findings.

As overload is most likely to occur due to partial network failures [3], NAC must be resilient in these cases to maintain QoS. Therefore, we proposed the concept of resilient NAC and extended the performance evaluation towards resilience requirements. This means that a set $\mathcal{S}$ of failure scenarios is to be protected in the sense that enough capacity must be provided to carry all admissible traffic patterns also in outage scenarios where the traffic is rerouted. Thus, some of the network capacity is reserved during normal operation for backup purposes in outage scenarios, which reduces the achievable resource utilization for all BNAC methods. The most interesting finding is that under resilience requirements the BBB NAC is most efficient for large offered load, followed by the ILB/ELB NAC, the ILB NAC, and the LB NAC. Hence, the BBB NAC is the recommended option

for resource efficient resilient QoS provisioning. Therefore, network management software must assign suitable capacities to border-to-border (b2b) budgets (BBB) depending on the fixed bandwidth of the links in operational networks and a given traffic matrix. The budget sizes must be set in such a way that unintended resource overbooking is avoided and that all admissible traffic patterns can be carried after rerouting in all protected failure scenarios $\mathcal{S}$. We identified the aspects of link budget assignment, network budget assignment, and resilient budget assignment on the way to the solution of that task and proposed two algorithmic alternatives for each issue. We showed that the more elaborate algorithms provide fairer results in terms of flow blocking and that they lead to larger budget capacities without violating QoS and resilience constraints. The runtime of the algorithms depends on the network capacity. It is in the order of a second on a standard Pentium IV PC to calculate suitable budget capacities for the KING testbed when the links have a bandwidth of 1 *Gbit/s*.

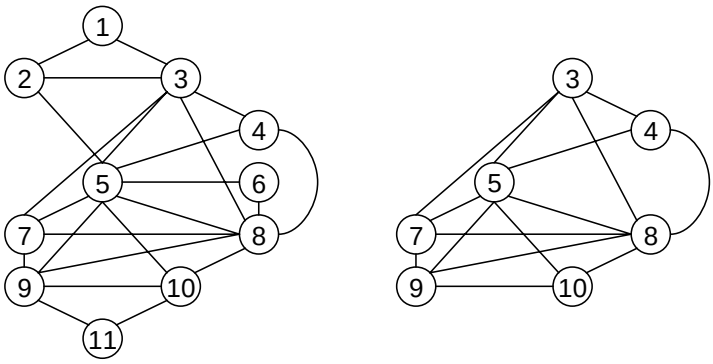
If resilience against network failures is required, routing and rerouting has a major influence on the required backup capacity and the overall resource utilization. Hence, for economic reasons, the objective is to design routing and rerouting for specific failure cases in such a way that a minimum amount of bandwidth is required by subsequent network dimensioning. We chose MPLS as the base technology for routing optimization because it provides finer control on data forwarding than plain IP routing. After the discussion of related work, we pointed out the shortcomings of existing routing optimizations. We designed so-called Path Protection (PP) mechanisms and the Self-Protecting Multi-path (SPM) whose main features are multi-path routing, load balancing, and a relatively simple implementation compared to other methods. We derived a joint description for optimum path layout and load balancing which contains, however, quadratic equations that prohibit an efficient solution of the optimization problem. We separated the problem into a heuristic path layout and an optimization of the load balancing using linear programs. Our experiments showed that the combination of multi-path routing and optimized load balancing reduces the required backup capacity considerably. The required backup capacity depends

significantly on the structure of the traffic matrix and on resilience constraints. We investigated about 20 different artificial and realistic network topologies. In comparison with other mechanisms, the SPM revealed to be the most economic viable solution. In the COST239 network, only 17% backup capacity is required to protect all single link and node failures. We also compared the SPM with the recently discussed p-cycle approach and demonstrated its superiority. Finally, we showed that the required backup capacity decreases about exponentially with the number of disjoint paths per aggregate in the network and that the advantage of the SPM over single shortest path routing increases in the same way.

In conclusion, resilience requirements introduce new challenges for Quality of Service in NGNs. The KING project [109, 2] has adopted the NAC solution presented in this work together with the accompanying software for network management. As the presented resilience mechanisms rely on explicit routing, they cannot be supported efficiently by plain IP technology in the KING project. However, MPLS has already emerged in operational networks and it facilitates a natural implementation of BBB NAC concept. Therefore, a combination of resilient NAC and routing as presented in this work is attractive for NGN solutions.

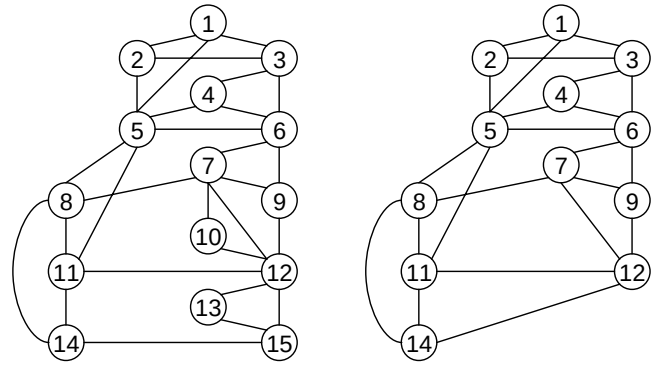
Appendix

In this section, we provide the example networks used in Section 4.4.1. Lowercase letters correspond to networks taken from [261] while uppercase letters correspond to these networks with the modification that nodes with a node degree of at most 2 are successively removed.



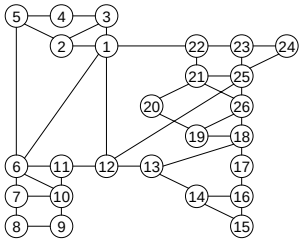
(a) Network “a”. (b) Modified network “A”.

Figure 5.1: *Example networks from [261].*

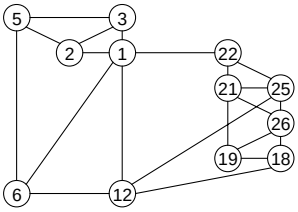


(a) Network “b”.

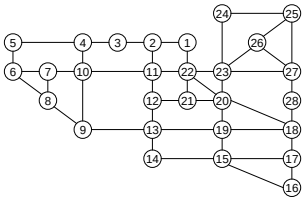
(b) Modified network “B”.



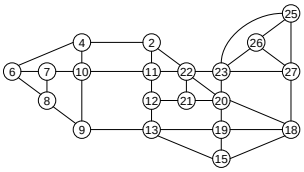
(c) Network “c”.



(d) Modified network “C”.

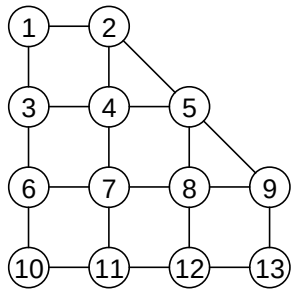


(e) Network “d”.

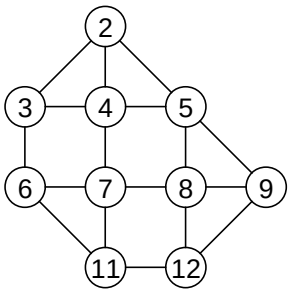


(f) Modified network “D”.

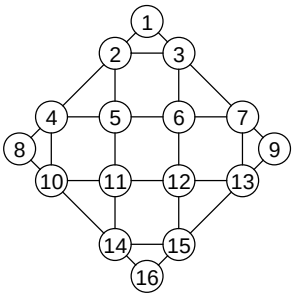
Figure 5.2: *Example networks from [261].*



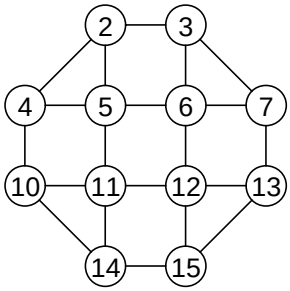
(a) Network “f”.



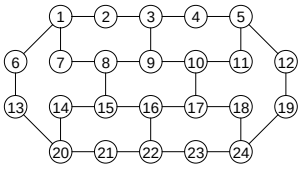
(b) Modified network “F”.



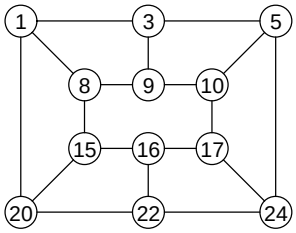
(c) Network “g”.



(d) Modified network “G”.

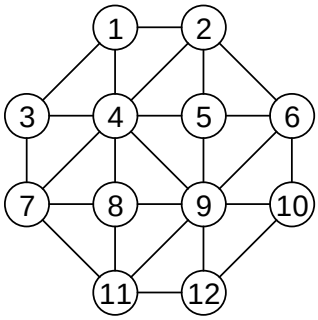


(e) Network “h”.



(f) Modified network “H”.

Figure 5.3: *Example networks from [261].*



(a) Network “e” contains only nodes with a node degree of at least 3.

Figure 5.4: *Example network from [261] which does not need to be modified.*

List of Abbreviations

A-MBAC	MBAC with aggregate measurements, page 45
AAL2	ATM Adaptation Layer type 2, page 61
AC	admission control, page 2
AL	application layer, page 9
AS	autonomous system, page 18
ASN	AS number, page 19
ATM	Asynchronous Transfer Mode, page 2
BAM	budget assignment method, page 136
BB	bandwidth broker, page 56
BBB	b2b budget, page 48
BGP	Border Gateway Protocol, page 26
BMBF	Bundesministerium für Bildung und Forschung, page 38
BNAC	budget-based NAC, page 3

CAPEX capital expenses, page 1

CIDR Classless Interdomain Routing, page 21

CM construction method, page 109

CNBA concurrent NBA, page 136

CR-LDP Constraint-Based LDP, page 28

CRBA concurrent RBA, page 140

CS complete sharing, page 69

DHCP Dynamic Host Configuration Protocol, page 17

DiffServ Differentiated Services, page 32

DNS Domain Name System, page 18

DSCP DiffServ Code Point, page 32

DSL Digital Subscriber Line, page 11

E equal load balancing, page 158

E-BGP Exterior BGP, page 27

e2e end-to-end, page 2

EB egress budget, page 48

EBAC experience-based AC, page 44

ECMP Equal Cost Multi-Path, page 25

EDF Earliest Deadline First, page 34

EGP exterior gateway routing protocol, page 23

ELB egress link budget, page 48

EQOS Edge Assisted QoS, page 57

F-MBAC MBAC with flow measurements, page 45

FDDI Fiber Distributed Data Interface, page 11

FEC forwarding equivalent class, page 147

FIFO First-In-First-Out, page 33

FLBA fair LBA, page 127

FNAC feedback-based NAC, page 46

FTP File Transfer Protocol, page 18

FWP framework program, page 38

GPS Generalized Processor Sharing, page 33

HDLC High Level Data Link Control Protocol, page 10

HTTP Hypertext Transfer Protocol, page 8

I-BGP Interior BGP, page 27

IB ingress budget, page 48

ICANN Internet Corporation for Assigned Names and Numbers, page 19

ICMP Internet Control Message Protocol, page 12

IETF Internet Engineering Task Force, page 8

IGP interior gateway routing protocol, page 23

ILB ingress link budget, page 48

ILM	incoming label map, page 27
INBA	independent NBA, page 134
IP	Internet Protocol, page 1
IPv4	IP version 4, page 13
IPv6	IP version 6, page 13
IRBA	independent RBA, page 140
IS-IS	Intermediate System to Intermediate System Routing Exchange Protocol, page 25
ISO	International Standardization Organization, page 10
ISP	Internet service provider, page 2
IST	information society technologies, page 38
ITU	International Telecommunication Union, page 18
IXP	Internet Exchange Point, page 20
kDSP	k disjoint shortest paths, page 157
KING	Key Components for the Internet of the Next Generation, page 38
LAC	link AC, page 2
LB	link budget, page 48
LBA	link budget assignment, page 126
LDP	Label Distribution Protocol, page 28
LIB	label information base, page 148

LL link layer, page 10

LLC logical link control, page 9

LP linear program, page 4

LSP label-switched path, page 27

LSP link state package, page 24

LSR label switching router, page 27

MAC media access control, page 10

MBAC measurement-based AC, page 43

MEDF Modified EDF, page 34

MIB management information base, page 28

MP multi-path, page 106

MPLS Multiprotocol Label Switching, page 3

MT minimum traffic, page 157

MTU Maximum Transfer Unit, page 12

NAC network AC, page 2

NAP Network Access Point, page 20

NAT Network Address Translation, page 13

NBA network budget assignment, page 126

NCS network control server, page 63

NGN Next Generation Network, page 1

NL	network layer, page 9
NSIS	Next Steps in Signaling, page 49
O	optimized load balancing, page 159
OPEX	operational expenses, page 1
OPT	optimum multi-path backup structure, page 158
OPWA	one-pass with advertising, page 53
OSI	Open System Interconnection, page 10
OSPF	Opens Shortest Path First, page 24
PHB	Per-Hop Behavior, page 32
PL	physical layer, page 10
PLAC	parameter-based LAC, page 43
PLBA	proportional LBA, page 127
POP	Point of Presence, page 20
PP	Path Protection, page 154
PPP	Point-to-Point Protocol, page 10
PS	PATH state, page 52
QoS	quality of service, page 1
R	reciprocal load balancing, page 159
RBA	resilient budget assignment, page 126
RED	Random Early Detection, page 33

REM rate envelope multiplexing, page 43

RFC Request for Comments, page 8

RIP Routing Information Protocol, page 24

RMD Resource Management in Differentiated Services IP Networks, page 58

RR receiver report, page 54

RS RESV state, page 52

RSVP Resource Reservation Protocol, page 16

RSVP-TE RSVP Tunneling Extensions, page 28

RTCP RTP Control Protocol, page 16

RTP Real-Time Transport Protocol, page 16

RTSP Real-Time Streaming Protocol, page 17

SDH Synchronous Digital Hierarchy, page 12

SIP Session Initiation Protocol, page 17

SLA service level agreement, page 32

SMTP Simple Mail Transfer Protocol, page 18

SP Static Priority, page 33

SP shortest path, page 4

SP single-path, page 106

SPM Self-Protecting Multi-Path, page 4

SR	sender report, page 54
SRLG	Shared Risk Link Group, page 153
SRP	Scalable Resource Reservation Protocol, page 58
ST2	Internet Stream Protocol version 2, page 54
TCA	traffic conditioning agreement, page 32
TCP	Transmission Control Protocol, page 15
TL	transport layer, page 9
ToS	Type of Service, page 12
TR	trunk reservation, page 69
TTL	Time-to-Live, page 12
UDP	User Datagram Protocol, page 15
UMTS	Universal Mobile Telecommunication System, page 12
URL	uniform resource locator, page 8
UTRAN	UMTS Terrestrial Radio Access Network, page 12
VCC	Virtual Channel Connections, page 34
VPC	virtual path connection, page 61
VPN	Virtual Private Network, page 29
WFQ	Weighted Fair Queuing, page 33
WLAN	Wireless Local Access Network, IEEE 802.11, page 11
WRR	Weighted Round Robin, page 33
YESSIR	YEt another Sender Session Internet Reservation, page 54

List of Figures

Bibliography

[1] J. Roberts, U. Mocci, and J. Virtamo, *Broadband Network Teletraffic - Final Report of Action COST 242*. Berlin, Heidelberg: Springer, 1996.

[2] 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), April 2003.

[3] 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), April 2003.

[4] "Internet Engineering Task Force (IETF)." http://www.ietf.org/.

[5] R. Fiedling, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "RFC2616: Hypertext Transfer Protocol – HTTP/1.1." ftp://ftp.isi.edu/in-notes/rfc2616.txt, June 1999.

[6] D. Perkins, "RFC1547: Requirements for an Internet Standard Point-to-Point Protocol." http://www.ietf.org/rfc/rfc1547.txt, Dec. 1993.

[7] W. S. (ed.), "RFC1661: The Point-to-Point Protocol (PPP)." http://www.ietf.org/rfc/rfc1661.txt, July 1994.

[8] W. Simpson, "RFC2153: PPP Vendor Extensions." ftp://ftp.isi.edu/in-notes/rfc2153.txt, May 1997.

[9] G. McGregor, "RFC1332: The PPP Internet Protocol Control Protocol (IPCP)." http://www.ietf.org/rfc/rfc1332.txt, May 1992.

[10] R. Jain, *FDDI Handbook: High-Speed Networking Using Fiber and Other Media*. MA: Addison-Wesley, 1994.

[11] R. Steinmetz and K. Nahrstedt, *Multimedia Systems*. Heidelberg: Springer, 2004.

[12] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*. San Mateo, CA: Morgan and Kaufman, 1996.

[13] N. Vicari, *Modeling of Internet Traffic: Internet Access Influence, User Interference, and TCP Behavior*. PhD thesis, University of Würzburg, Faculty of Computer Science, Am Hubland, Apr. 2003.

[14] I. S. Institute, "RFC793: Transmission Control Protocol." http://www.ietf.org/rfc/rfc0793.txt, September 1981.

[15] J. Postel, "RFC768: User Datagram Protocol." http://www.ietf.org/rfc/rfc0791.txt, Sep. 1980.

[16] J. Postel, "RFC792: Internet Control Message Protocol." http://www.ietf.org/rfc/rfc0792.txt, Sept. 1981.

[17] A. Conta and S. Deering, "RFC2463: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)." ftp://ftp.isi.edu/in-notes/rfc2463.txt, Dec. 1998.

[18] B. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "RFC2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification." ftp://ftp.isi.edu/in-notes/rfc2205.txt, Sep. 1997.

[19] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC1889: RTP - A Transport Protocol for Real-Time Applications." ftp://ftp.isi.edu/in-notes/rfc1889.txt, Jan. 1996.

[20] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC3550: RTP - A Transport Protocol for Real-Time Applications." ftp://ftp.isi.edu/in-notes/rfc3550.txt, July 2003.

[21] S. Boll, C. Heinlein, W. Klas, and M. Menth, "MPEG-L/MRP: Implementing Adaptive Streaming of MPEG Videos for Interactive Internet Applications," in *ACM Multimedia*, (Ottawa/Ontario, Canada), Oct. 2001.

[22] H. Schulzrinne, A. Rao, and R. Lanphier, "RFC2326: Real Time Streaming Protocol (RTSP)." ftp://ftp.isi.edu/in-notes/rfc2326.txt, April 1998.

[23] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC3261: SIP: Session Initiation Protocol." ftp://ftp.isi.edu/in-notes/rfc3261.txt, June 2002.

[24] J. K. (Editor), "RFC2821: Simple Mail Transfer Protocol." ftp://ftp.isi.edu/in-notes/rfc2821.txt, April 2001.

[25] J. Postel and J. Reynolds, "RFC959: File Transfer Protocol (FTP)." http://www.ietf.org/rfc/rfc0959.txt, October 1985.

[26] P. Mockapetris, "RFC1034: Domain Names - Concepts and Facilities." http://www.ietf.org/rfc/rfc1034.txt, November 1987.

[27] P. Mockapetris, "RFC1035: Domain Names - Implementation and Specification." http://www.ietf.org/rfc/rfc1035.txt, November 1987.

[28] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, 2 ed., 2003.

[29] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP Topologies with Rocketfuel," *IEEE/ACM Transactions on Networking*, vol. 12, Feb. 2004.

[30] M. Dodge, "An Atlas of Cyberspaces." http://www.cybergeography.org/atlas/more_isp_maps.html.

[31] E. W. Zegura, K. L. Calvert, and M. J. Donahoo, "A Quantitative Comparison of Graph-Based Models for Internet Topology," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 770–783, 1997.

[32] "Internet Corporation For Assigned Names and Numbers (ICANN)." http://www.icann.org/.

[33] K. Xu, Z. Duan, Z.-L. Zhang, and J. Chandrashekar, "On Properties of Internet Exchange Points and Their Impact on AS Topology and Relationship," in 3^{rd} *IFIP-TC6 Networking Conference (Networking)*, (Athens, Greece), pp. 284 – 295, May 2004.

[34] V. Fuller, T. Li, J. Yu, and K. Varadhan, "RFC1519: Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy." http://www.ietf.org/rfc/rfc1519.txt, Sept. 1993.

[35] J. Mogul and J. Postel, "RFC950: Internet Standard Subnetting Procedure." http://www.ietf.org/rfc/rfc0950.txt, August 1985.

[36] C. Huitema, *Routing in the Internet*. New Jersey: Prentice Hall, 2000.

[37] G. Malking, "RFC2453: Routing Information Protocol (RIP) Version 2." ftp://ftp.isi.edu/in-notes/rfc2453.txt, Nov. 1998.

[38] E. W. Disjkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, pp. 269 – 271, 1959.

[39] J. Moy, "RFC2328: OSPF Version 2." ftp://ftp.isi.edu/in-notes/rfc2328.txt, April 1998.

[40] "Cooperative Association for Internet Data Analysis (CAIDA)." http://www.caida.org/analysis/topology/as_core_network/.

[41] Y. Rekhter and T. Li, "RFC1771: A Border Gateway Protocol Version 4 (BGP-4)." http://www.ietf.org/rfc/rfc1771.txt, Mar. 1995.

[42] Y. Rekhter and P. Gross, "RFC1772: Application of the Border Gateway Protocol in the Internet." http://www.ietf.org/rfc/rfc1772.txt, Mar. 1995.

[43] P. Traina, "RFC1773: Experience with the BGP-4 Protocol." http://www.ietf.org/rfc/rfc1773.txt, Mar. 1995.

[44] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet Routing Instability," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 515 – 528, 1998.

[45] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet Routing Convergence," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 293 – 306, June 2001.

[46] E. C. Rosen, A. Viswanathan, and R. Callon, "RFC3031: Multiprotocol Label Switching Architecture." http://www.ietf.org/rfc/rfc3031.txt, Jan. 2001.

[47] D. O. Awduche, L. Berger, D.-H. Gan, T. Li, V. Srinivasan, and G. Swallow, "RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels." http://www.ietf.org/rfc/rfc3209.txt, Dec. 2001.

[48] B. Jamoussi *et al.*, "RFC3212: Constraint-Based LSP Setup using LDP." http://www.ietf.org/rfc/rfc3212.txt, Jan. 2002.

[49] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas, "LDP Specification." http://www.ietf.org/rfc/rfc3036.txt, Jan. 2001.

[50] X. Xiao, A. Hannan, and L. M. Ni, "Traffic Engineering with MPLS in the Internet," *IEEE Network Magazine*, vol. 38, Mar. 2000.

[51] G. Swallow, "MPLS Advantages for Traffic Engineering," *IEEE Communications Magazine*, pp. 54–57, Dec 1999.

[52] T. Li, "MPLS and the Evolving Internet Architecture," *IEEE Communications Magazine*, pp. 38–41, Dec 1999.

[53] A. Ghanwani, B. Jamoussi, D. Fedyk, P. Ashwook-Smith, L. Li, and N. Feldman, "Traffic Engineering Standards in IP Networks Using MPLS," *IEEE Personal Communications*, pp. 49–53, Dec 1999.

[54] D. O. Awduche, "MPLS and Traffic Engineering in IP Networks," *IEEE Communications Magazine*, pp. 42–47, Dec 1999.

[55] T. M. Chen and S. S. Liu, *ATM Switching Systems*. Artech House, Inc., 1995.

[56] E. Rathgeb and E. Wallmeier, *ATM - Infrastruktur für die Hochleistungskommunikation*. Heidelberg: Springer, 1997.

[57] M. Menth and N. Hauck, "A Graph-Theoretical Notation for the Construction of LSP Hierarchies," in 15^{th} *ITC Specialist Seminar*, (Würzburg, Germany), July 2002.

[58] M. Menth, A. Reifert, and J. Milbrandt, "CSPF Routed and Traffic-Driven Construction of LSP Hierarchies," in *Architectures for Quality of Service in the Internet (Art-QoSM)*, (Warsaw, Poland), Mar. 2003.

[59] K. Kompella and Y. Rekhter, "LSP Hierarchy with Generalized MPLS TE." http://www.ietf.org/internet-drafts/draft-ietf-mpls-lsp-hierarchy-08.txt, March 2002.

[60] H. Hummel and J. Grimminger, "Hierarchical LSP." http://www.ietf.org/internet-drafts/draft-hummel-mpls-hierarchical-lsp-01.txt, May 2002.

[61] H. Hummel and B. Hoffmann, "O(n**2) Investigations." http://www.ietf.org/internet-drafts/draft-hummel-mpls-n-square-investigations-00.txt, June 2002.

[62] ITU-T, "Recommendation E.600. Terms and Definitions of Traffic Engineering." International Telecommunication Union, March 1993.

[63] ITU-T, "Recommendation E.800. Terms and Definitions Related to Quality of Service and Network Performance Including Dependability." International Telecommunication Union, August 1994.

[64] C. Fraleigh, F. Tobabi, and C. Diot, "Provisioning IP Backbone Networks to Support Latency Sensitive Traffic," in *IEEE Infocom*, April 2003.

[65] H. van den Berg, M. Mandjes, R. van de Meent, A. Pras, F. Roijers, and P. Venemans, "QoS-Aware Bandwidth Provisioning in IP Backbone Networks," Tech. Rep. 279TD(03) 034, COST-279, 2003.

[66] H. van den Berg, M. Mandjes, R. van de Meent, A. Pras, F. Roijers, and P. Venemans, "QoS-Aware Bandwidth Provisioning for IP Network Links," *under submission*, 2004.

[67] S. Blake, D. L. Black, M. A. Carlson, E. Davies, Z. Wang, and W. Weiss, "RFC2475: An Architecture for Differentiated Services." ftp://ftp.isi.edu/in-notes/rfc2475.txt, Dec. 1998.

[68] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "RFC2597: Assured Forwarding PHB Group." ftp://ftp.isi.edu/in-notes/rfc2597.txt, June 1999.

[69] V. Jacobson, K. Nichols, and K. Poduri, "RFC2598: An Expedited Forwarding PHB." ftp://ftp.isi.edu/in-notes/rfc2598.txt, June 1999.

[70] K. Nichols, S. Blake, F. Baker, and D. L. Black, "RFC2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers." ftp://ftp.isi.edu/in-notes/rfc2474.txt, Dec. 1998.

[71] M. Menth and S. Schneeberger, "A Scalable Scheduling Mechanism with Application to AAL2/ATM Multiplexing," in 14^{th} *ITC Specialist Seminar*, (Barcelona, Spain), April 2001.

[72] M. May, J.-C. Bolot, A. Jean-Marie, and C. Diot, "Simple Performance Models of Differentiated Services Schemes for the Internet," in *IEEE Infocom*, (New York, USA), Apr. 1999.

[73] T. Bonald and J. W. Roberts, "Performance of Bandwidth Sharing Mechanisms for Service Differentiation in the Internet," in 13^{th} *ITC Specialist Seminar*, (Monterey, USA), September 2000.

[74] N. Christin and J. Liebeherr, "A QoS Architecture for Quantitative Service Differentiation," *IEEE Communications Magazine*, vol. 41, pp. 38 – 45, June. 2003.

[75] N. Christin and J. Liebeherr, "Marking Algorithms for Service Differentiation of TCP Traffic," *Journal of Computer Communications*, 2003.

[76] R. Martin, M. Menth, and V. Phan, "Performance of TCP/IP with MEDF Scheduling," in 3^{th} *International Workshop on Quality of future Internet Services (QofIS)*, (Barcelona, Spain), September 2004.

[77] R. Guérin and V. Peris, "Quality-of-Service in Packet Networks: Basic Mechanisms and Directions," *Computer Networks*, vol. 31, pp. 169–189, Feb. 1999.

[78] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.

[79] S. Köhler, M. Menth, and N. Vicari, "Analytic Performance Evaluation of the RED Algorithm for QoS in TCP/IP Networks," in 9^{th} *IFIP Working Conference on Performance Modelling and Evaluation of ATM & IP Networks*, (Budapest, Hungary), pp. 178–190, June 2001.

[80] D. Lin and R. Morris, "Dynamics of Random Early Dectection," *ACM SIGCOMM Computer Communications Review*, vol. 27, pp. 127–136, Oct. 1997.

[81] T. Bonald, M. May, and J.-C. Bolot, "Analytic Evaluation of RED Performance," in *IEEE Infocom*, (Tel Aviv, Israel), Apr. 2000.

[82] W. Feng, D. Kandlur, D. Saha, and K. Shin, "A Self-Configuring RED Gateway," in *IEEE Infocom*, (New York, USA), Mar. 1999.

[83] K. Kumaran, G. Margrave, D. Mitra, and K. R. Stanley, "Novel Techniques for Design and Control of Generalized Processor Sharing Schedulers for Multiple QoS Classes," in *IEEE Infocom*, pp. 932–941, 2000.

[84] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm," in *ACM SIGCOMM* , pp. 3 – 12, 1989.

[85] J. C. Bennett and H. Zhang, "Why WFQ Is Not Good Enough for Integrated Services Networks," in *Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, pp. 524–532, April 1996.

[86] E. L. Hahne, "Round Robin Scheduling for Max-Min Fairness in Data Networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 1024 – 1039, 1991.

[87] M. Andrews, "Probabilistic End-to-End Delay Bounds for Earliest Deadline First Scheduling," in *IEEE Infocom*, 2000.

[88] V. Sivaraman and F. M. Chiussi, "Providing End-to-End Statistical Delay Guarantees with Earliest Deadline First Scheduling and Per-Hop Traffic Shaping," in *IEEE Infocom*, pp. 631–640, 2000.

[89] D. E. Wrege and J. Liebeherr, "A Near-Optimal Packet Scheduler for QoS Networks," in *IEEE Infocom*, 1997.

[90] M. Menth, M. Schmid, H. Heiß, and T. Reim, "MEDF - A Simple Scheduling Algorithm for Two Real-Time Transport Service Classes with Application in the UTRAN," in *IEEE Infocom*, (San Francisco, USA), Mar. 2003.

[91] M. Menth, J. Milbrandt, and F. Zeiger, "Elastic Token Bucket - A Traffic Characterization for Time-Limited Bursty Traffic," 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), Sept. 2004.

[92] B. Braden, D. Clark, and S. Shenker, "RFC1633: Integrated Services in the Internet Architecture: an Overview." http://www.ietf.org/rfc/rfc1633.txt, June 1994.

[93] D. Clark, S. Shenker, and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism," in *ACM SIGCOMM* , Sep. 1992.

[94] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine, "RFC2998: A Framework for Integrated Services Operation over Diffserv Networks." http://www.ietf.org/rfc/rfc2998.txt, Nov. 2000.

[95] J. Schmitt, M. Karsten, L. Wolf, and R. Steinmetz, "Aggregation of Guaranteed Service Flows," in 7^{th} *IEEE International Workshop on Quality of Service (IWQoS)*, June 1999.

[96] M. Schwartz, *Computer Communication Network Design and Analysis*. New Jersey: Prentice Hall, 1977.

[97] R. L. Freeman, *Telecommunication System Engineering*. Wiley, 1980.

[98] C. S. Inc., "IP Telephony: The Five Nines Story (Whitepaper)." http://www.cisco.com/warp/public/cc/so/neso/vvda/iptl/5nine_wp.htm, Aug. 2002.

[99] M. Dahlin, B. B. V. Chandra, L. Gao, and A. Nayate, "End-to-End WAN Service Availability," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 300–313, April 2003.

[100] C. Boutremans, G. Iannaccone, and C. Diot, "Impact of Link Failures on VoIP Performance," in *Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (Miami, Florida), May 2002.

[101] L. W. (EE-Times), "Packet, Circuit Worlds Collide on Net Reliability." http://www.eetimes.com, July 1999.

[102] B. Krause, "Use Processor Redundancy for Maximum Reliability (Whitepaper)." http://www.commsdesign.com, Feb. 2002.

[103] Nokia, "The Five Nines IP Network." http://www.layer3direct.com/docs/wireless/nokia_wireless/5_Nines_IP_Network.pdf, Jan. 2001.

[104] Nokia, "Five Nines at the IP Edge." http://www.nigeriancomputer-society.com/contentimages/Five-Nines_At_The_Ip_Edge.pdf, Jan. 2003.

[105] N. Networks, "Passport 8600 Routing Switch (Product Information)." http://www.nortelnetworks.com/products/01/passport/8600/fandb.html, 2003.

[106] "Internet2." http://www.internet2.org/.

[107] "Traffic Engineering for Quality of Service in the Internet, at Large Scale." http://www.ist-tequila.org/.

[108] "Adaptive Resource Control for QoS Using an IP-based Layered Architecture (AQUILA)." http://www.ist-aquila.org/.

[109] "Key Components for the Internet of the Next Generation (KING)." http://www.siemens.com/king.

[110] COST-279 Management Committee and M. Menth (Eds.), *COST-279 Midterm Report: Analysis and Design of Advanced Multiservice Networks Supporting Mobility, Multimedia, and Internetworking*. Roma: ARACNE, 1^{st} ed., Jan. 2004.

[111] F. P. Kelly, *Stochastic Networks: Theory and Applications*, vol. 4, ch. Notes on Effective Bandwidths, pp. 141 – 168. Oxford University Press, 1996.

[112] R. J. Gibbens and Y. C. Teh, "Critical Time and Space for Statistical Multiplexing in Multiservice Networks," in 16^{th} *International Teletraffic Congress (ITC)*, (Edinburg), pp. 87 – 96, 6 1999.

[113] S. Bodamer and J. Charzinski, "Evaluation of Effective Bandwidth Schemes for Self-Similar Traffic," in 13^{th} *ITC Specialist Seminar*, (Monterey, USA), Sept. 2000.

[114] L. Kleinrock, *Queueing Systems*, vol. 1: Theory. New York: John Wiley & Sons, 1^{st} ed., 1975.

[115] D. Abendroth and U. Killat, "Intelligent Shaping: Well Shaped Throughout the Entire Network?," in *IEEE Infocom*, (New York City, NY), June 2002.

[116] D. Abendroth and U. Killat, "A Poissonian Traffic Descriptor," in 15^{th} *ITC Specialist Seminar*, (Würzburg, Germany), July 2002.

[117] M. Menth and O. Rose, "Performance Tradeoffs for Header Compression in MPLS Networks," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Munich, Germany), pp. 503 – 508, June 2002.

[118] C. Brandauer, W. Burakowski, M. Dabrowski, B. Koch, and H. Tarasiuk, "AC Algorithms in AQUILA QoS IP Network," in 2^{nd} *Polish-German Teletraffic Symposium (PGTS)*, (Gdansk, Poland), Sept. 2002.

[119] W. Burakowski and H. Tarasiuk, "Admission Control for TCP Connections in QoS IP Network," in 2^{nd} *International Conference on Human.Society@Internet (HSI2003)*, (Seoul, Korea), pp. 283 – 293, Jan. 2003.

[120] 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), Sept. 2004.

[121] T. K. Lee, M. Zukerman, and R. G. Addie, "Admission Control Schemes for Bursty Multimedia Traffic," in *IEEE Infocom*, 2001.

[122] S. Jamin, P. Danzig, S. J. Shenker, and L. Zhang, "Measurement-Based Admission Control Algorithms for Controlled-Load Services Packet Networks," in *ACM SIGCOMM* , 1995.

[123] M. Dabrowski and F. Strohmeier, "Measurement-Based Admission Control in AQUILA Network and Improvements by Passive Measurements," in *Architectures for Quality of Service in the Internet (Art-QoSM)*, (Warsaw, Poland), Mar. 2003.

[124] M. Grossglauser and D. N. C. Tse, "A Framework for Robust Measurement-Based Admission Control," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 293–309, 1999.

[125] M. Grossglauser and D. N. C. Tse, "A Time-Scale Decomposition Approach to Measurement-Based Admission Control," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 550–563, 2003.

[126] S. Jamin, S. Shenker, and P. B. Danzig, "Comparison of Measurement-Based Call Admission Control Algorithms for Controlled-Load Service," in *IEEE Infocom*, pp. 973–980, 1997.

[127] L. Breslau, S. Jamin, and S. Shenker, "Comments on the Performance of Measurement-Based Admission Control Algorithms," in *IEEE Infocom*, pp. 1233–1242, March 2000. ISBN 0-7803-5880-5.

[128] Z. Turanyi, A. Veres, and A. Olah, "A Family of Measurement-Based Admission Control Algorithms," in *IFIP Conference on Performance of Information and Communication Systems*, (Lund, Sweden), May 1998.

[129] K. Shiomoto, N. Yamanaka, and T. Takahashi, "Overview of Measurement-Based Connection Admission Control Methods in ATM Networks," *IEEE Communications Surveys*, vol. 2, no. 1, 1999.

[130] E. Knightly and N. Shroff, "Admission Control for Statistical QoS: Theory and Practice," *IEEE Network Magazine*, vol. 13, no. 2, pp. 20 – 29, 1999.

[131] H. van den Berg and M. Mandjes, "Transient Analysis of Traffic Generated by Bursty Sources, and its Application to Measurement-Based Admission Control," *Telecommunication Systems*, vol. 15, pp. 273 – 293, 2000.

[132] H. van den Berg and M. Mandjes, "Admission Control in Integrated Networks: Overview and Evaluation," in 8^th*International Conference on Telecommunication Systems, Modeling and Analysis (ICTSM)*, (Nashville, US), pp. 132 – 151, 2000.

[133] J. Qiu and E. W. Knightly, "Measurement-Based Admission Control with Aggregate Traffic Envelopes," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 199–210, 2001.

[134] M. Menth, J. Milbrandt, and S. Oechsner, "Experience-Based Admission Control (EBAC)," in 9^{th} *IEEE Symposium on Computers and Communications (ISCC)*, (Alexandria, Egypt), June 2004.

[135] M. Menth, S. Kopf, J. Milbrandt, and J. Charzinski, "Introduction to Budget-Based Network Admission Control Methods," in 28^{th} *IEEE Conference on Local Computer Networks (LCN)*, (Bonn, Germany), Oct. 2003.

[136] R. J. Gibbens and F. P. Kelly, "Distributed Connection Acceptance Control for a Connectionless Network," in 16^{th} *International Teletraffic Congress (ITC)*, (Edinburg), pp. 941 – 952, 6 1999.

[137] L. Breslau, E. W. Knightly, S. Shenker, and H. Zhang, "Endpoint Admission Control: Architectural Issues and Performance," in *ACM SIGCOMM* , Aug. 2000.

[138] C. Cetinkaya and E. W. Knightly, "Egress Admission Control," in *IEEE Infocom*, pp. 1471–1480, 2000.

[139] I. Más and G. Karlsson, "PBAC: Probe-Based Admission Control," in 2^{nd} *International Workshop on Quality of future Internet Services (QofIS)*, September 2001.

[140] G. Bianchi, N. Blefari-Melazzi, M. Femminella, and F. Pugini, "Performance Evaluation of a Measurement-Based Algorithm for Distributed Admission Control in a DiffServ Framework," in *IEEE Globecom*, (San Antonio, Texas), Nov. 2001.

[141] N. Blefari-Melazzi and M. Femminella, "Stateful vs. Stateless Admission Control: Which Can Be the Gap in Utilization Efficiency?," in *IEEE Infocom*, Nov. 2002.

[142] F. Kelly, P. Key, and S. Zachary, "Distributed Admission Control," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2617–2628, 2000.

[143] G. Bianchi, A. Capone, and C. Petrioli, "Throughput Analysis of End-to-End Measurement-Based Admission Control in IP," in *IEEE Infocom*, pp. 1461–1470, 2000.

[144] O. Hagsand, I. Más, I. Marsh, and G. Karlsson, "Self-Admission Control for IP Telephony Using Early Quality Estimation," in 3^{rd}*IFIP-TC6 Networking Conference (Networking)*, (Athens, Greece), pp. 381 – 391, May 2004.

[145] G. Karlsson, H. Lundqvist, and I. M. Ivars, "Single-Service Quality Differentiation," in *IEEE International Workshop on Quality of Service (IWQoS)*, June 2004.

[146] R. Mortier, I. Pratt, C. Clarc, and S. Crosby, "Implicit Admission Control," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, 2000.

[147] S. B. Fredj, S. Oueslati-Boulahia, and J. W. Roberts, "Measurement-Based Admission Control for Elastic Traffic," in 17^{th} *International Teletraffic Congress (ITC)*, (Salvador de Bahia, Brazil), Dec. 2001.

[148] N. Benameur, A. Kortebi, S. Oueslati, and J. W. Roberts, "Selective Service Protection in Overload: Differentiated Services or Per-Flow Admission Control?," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Vienna, Austria), pp. 217 – 222, June 2004.

[149] A. Riedl, T. Bauschert, and J. Frings, "A Framework for Multi-Service IP Network Planning," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Munich, Germany), pp. 183 – 190, June 2002.

[150] A. Riedl, T. Bauschert, and J. Frings, "On the Dimensioning of Voice over IP Networks for Various Call Admission Control Schemes," in 18^{th} *International Teletraffic Congress (ITC)*, (Berlin, Germany), pp. 1311 – 1320, Sept. 2003.

[151] R. Hancock, I. Freytsis, G. Karagiannis, J. Loughney, and S. V. den Bosch, "Next Steps in Signaling: Framework." http://www.ietf.org/internet-drafts/draft-ietf-nsis-fw-05.txt, Oct 2003.

[152] J. Manner, X. Fu, and P. Pan, "Analysis of Existing Quality of Service Signaling Protocols." http://www.ietf.org/internet-drafts/draft-ietf-nsis-signalling-analysis-03.txt, Oct 2003.

[153] J. Wroclawski, "RFC2210: The Use of RSVP with IETF Integrated Services." ftp://ftp.isi.edu/in-notes/rfc2210.txt, Sep. 1997.

[154] L. Berger, D.-H. Gan, G. Swallow, P. Pan, F. Tommasi, and S. Molendini, "RFC2961: RSVP Refresh Overhead Reduction Extensions." http://www.ietf.org/rfc/rfc2961.txt, April 2001.

[155] P. Pan and H. Schulzrinne, "Staged Refresh Timers for RSVP," in *Global Internet Symposium atIEEE Globecom*, Nov. 1997.

[156] L. Wang, A. Terzis, and L. Zhang, "A New Proposal for RSVP Refreshes," in *IEEE International Conference on Network Protocols (ICNP)*, pp. 163–172, 1999.

[157] V. Eramo, U. Mocci, M. Fratini, and M. Listanti, "Reliability Evaluation of RSVP," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, Okt. 1998.

[158] 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), 2002.

[159] M. Karsten, J. Schmitt, and R. Steinmetz, "Implementation and Evaluation of the KOM RSVP Engine," in *IEEE Infocom*, pp. 1290–1299, IEEE, Apr. 2001.

[160] T. Chiueh, A. Neogi, and P. Stirpe, "Performance Analysis of an RSVP-Capable Router," in 4^{th} *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 1998.

[161] G. Fehér, K. Németh, M. Maliosz, I. Czslényi, J. Bergkvist, D. Ahlard, and T. Engborg, "Boomerang - A Simple Protocol for Resource Reservation in IP Networks," in *"IEEE Workshop on QoS Support for Real-Time Internet Applications "*, (Vancouver, Canada), June 1999.

[162] G. Fehér, K. Németh, and I. Czslényi, "Performance Evaluation Framework for IP Resource Reservation Signalling," in 8^{th} *IFIP Working Conference on Performance Modelling and Evaluation of ATM & IP Networks*, July 2000.

[163] G. Fehér, K. Németh, and I. Czslényi, "Performance Evaluation Framework for IP Resource Reservation Signalling," *Performance Evaluation*, vol. 48, pp. 131 – 156, May 2002.

[164] P. Pan and H. Schulzrinne, "YESSIR: A Simple Reservation Mechanism for the Internet," *ACM SIGCOMM Computer Communications Review*, vol. 29, April 1999.

[165] H. Schulzrinne and J. Rosenberg, "Internet Telephony: Architecture and Protocols - an IETF Perspective," *Computer Networks*, vol. 31, no. 3, pp. 237–255, 1999.

[166] L. Berger, L. Delgrossi, D. Duong, and S. Schaller, "RFC1819: Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+." ftp://ftp.rfc-editor.org/in-notes/rfc1819.txt, Aug. 1995.

[167] L. Delgrossi, R. G. Herrtwich, C. Vogt, and L. C. Wolf, "Reservation Protocols for Inter-Networks: A Comparison of ST-II and RSVP," in *Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (Lancaster, UK), pp. 195–203, Nov. 1993.

[168] D. J. Mitzel, D. Estrin, S. Shenker, and L. Zhang, "An Architectural Comparison of ST-II and RSVP," in *IEEE Infocom*, pp. 716–725, 1994.

[169] P. Pan and H. Schulzrinne, "BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations," *Journal of Communications and Networks*, vol. 2, pp. 157–167, June 2000.

[170] O. Schelén and S. Pink, "Aggregating Resource Reservations over Multiple Routing Domains," in *IEEE International Workshop on Quality of Service (IWQoS)*, May 1998.

[171] K. Nichols, V. Jacobson, and L. Zhang, "RFC2638: A Two-Bit Differentiated Services Architecture for the Internet." ftp://ftp.isi.edu/in-notes/rfc2638.txt, July 1999.

[172] A. Terzis, J. Wang, J. Ogawa, and L. Zhang, "A Two-Tier Resource Management Model for the Internet," in *Global Internet Symposium atIEEE Globecom*, Dec. 1999.

[173] Z.-L. Zhang, Z. Duan, Y. T. Hou, and L. Gao, "Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services," in *ACM SIGCOMM* , pp. 71–83, 2000.

[174] B. Teitelbaum, S. Hares, L. Dunn, V. Narayan, R. Neilson, and F. Reichmeyer, "Internet2 QBone: Building a Testbed for Differentiated Services," *IEEE Network Magazine*, Sep. 1999.

[175] Z.-L. Z. Zhang, Z. Duan, and Y. T. Hou, "On Scalable Design of Bandwidth Brokers," *IEICE Transaction on Communications*, vol. E84-B, pp. 2011–2025, Aug 2001.

[176] M. Günther and T. Braun, "Evaluation of Bandwidth Broker Signaling," in *IEEE International Conference on Network Protocols (ICNP)*, pp. 145–152, Nov. 1999.

[177] S. Solhail and S. Jha, "The Survey of Bandwidth Broker," Technical Report, No. UNSW-CSE-TR-0206, School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia, May 2002.

[178] S. Bhatnagar and B. J. Vickers, "Providing Quality of Service Guarantees Using Only Edge Routers," in *IEEE Globecom*, (San Antonio, USA), Nov. 2001.

[179] W. Almesberger, T. Ferrari, and J.-Y. Le Boudec, "SRP: A Scalable Resource Reservation for the Internet," in 6^{th} *IEEE International Workshop on Quality of Service (IWQoS)*, May 1998.

[180] W. Almesberger, T. Ferrari, and J.-Y. Le Boudec, "SRP: A Scalable Resource Reservation for the Internet," *Journal of Computer Communications*, vol. 21, pp. 1200–1211, November 1998.

[181] W. Almesberger, *Scalable Resource Reservation for the Internety*. PhD thesis no. 2051, École Polytechnique Fédérale de Lausanne (EPFL), Nov. 1999.

[182] I. Stoica and H. Zhang, "Providing Guaranteed Services Without Per Flow Management," *ACM SIGCOMM Computer Communications Review*, vol. 29, October 1999.

[183] I. Stoica, *Stateless Core: A Scalable Approach for Quality of Service in the Internet*. PhD thesis no. cmu-cs-00-176, Carnegie Mellon University (CMU), Pittsburg, PA 15213, Dec. 2000.

[184] R. Szábó, T. Henk, V. Rexhepi, and G. Karagiannis, "Resource Management in Differentiated Services (RMD) IP Networks," in *International Conference on Emerging Telecommunications Technologies and Applications (ICETA 2001)*, (Kosice, Slovak Republic), Oct. 2001.

[185] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1048–1056, 1995.

[186] J.-Y. L. Boudec, "Application of Network Calculus to Guaranteed Service Networks," *IEEE Transactions on Information Theory*, vol. 44, May 1998.

[187] J.-Y. L. Boudec and P. Thiran, *Network Calculus*. No. 2050 in Lecture Notes in Computer Science, Springer, 2004. http://ica1www.epfl.ch/PS_files/NetCal.htm.

[188] C. Li, A. Burchard, and J. Liebeherr, "A Network Calculus with Effective Bandwidth," Technical Report, No. CS-2003-20, University of Virginia, Department of Computer Science, Nov. 2003.

[189] X. Xiao and L. M. Ni, "Internet QoS: A Big Picture," *IEEE Network Magazine*, vol. 13, pp. 8–18, Mar. 1999.

[190] B. F. Koch, "A QoS Architecture with Adaptive Resource Control: The AQUILA Approach," in 8^{th} *International Conference on Advances in Communications and Control*, (Crete, Greece), June 2001.

[191] T. Engel, F. Ricciato, S. Salsano, and M. Winter, "Resource Management in QoS Enabled IP Networks with the AQUILA RCL," in 10^{th} *International Conference on Telecommunication Systems, Modeling and Analysis (ICTSM)*, (Monterey, CA, USA), Oct. 2002.

[192] T. Engel, H. Granzer, B. F. Koch, P. Sampatakos, I. S. Venieris, H. Hussmann, F. Ricciato, and S. Salsano, "AQUILA: Adaptive Resource Control for QoS Using an IP-Based Layered Architecture," *IEEE Communications Magazine*, vol. 41, Jan. 2003.

[193] N. Gerlich and M. Menth, "The Performance of AAL-2 Carrying CDMA Voice Traffic," in 11^{th} *ITC Specialist Seminar*, (Yokohama, Japan), Oct. 1998.

[194] M. Menth and N. Gerlich, "A Numerical Framework for Solving Discrete Finite Markov Models Applied to the AAL-2 Protocol," in 10^{th} *GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB)*, (Trier), pp. 163–172, Sep. 1999.

[195] F. Baker, C. Iturralde, F. Le Faucheur, and B. Davie, "RFC3175: Aggregation of RSVP for IPv4 and IPv6 Reservations." http://www.ietf.org/rfc/rfc3175.txt, Sept. 2001.

[196] A. Terzis, L. Zhang, and E. L. Hahne, "Reservations for Aggregate Traffic: Experiences from an RSVP Tunnels Implementation," in 6^{th} *IEEE International Workshop on Quality of Service (IWQoS)*, May 1998.

[197] H. Fu and E. Knightly, "Aggregation and Scalable QoS: A Performance Study," in *IEEE International Workshop on Quality of Service (IWQoS)*, (Karlsruhe, Germany), June 2001.

[198] M. Menth, "A Scalable Protocol Architecture for End-to-End Signaling and Resource Reservation in IP Networks," in 17^{th} *International Teletraffic Congress*, (Salvador de Bahia, Brazil), pp. 211–222, Dec. 2001.

[199] O. Heckmann, J. Schmitt, and R. Steinmetz, "Robust Bandwidth Allocation Strategies," in *IEEE International Workshop on Quality of Service (IWQoS)*, June 2002.

[200] O. Heckmann, J. Schmitt, and R. Steinmetz, "Multi-Period Resource Allocation at System Edges," in 10^{th} *International Conference on Telecommunication Systems, Modeling and Analysis (ICTSM)*, (Monterey, USA), pp. 1–25, Oct. 2002.

[201] T. Li and Y. Rekhter, "RFC2430: A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)." ftp://ftp.isi.edu/in-notes/rfc2430.txt, Oct. 1998.

[202] N. G. Duffield, P. Goyal, A. G. Greenberg, P. P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive, "A Flexible Model for Resource Management in Virtual Private Networks," in *ACM SIGCOMM* , pp. 95 – 108, 1999.

[203] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener, "Algorithms for Provisioning Virtual Private Networks in the Hose Model," in *ACM SIGCOMM* , Aug. 2001.

[204] A. Jüttner, I. Szabó, and Á. Szentesi, "On Bandwidth Efficiency of the Hose Resource Management Model in Virtual Private Networks," in *IEEE Infocom*, April 2003.

[205] S. Bhatnagar and B. Nath, "Distributed Admission Control to Support Guaranteed Services in Core-Stateless Networks," in *IEEE Infocom*, (San Francisco, USA), April 2003.

[206] S. Bhatnagar, B. Nath, and A. Acharya, "Distributed Admission Control for Heterogeneous Multicast with Bandwidth Guarantees," in *IEEE International Workshop on Quality of Service (IWQoS)*, June 2003.

[207] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226–244, June 1995.

[208] J. W. Roberts, "Traffic Theory and the Internet," *IEEE Communications Magazine*, vol. 1, pp. 94 – 99, Jan. 2001.

[209] H. Störmer and et. al., *Verkehrstheorie*. Munich, Germany: Oldenburg Verlag, 1966.

[210] J. S. Kaufman, "Blocking in a Shared Resource Environment," *IEEE/ACM Transactions on Networking*, vol. 29, no. 10, pp. 1474 – 1481, 1981.

[211] J. W. Roberts, "A Service System with Heterogeneous User Requirements – Application to Multi-Service Telecommunications Systems," in *International Conference on Performance Data Communication Systems and Their Applications*, (Amsterdam, The Netherlands), pp. 423 – 431, 1981.

[212] P. Tran-Gia and M. Ritter, *Multi-Rate Models for Dimensioning and Performance Evaluation of ATM Networks - Interim Report of Action COST 242*. June 1994.

[213] W. Bziuk, *Beitrag zur Analyse von Mehrdienste Verlustsystemen mit Bandbreiten-Reservierung*. PhD thesis, Institut für Datentechnik und Kommunikationsnetze, Abteilung Parallelsysteme und Kommunikationsnetze, Technische Universität Braunschweig, 2003.

[214] S. A. Johnson, "A Performance Analysis of Integrated Communication Systems," *British Telecom Technical Journal*, vol. 3, Oct 1985.

[215] J. W. Roberts, "Teletraffic Models for the Telecom 1 Integrated Services Network," in 10^{th} *International Teletraffic Congress (ITC)*, (Montreal), 1983.

[216] P. Tran-Gia and F. Hübner, "An Analysis of Trunk Reservation and Grade of Service Balancing Mechanisms in Multiservice Broadband Networks," in *IFIP Workshop TC6, Modelling and Performance Evaluation of ATM Technology*, (La Martinique), 1993.

[217] P. T.-G. Frank Hübner, "An Analysis of Multi-Service Systems with Trunk Reservation Mechanisms," Technical Report, No. 40, University of Würzburg, Institute of Computer Science, April 1992.

[218] V. B. Iversen, "Teletraffic Engineering and Network Planning, COM Course 34340." http://www.com.dtu.dk/education/34340/material/telenook.pdf, Jan. 2004.

[219] G. B. Dantzig, *Linear Programming and Extensions*. Princeton: Princeton University Press, 1st ed., 1963.

[220] M. Menth, S. Kopf, and J. Milbrandt, "A Performance Evaluation Framework for Network Admission Control Methods," in *IEEE Network Operations and Management Symposium (NOMS)*, (Seoul, South Korea), April 2004.

[221] P. Batchelor et al., "Ultra High Capacity Optical Transmission Networks. Final report of Action COST 239." http://barolo.ita.hsr.ch/cost239/network/, 1999.

[222] 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), June 2004.

[223] M. Menth, S. Kopf, and J. Charzinski, "Impact of Network Topology on the Performance of Network Admission Control Methods," in *IEEE International Workshop on Multimedia Interactive Protocols and Systems (MIPS)*, (Naples, Italy), pp. 195 – 206, Nov. 2003.

[224] M. Menth, S. Kopf, and J. Charzinski, "Network Admission Control for Fault-Tolerant QoS Provisioning," in *IEEE High-Speed Networks for Multimedia Communication (HSNMC)*, (Toulouse, France), June 2004.

[225] M. Menth, S. Gehrsitz, and J. Milbrandt, "Fair Assignment of Efficient Network Admission Control Budgets," in 18^{th} *International Teletraffic Congress*, (Berlin, Germany), pp. 1121–1130, Sept. 2003.

[226] M. Menth, J. Milbrandt, and S. Kopf, "Capacity Assignment for NAC Budgets in Resilient Networks," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Vienna, Austria), pp. 193 – 198, June 2004.

[227] A. M. Jrad, C. K. Chan, T. B. Morawski, and G. O'Reilly, "Incorporating the Downtime Due to Disaster Events in the Network Reliability Model," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Vienna, Austria), pp. 365 – 371, June 2004.

[228] G. O'Reilly, D. J. Houck, E. Kim, T. B. Morawski, D. D. Picklesimer, and H. Uzunalioglu, "Infrastructure Simulations of Disaster Scenarios," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Vienna, Austria), pp. 205 – 210, June 2004.

[229] H. C. Cankaya, A. Lardies, and G. W. Ester, "Network Design Optimization from an Availability Perspective," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Vienna, Austria), pp. 359 – 364, June 2004.

[230] A. Autenrieth and A. Kirstädter, "Engineering End-to-End IP Resilience Using Resilience-Differentiated QoS," *IEEE Communications Magazine*, vol. 40, pp. 50–57, Jan 2002.

[231] L. Sahasrabuddhe, S. Ramamurthy, and B. Mukherjee, "Fault Tolerance in IP-Over-WDM Networking: WDM Protection vs. IP Restoration," *IEEE Journal on Selected Areas in Communications (Special Issue on WDM-Based Network Architectures)*, vol. 20, pp. 21–33, Jan. 2002.

[232] J. L. (ed.), "Link Management Protocol (LMP)." http://www.ietf.org/internet-drafts/draft-ietf-ccamp-lmp-10.txt, Oct 2003.

[233] K. Kompella, P. Pan, N. Sheth, D. Cooper, G. Swallow, S. Wadhwa, and R. Bonica, "Detecting MPLS Data Plane Failures." http://www.ietf.org/internet-drafts/draft-ietf-mpls-lsp-ping-05.txt, Feb. 2004.

[234] I. Gojmerac, F. Hammer, F. Ricciato, H. T. Tran, and T. Ziegler, "Scalable QoS: State-of-the-Art Architectural Solutions and Developments," Technical Report 3, FTW, 2004.

[235] S. Poretsky, "Benchmarking Applicability for IGP Data Plane Route Convergence." http://www.ietf.org/internet-drafts/draft-ietf-bmwg-igp-dataplane-conv-app-02.txt, Jan. 2004.

[236] A. Basu and J. Riecke, "Stability Issues in OSPF Routing," in *ACM SIGCOMM* , (San Diego, CA), Aug. 2001.

[237] G. Iannaccone, C.-N. Chuah, S. Bhattacharyya, and C. Diot, "Feasibility of IP Restoration in a Tier-1 Backbone," *IEEE Network Magazine(Special Issue on Protection, Restoration and Disaster Recovery)*, March 2004.

[238] M. Menth and R. Martin, "Network Resilience through Multi-Topology Routing," Technical Report, No. 335, University of Würzburg, Institute of Computer Science, Mai 2004.

[239] ANSI, "Technical Report on Enhanced Network Survivability Performance," Technical Report, No. ANSI-TI.TR.68, American National Standard for Telecommunications, Feb. 2001.

[240] J. Wang, L. Sahasrabuddhe, and B. Mukherjee, "Path vs. Subpath vs. Link Restoration for Fault Management in IP-over-WDM Networks: Performance Comparisons Using GMPLS Control Signaling," *IEEE Communications Magazine*, vol. 40, pp. 80–87, Nov. 2002.

[241] K. Murakami and H. S. Kim, "Optimal Capacity and Flow Assignment for Self–Healing ATM Networks Based on Line and End-to-End Restoration," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 207–221, Apr 1998.

[242] N. Benameur and J.W. Roberts, "Traffic Matrix Interference in IP Networks," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Munich, Germany), pp. 151 – 156, June 2002.

[243] S. Schnitter and M. Horneffer, "Traffic Matrices for MPLS Networks with LDP Traffic Statistics," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Vienna, Austria), pp. 231 – 236, June 2004.

[244] G. Hasslinger and S. Schnitter, *Telecommunications Network Design and Management*, ch. 7, pp. 125 – 141. Kluwer Academic Publishers, Jan. 2003.

[245] B. Fortz and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," in *IEEE Infocom*, pp. 519–528, 2000.

[246] B. Fortz, J. Rexford, and M. Thorup, "Traffic Engineering with Traditional IP Routing Protocols," *IEEE Communications Magazine*, 2002.

[247] 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.

[248] M. Ericsson, M. Resende, and P. Pardalos, "A Genetic Algorithm for the Weight Setting Problem in OSPF Routing," *Journal of Combinatorial Optimization*, vol. 6, pp. 299–333, 2002.

[249] E. Mulyana and K. U., "An Alternative Genetic Algorithm to Optimize OSPF Weights," in 15^{th} *ITC Specialist Seminar*, (Würzburg Germany), jul 2002.

[250] A. Riedl, "Optimized Routing Adaptation in IP Networks Utilizing OSPF and MPLS," in *IEEE International Conference on Communications (ICC)*, (Anchorage), May 2003.

[251] S. Köhler and A. Binzenhöfer, "MPLS Traffic Engineering in OSPF Networks - A Combined Approach," in 18^{th} *International Teletraffic Congress (ITC)*, (Berlin), 9 2003.

[252] G. Haßlinger and S. Schnitter, "IP Network Expansion for Growing Traffic Demand with Shortest Path Routing Compared to Traffic Engineering," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Vienna, Austria), pp. 81 – 86, June 2004.

[253] O. Heckmann and R. Steinmetz, "On the Elasticity of Traffic Matrices and the Impact on Capacity Expansion," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, (Vienna, Austria), pp. 223 – 229, June 2004.

[254] I. Gojmerac, T. Ziegler, F. Ricciato, and P. Reichl, "Adaptive Multipath Routing for Dynamic Traffic Engineering," in *IEEE Globecom*, (San Francisco), Nov 2003.

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

[256] 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), pp. 727–735, 2002.

[257] M. S. Kodialam and T. V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering," in *IEEE Infocom*, vol. 2, pp. 884–893, Mar 2000.

[258] G. Li, D. Wang, C. Kalmanek, and R. Doverspike, "Efficient Distributed Path Selection for Shared Restoration Connections," in *IEEE Infocom*, 2002.

[259] 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), 9 2003.

[260] R. R. Iraschko, M. H. MacGregor, and W. D. Grover, "Optimal Capacity Placement for Path Restoration in STM and ATM Mesh-Survivable Networks," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 328 – 336, June 1998.

[261] K. Murakami and H. S. Kim, "Comparative Study on Restoration Schemes of Survivable ATM Networks," in *IEEE Infocom*, (Kobe City, Japan), pp. 345 – 352, April 1997.

[262] J. Strand, A. L. Chiu, and R. Tkach, "Issues for Routing in the Optical Layer," *IEEE Communications Magazine*, vol. 39, pp. 81–87, Feb 2001.

[263] B. Rajagopalan, J. V. Luciani, and D. O. Awduche, "IP over Optical Networks: A Framework." http://www.ietf.org/internet-drafts/draft-ietf-ipo-framework-05.txt, Sep 2003.

[264] K. Kompella and Y. Rekhter, "Routing Extensions in Support of Generalized Multi–Protocol Label Switching." http://www.ietf.org/internet-drafts/draft-ietf-ccamp-gmpls-routing-09.txt, Oct 2003.

[265] M. Menth, A. Reifert, and J. Milbrandt, "Self-Protecting Multipaths - A Simple and Resource-Efficient Protection Switching Mechanism for MPLS Networks," in 3^{rd} *IFIP-TC6 Networking Conference (Networking)*, (Athens, Greece), pp. 526 – 537, May 2004.

[266] K. Kar, M. S. Kodialam, and T. V. Lakshman, "Routing Restorable Bandwidth Guaranteed Connections Using Maximum 2-Route Flows," in *IEEE Infocom*, Jun 2002.

[267] W. D. Grover, "Cycle-Oriented Distributed Preconfiguration: Ring-Like Speed with Mesh-Like Capacity for Self-Planning Network Restoration," in *IEEE International Conference on Communications (ICC)*, pp. 537–543, Jun 1998.

[268] W. D. Grover and D. Stamatelakis, "Bridging the Ring-Mesh Dichotomy with p-Cycles," in *International Workshop on the Design of Reliable Communication Networks (DRCN)*, Apr 2000.

[269] W. D. Grover, J. Dourcette, and M. Cloqueur, "New Options and Insights for Survivable Transport Networks," *IEEE Communications Magazine*, no. 1, 2002.

[270] D. Stamatelakis and W. D. Grover, "IP Layer Restoration and Network Planning Based on Virtual Protection Cycles," *IEEE Journal on Selected Areas in Communications*, vol. 18, Oct. 2000.

[271] D. Staatelakis and W. D. Grover, "Theoretical Unterpinnings for the Efficiency of Restorable Networks Using Preconfigured Cycles ("P-Cycles")," *IEEE/ACM Transactions on Networking*, vol. 48, Aug. 2000.

[272] D. A. Schupke, C. G. Gruber, and A. Autenrieth, "Optimal Configuration of p-Cycles in WDM Networks," in *IEEE International Conference on Communications (ICC)*, (New York), 2002.

[273] C. G. Gruber and D. A. Schupke, "Capacity-Efficient Planning of Resilient Networks with p-Cycles," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, pp. 389–395, Jun 2002.

[274] J. W. Suurballe, "Disjoint Paths in a Network," *Networks Magazine*, vol. 4, pp. 125–145, 1974.

[275] J. Edmonds and R. M. Karp, "Theoretical Improvements in the Algorithmic Efficiency for Network Flow Problems," *Journal of the ACM*, vol. 19, pp. 248–264, Apr 1972.

[276] M. Menth, A. Reifert, and J. Milbrandt, "Optimization of End-to-End Protection Switching Mechanisms for MPLS Networks," Technical Report, No. 320, University of Würzburg, Institute of Computer Science, Feb. 2004.

[277] M. Pióro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan and Kaufman, June 2004.

[278] ILOG, Inc., www.cplex.com, *CPLEX*.

[279] A. Makhorin, *GNU Linear Programming Kit Reference Manual Version 4.0*. Free Software Foundation, Inc., 59 Temple Place — Suite 330, Boston, MA 02111, USA, May 2003.

[280] H. Saito and M. Yoshida, "An Optimal Recovery LSP Assignment Scheme for MPLS Fast Reroute," in *International Telecommunication Network Strategy and Planning Symposium (Networks)*, pp. 229–234, Jun 2002.

[281] D. A. Dunn, W. D. Grover, and M. H. MacGregor, "Comparison of k–Shortest Paths and Maximum Flow Routing for Network Facility Restoration," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 1, pp. 88–99, 1994.

[282] D. Sidhu, R. Nair, and S. Abdallah, "Finding Disjoint Paths in Networks," in *ACM SIGCOMM* , 1991.

[283] S. Bahk and M. E. Zarki, "Dynamic Multi-Path Routing and How it Compares with Other Dynamic Routing Algorithms for High-Speed Wide Area Network," in *ACM SIGCOMM* , March 1992.

[284] M. Klein, "A Primal Method for Minimal Cost Flows with Applications to the Assignment and Transportation Problems," *Management Science*, vol. 14, pp. 205 – 220, 1967.

[285] J. W. Suurballe and R. E. Tarjan, "A Quick Method for Finding Shortest Pairs of Disjoint Paths," *Networks Magazine*, vol. 14, pp. 325–336, 1984.

[286] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, and C.-N. Chuah, "Characterization of Failures in an IP Backbone," in *IEEE Infocom*, (Hongkong), March 2004.

[287] M. Menth, J. Milbrandt, and A. Reifert, "Sensitivity of Backup Capacity Requirements to Traffic Distribution and Resilience Constraints," Technical Report, No. 322, University of Würzburg, Institute of Computer Science, Feb. 2004.

[288] B. M. Waxman, "Routing of Multipoint Connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, 1988.

List of Abbreviations

A-MBAC	MBAC with aggregate measurements, page 45
AAL2	ATM Adaptation Layer type 2, page 61
AC	admission control, page 2
AL	application layer, page 9
AS	autonomous system, page 18
ASN	AS number, page 19
ATM	Asynchronous Transfer Mode, page 2
BAM	budget assignment method, page 136
BB	bandwidth broker, page 56
BBB	b2b budget, page 48
BGP	Border Gateway Protocol, page 26
BMBF	Bundesministerium für Bildung und Forschung, page 38
BNAC	budget-based NAC, page 3
CAPEX	capital expenses, page 1
CIDR	Classless Interdomain Routing, page 21
CM	construction method, page 109
CNBA	concurrent NBA, page 136

CR-LDP Constraint-Based LDP, page 28

CRBA concurrent RBA, page 140

CS complete sharing, page 69

DHCP Dynamic Host Configuration Protocol, page 17

DiffServ Differentiated Services, page 32

DNS Domain Name System, page 18

DSCP DiffServ Code Point, page 32

DSL Digital Subscriber Line, page 11

E equal load balancing, page 158

E-BGP Exterior BGP, page 27

e2e end-to-end, page 2

EB egress budget, page 48

EBAC experience-based AC, page 44

ECMP Equal Cost Multi-Path, page 25

EDF Earliest Deadline First, page 34

EGP exterior gateway routing protocol, page 23

ELB egress link budget, page 48

EQOS Edge Assisted QoS, page 57

F-MBAC MBAC with flow measurements, page 45

FDDI Fiber Distributed Data Interface, page 11

FEC forwarding equivalent class, page 147

FIFO First-In-First-Out, page 33

FLBA fair LBA, page 127

FNAC feedback-based NAC, page 46

FTP File Transfer Protocol, page 18

FWP framework program, page 38

GPS Generalized Processor Sharing, page 33

HDLC High Level Data Link Control Protocol, page 10

HTTP Hypertext Transfer Protocol, page 8

I-BGP Interior BGP, page 27

IB ingress budget, page 48

ICANN Internet Corporation for Assigned Names and Numbers, page 19

ICMP Internet Control Message Protocol, page 12

IETF Internet Engineering Task Force, page 8

IGP interior gateway routing protocol, page 23

ILB ingress link budget, page 48

ILM incoming label map, page 27

INBA independent NBA, page 134

IP Internet Protocol, page 1

IPv4 IP version 4, page 13

IPv6 IP version 6, page 13

IRBA independent RBA, page 140

IS-IS Intermediate System to Intermediate System Routing Exchange Protocol, page 25

ISO International Standardization Organization, page 10

ISP Internet service provider, page 2

IST information society technologies, page 38

ITU International Telecommunication Union, page 18

IXP Internet Exchange Point, page 20

kDSP k disjoint shortest paths, page 157

KING Key Components for the Internet of the Next Generation, page 38

LAC link AC, page 2

LB link budget, page 48

LBA link budget assignment, page 126

LDP Label Distribution Protocol, page 28

LIB label information base, page 148

LL link layer, page 10

LLC logical link control, page 9

LP linear program, page 4

LSP label-switched path, page 27

LSP link state package, page 24

LSR label switching router, page 27

MAC media access control, page 10

MBAC measurement-based AC, page 43

MEDF Modified EDF, page 34

MIB management information base, page 28

MP multi-path, page 106

MPLS Multiprotocol Label Switching, page 3

MT minimum traffic, page 157

MTU Maximum Transfer Unit, page 12

NAC network AC, page 2

NAP Network Access Point, page 20

NAT Network Address Translation, page 13

NBA network budget assignment, page 126

NCS network control server, page 63

NGN Next Generation Network, page 1

NL network layer, page 9

NSIS Next Steps in Signaling, page 49

O optimized load balancing, page 159

OPEX operational expenses, page 1

OPT optimum multi-path backup structure, page 158

OPWA one-pass with advertising, page 53

OSI Open System Interconnection, page 10

OSPF Opens Shortest Path First, page 24

PHB Per-Hop Behavior, page 32

PL physical layer, page 10

PLAC parameter-based LAC, page 43

PLBA proportional LBA, page 127

POP Point of Presence, page 20

PP Path Protection, page 154

PPP Point-to-Point Protocol, page 10

PS PATH state, page 52

QoS quality of service, page 1

R reciprocal load balancing, page 159

RBA resilient budget assignment, page 126

RED Random Early Detection, page 33

REM rate envelope multiplexing, page 43

RFC Request for Comments, page 8

RIP Routing Information Protocol, page 24

RMD Resource Management in Differentiated Services IP Networks, page 58

RR receiver report, page 54

RS RESV state, page 52

RSVP Resource Reservation Protocol, page 16

RSVP-TE RSVP Tunneling Extensions, page 28

RTCP RTP Control Protocol, page 16

RTP Real-Time Transport Protocol, page 16

RTSP Real-Time Streaming Protocol, page 17

SDH Synchronous Digital Hierarchy, page 12

SIP Session Initiation Protocol, page 17

SLA service level agreement, page 32

SMTP Simple Mail Transfer Protocol, page 18

SP Static Priority, page 33

SP shortest path, page 4

SP single-path, page 106

SPM Self-Protecting Multi-Path, page 4

SR sender report, page 54

SRLG Shared Risk Link Group, page 153

SRP Scalable Resource Reservation Protocol, page 58

ST2 Internet Stream Protocol version 2, page 54

TCA traffic conditioning agreement, page 32

TCP Transmission Control Protocol, page 15

TL transport layer, page 9

ToS Type of Service, page 12

TR trunk reservation, page 69

TTL Time-to-Live, page 12

UDP User Datagram Protocol, page 15

UMTS Universal Mobile Telecommunication System, page 12

URL uniform resource locator, page 8

UTRAN UMTS Terrestrial Radio Access Network, page 12

VCC Virtual Channel Connections, page 34

VPC virtual path connection, page 61

VPN Virtual Private Network, page 29

WFQ Weighted Fair Queuing, page 33

WLAN Wireless Local Access Network, IEEE 802.11, page 11

WRR Weighted Round Robin, page 33

YESSIR YEt another Sender Session Internet Reservation, page 54

ISSN 1432-8801