An optical packet network based on Arrayed Waveguide Gratings

vorgelegt von
Diplom-Informatiker
Hagen Woesner
aus Berlin

Von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Klaus Petermann
Berichter: Prof. Dr.-Ing. Adam Wolisz
Berichter: Prof. Dr. Harmen R. van As

Tag der wissenschaftlichen Aussprache: 6. Juni 2003

Berlin 2003
D 83

Abstract

This work presents PrimeNet, a novel architecture for optical multihop networks and investigates a Medium Access Control (MAC) protocol and a fairness algorithm for it. The network architecture is based on an Arrayed Waveguide Grating, a passive optical device that is used widely as wavelength multiplexer/demultiplexer, today. Relying on a physical star architecture logical rings are set up on each wavelength. The number of nodes N has to be a prime number to lead to $\frac{(N-1)}{2}$ pairs of counterdirectional rings. It is possible to start the deployment of the network with a single ring, and to add further rings when the demand increases. With the addition of new rings the mean hop distance in the multihop network decreases from N/2 down to 1, which is the full mesh.
The capacity of this multihop network is compared to a single-hop network. It shows that a small number of fixed transmitters and receivers per node (2 to 3, mostly) are enough to gain the same capacity as a single-hop network with one tunable transmitter/receiver pair. On the basis of the proposed multi-ring architecture a node structure, a MAC protocol, and a fairness algorithm are developed and evaluated analytically and by simulation. An estimation of some physical parameters shows that the network is suitable for the local and metropolitan area. The MAC protocol is based on a carrier-sensing and a fiber delay line (FDL) large enough to store a full packet in order to allow for an unslotted, immediate access to the medium. An aggregation of smaller packets to so-called "Jumbo"-frames helps to reduce the overhead for optical switching. Because of the potentially very small mean hop distance, we favor local fairness algorithms over global. Among three candidates, the back-pressure based Spatial Reuse Protocol (SRP) was chosen for the fairness algorithm. It had to be modified to suit the optical node architecture. The simulations of the fairness algorithm were performed using unidirectional traffic with a packet size distribution that is typical for today's Internet. While a fair access to the medium can obviously be guaranteed for this kind of traffic, a modeling of "real" TCP revealed interactions between the MAC protocol and TCP that lead to unfairness for certain TCP connections. With the introduction of a head-of-line timer to avoid the blocking of the *slow start mechanism* of TCP, fairness can be guaranteed. Another problem arising from the optical node architecture are reorderings of "Jumbo" frames. These lead to spurious retransmissions of TCP segments. A discussion of possibilities to make TCP robust against packet reorderings concludes the chapter.
At last we give an outlook on the design of large networks based on PrimeNets. Every AWG-based multihop network can be seen as a permutation or Cayley graph. This family of graphs incorporates many of known regular graphs, such as the ring, the hypercube, or the star graph.

PrimeNet – Ein optisches Packetnetz auf Basis eines Arrayed Waveguide Gratings

Die vorliegende Arbeit beschreibt eine neuartige Architektur optischer Multihop–Packetnetze und untersucht daraus hervorgehende Zugriffsprotokolle und Fairnessalgorithmen.
Die Netzarchitektur basiert auf einem $(N \times N)$ AWG, einem passiven Bauelement, das ähnlich einem Prisma in der Lage ist, Wellenlängen zu demultiplexen bzw. zu multiplexen, allerdings mit einer zyklischen Vertauschung der Wellenlängen an den Ausgängen. Durch diese Vertauschung sind keinerlei Kollisionen von Signalen im Bauelement möglich und dieselbe Wellenlänge lässt sich von allen angeschlossenen Stationen gleichzeitig verwenden, was zu einer Vervielfachung der Bandbreite gegenüber einer passiven Sternkoppler-Architektur führt. Auf Basis einer physischen Sternarchitektur werden nunmehr logische Ringe auf jeder Wellenlänge gebildet. Es wird gezeigt, dass die Anzahl der Knoten im Netz eine Primzahl sein muss, damit alle Knoten auf allen (N-1) Ringen liegen. Der Ausbau des Netzes kann mit nur einer Wellenlänge (und nur einem Sender/Empfängerpaar pro Knoten) beginnen, um nach Bedarf weitere Ringe zu installieren. Die Reihenfolge der Knoten in den Ringen ist aber unterschiedlich, was bei einem vollen Ausbau zur Totalvermaschung des Netzes führt.
Die Kapazität des Netzes wird analytisch als Funktion der Anzahl der Ringe berechnet. Es wird gezeigt, dass die totale Netzkapazität mehr als quadratisch mit der Zahl der Ringe wächst. Ein Vergleich mit einer Single-Hop-Architektur ergibt, dass im allgemeinen eine sehr geringe Anzahl (2-3) fester Sender/Empfängerpaare ausreicht, um dieselbe Kapazität wie mit einem Paar abstimmbarer Sender/Empfänger pro Knoten zu erreichen.
Auf Basis der sich überlagernden Ringe wird dann eine Knotenarchitektur entwickelt, die ein einfaches Zugriffsprotokoll ermöglicht. Ein Vielfachzugriffsverfahren basierend auf einer optischen Verzögerungsleitung und einer elektronischen Auswertung des Packet-*Headers* wird daraufhin simulativ untersucht. Um Zeit für die Auswertung des Headers und das optische Schalten zu gewinnen, werden kleinere Packete zu sogenannten "Jumbo"-Rahmen aggregiert. Zur Gewährleistung der Fairness beim Zugriff auf den Ring wird ein lokales Verfahren benutzt, das aus dem SRP-Protokoll hervorging. Dieses Verfahren musste an die optische Knotenarchitektur angepasst werden. Eine simulative Untersuchung zeigt Schwachstellen des Verfahrens auf, wenn statt eines unidirektionalen Datenverkehrs das tatsächliche Verhalten des TCP-Transportprotokolls nachgebildet wird. Die Einführung eines Alarmgebers, der die Wartezeit des ersten Segments in der Warteschlange überwacht, führt zu einem fairen Zugriff aller Knoten auf das Medium.
Weitere Probleme treten durch das Umordnen von Aggregaten im Netz auf. TCP reagiert aufgrund des *fast-retransmit*–Mechanismus' mit einer schnellen Wiederholung verloren *geglaubter* Packete und einer Reduktion der Senderate. Eine Diskussion der Möglichkeiten, TCP robust gegen solche Fehler zu machen, beschließt das Kapitel.
Der letzte Teil der Arbeit bietet einen Ausblick auf Möglichkeiten, größere Netze auf PrimeNet-Basis zu entwerfen. Hierfür bieten sich die sogenannten Cayley-Graphen an, Permutationsgraphen, deren Eigenschaften wie maximale Fehlertoleranz und einfaches Routing von vornherein bekannt sind.

Contents

List of Tables

List of Figures

1. Introduction

1.1. Optical Networks as of Today

When looking at the exponential growth of data networks within the last years, one can observe two driving forces for it: The number of nodes attached to the Internet as the largest worldwide data network grows as well as the data rate per node. Both these trends together add to a huge amount of bandwidth that is required within the backbone of the network. Optical data transmission is the natural candidate to reach this goal of transmitting high volumes of data with low latency.
Unfortunately, the Internet traffic is not only high in volume but also highly dynamic. This means that data flows appear and disappear within milliseconds which makes it inefficient to set up connections for every data flow. In result, datagram switched networks have proven to be advantageous over the classical circuit switched approach. These datagrams carry their own destination address (among other information) in a *header* which has to be evaluated at every node in the network to determine the link on which the datagram has to leave the node. Consequently, a huge number of these evaluation operations have to be performed for every incoming link in a node. The contrast between the low electronic port processing speed (currently around 10 Gbit/s) and the high possible bandwidth of a single fiber (several 10 Tbit/s) is commonly called the electro-optical bottleneck. Over the last few years Wave Division Multiplexing (WDM) has been seen as a proper workaround for this situation. Here, the transmission spectrum of an optical fiber is partitioned into wavelength channels that carry e.g. 10 Gbit/s each and thus fully serve the electronic equipment. A further increase of the port processing speed depends on new concepts either in the design of the electronic switches or in the packet switching techniques. This work aims at the latter, namely an *Optical Packet Switching (OPS)*.

1.2. Motivation and Scope

The aim of OPS is to increase the amount of data that is transported on a single wavelength channel and to reduce the cost of the equipment by avoiding the O/E/O (optical/electronic/optical) conversion of each datagram in every node. Instead, the header information is evaluated *somehow* optically and the datagram is sent through a number of optical switches (meanwhile possibly changing its wavelength) towards the outgoing fiber. The main difference between an electrical and an optical packet switching is the impossibility to store light pulses infinitely, that is, the lack of an optical Random Access Memory (RAM).

OPS architectures therefore have to be different from conventional packet switching architectures that rely on the *store-and-forward* concept. Because optical data processing is not yet in the state of technological maturity (and simply because an electronic processing is much cheaper) the evaluation of the header information is done electronically in most of the proposed OPS networks. This is accomplished by extracting and converting, e.g. 10% of the optical power of a signal before it enters the node and utilizing the electrical signal to evaluate the header.
A number of combined WDM/OPS architectures have been proposed for future networks, mostly for Local and Metropolitan area networks. The fact that the Wide Area Network (WAN) is continuously considered to be circuit (i.e. wavelength) switched is due to the large number of connections that are multiplexed onto a single circuit (the so-called multiplexing factor). This makes it reasonable to switch wavelengths rather than single datagrams within the backbone of a large data network. Currently, much effort is being put into the development of fast setup and reconfiguration of wavelength paths through the backbone network, mostly in the context of Generalized Multi-Protocol Label Switching (GMPLS).
In the periphery of the network, however, the multiplexing factor is much lower. Here, ways to share the medium between the attached nodes are explored. Doing so, the number of transmissions sharing a single wavelength can be increased again, leading to an improved use of the available capacity. The sharing of the medium (here: a single wavelength or the set of all wavelengths) requires a set of rules, commonly called a Medium Access Control (MAC) protocol. The variety of MAC protocols for WDM/OPS architectures that have been proposed over the last years can be grouped into two categories: *single-hop* and *multihop* networks. While in single-hop networks two nodes communicate directly, in a multihop network a datagram has to be forwarded by intermediate nodes. Multihop networks generally use the medium less efficiently, because the transmission of a datagram occupies more than one link in the network. On the other hand, the number of input/output links per node (the *degree* of the node) can be kept low, resulting in a simpler node structure. In addition, the transmitters and receivers in a multihop hop network do not have to be able to tune between wavelengths very rapidly which is the greatest technological challenge for single-hop networks. Because of these reasons, it was decided to follow the multihop way within this work.
Rings are natural candidates for multihop networks. In fact, a unidirectional ring is the simplest possible form of a multihop network. The parameter that is most important for the calculation of the capacity of a multihop network is the mean hop distance, i.e. the number of links a datagram has to traverse in average from source to destination. The reduction of the mean hop distance in ring networks is possible when multiple rings are used concurrently to interconnect a number of nodes. Before transmitting a datagram, a node has to choose the ring with the lowest number of hops towards the destination. The part of the ring that is not affected by the transmission may be used for other – parallel – transmissions. This *spatial reuse* of the rings requires the ability to take a packet off the ring, whenever the node is the destination of that packet. However, the large gain in capacity allied with the *spatial reuse* does not come for free: The information about the incoming

packet has to be processed in real time. For a 40 byte short Internet Protocol (IP) packet as it is observed very often in recent traffic measurements, at a line rate of 10 Gbit/s only about 30 nanoseconds remain for the extraction and processing of the header information. To relax this speed requirement, two main strategies may be followed: *time slotting* the channel and *aggregation* of small packets to larger ones.
A *time slotted* channel on one hand reduces the amount of information that has to be processed in real time. The decision about a reception or transmission in a time slot can be taken by simply counting time slots or by a *free/busy* bit in the header of the slot. On the other hand, a reservation phase is needed prior to the transmission to make the information about the following slots public. These systems usually show a cyclic alternation of reservation and data transmission periods. Hence, the access delay to the channel increases for a node that has to wait for its reserved bandwidth.
To ensure a collision-free access to the ring in an *un-slotted* system, some kind of carrier sensing (CS) is necessary. When using a Fiber Delay Line (FDL) in each node, it is possible to delay incoming datagrams until the decision about their destination is taken. This way it is possible to leave a datagram on the ring (and in the optical domain) if it is not destined for the node. A node should refrain from transmission whenever it "senses" a packet arriving in the FDL.
The *aggregation* of smaller packets to larger ones extends the time that is available for the evaluation of the header information. A packet classifier is needed – possibly in conjunction with a number of (virtual) output queues per node – to assure that all packets contained in a larger aggregate have the same destination address and Quality of Service (QoS) requirements.
Within the work presented here a multi-ring network architecture is developed, that can be gradually expanded from a unidirectional ring to a fully meshed network. The capacity of such a multi-ring network grows more than quadratically with the number of rings in use. To achieve this, we exploit the passive wavelength routing capability of a particular optical device, the Arrayed Waveguide Grating (AWG), an optical component that is used widely as wavelength multiplexer/demultiplexer, today. Using this device in a physical star topology, one logical ring may be set up on each wavelength. These rings allow for *spatial reuse* and decrease the mean hop distance in the network.
The decrease in the mean hop distance influenced the choice of the MAC protocol. It was decided to avoid any reservation of bandwidth resources, because the number of nodes that are affected by any transmission becomes smaller with the decreasing mean hop distance. It is therefore useless to bother all nodes with the processing of reservation requests not influencing them anyhow. In other words: *local fairness* becomes more important than *global fairness* when the influence of a transmission is local rather than global. A *back-pressure* based MAC similar to the Spatial Reuse Protocol (SRP) [TS00] proposed in 2000 was therefore considered more appropriate for this architecture. Here, *back-pressure* means that a congested node is able to throttle other (upstream) nodes that are responsible for the congestion.

The SRP protocol was designed for an *insertion buffer* architecture and requires a number of (electrical) packet queues. Because of the lack of optical RAM this approach is inherently

not usable for OPS. Following an optical node architecture employing an FDL like the one described above, the SRP had to be modified. The investigation of the inter-working of the new MAC and the Transmission Control Protocol (TCP), the transport protocol dominating the Internet today, makes up a large part of the work presented here.

1.3. Outline of the Dissertation

Within the next chapter a short introduction into the physical aspects of components for optical networks will be given that includes a special section devoted to the Arrayed Waveguide Grating Multiplexer.
The following sections outline the directions in WDM network development that can be observed today. Traditionally, WDM networks have been recognized as circuit switched networks, because of long laser tuning times and the relatively static nature of the wavelength paths that can be switched in such a network. The main problem of Wavelength Routed Networks (WRN) is the Routing and Wavelength Assignment (RWA) (see section 3.3.1).
Multiplexing of connections onto the wavelengths is almost exclusively done using the Synchronous Optical Network (SONET)/Synchronous Digital Hierachy (SDH) infrastructure that is being laid out on top of the optical (WDM) layer. Because these networks were built to transport voice traffic, new concepts had to be developed for packet data networks like the Internet. A number of these, like IP over Asynchronous Transfer Mode (ATM), Packet over SONET (PoS), or Multiple Access Protocol Over SONET (MAPOS) will be introduced. To make the network more responsive to changes in the traffic patterns, the lifetime of the circuits is decreased using technologies like Multi-Protocol Label Switching (MPLS) and Optical Burst Switching (OBS). Finally we arrive at OPS networks.
After this introduction, three main blocks follow: A novel multihop network concept, the analysis of its capacity as a function of the mean hop distance and a simulative evaluation of the MAC and fairness protocol.

The multihop network concept developed in chapter 8 is aimed at the cost-effective support of connection-less transmission of optical packets. The main component that is considered here is the AWG. Using this passive wavelength router, it is possible to set up ring networks that connect all stations attached to the AWG. When the number of inputs to the AWG is a prime number, it is possible to set up one ring on each wavelength that is used in the network. Because of the need for prime numbers, the resulting network is called PRIMENET. The network is of the FT^r/FR^r type (r fixed transmitters and r fixed receivers per node) and works in a multihop fashion. The number of rings r is variable from 1 (the unidirectional ring) to $(N-1)$ for N nodes. The latter case can be seen as a full mesh and therefore converges to the single-hop network. To assess the feasibility of such a network, a simple calculation of the signal attenuation and the noise figure of a single hop and cascaded hops are performed. We show that the network is feasible provided that additional amplifiers compensate the insertion loss of the AWG and the attenuation of the fiber.

An analysis of the mean hop distance and the total network capacity as a function of the number of rings in use shows the superiority of PRIMENET over other multi-ring architectures. A comparison between single- and multihop architectures based on the AWG shows that the number of fixed transmitter/receiver pairs (FT/FR) that are necessary in a multihop network to achieve the same total capacity as a TT/TR (tunable/tunable) single-hop network is relatively low. This means that with a given budget and today's components it is in most cases advantageous to opt for the multihop architecture. Following this, in section 9.4 two different routing strategies in PRIMENET are compared. While the sequential transmission of packets over the shortest path maximizes the network capacity under high load, it is possible to decrease the transmission delay for a given flow by parallel transport of the packets over all available paths. The switching point between both strategies is calculated as a function of the background load.
To complete the analysis of the network, two possible logical node architectures are compared in section 9.5. Assuming a full wavelength conversion in each node it is possible to find shorter paths through the network while drastically increasing the complexity of the architecture. It is shown that the increase of the capacity is larger for low numbers of wavelengths per node. Because the switching of packets between wavelengths requires long inter-frame times and/or optical buffers, the wavelength-converting node is taken as a rather hypothetical, but somehow ideal benchmark for PRIMENET's simple node architecture without any wavelength conversion.
Having a network concept that is potentially superior to conventional WDM ring architectures and single-hop networks the next step in chapter 10 is to design an access protocol that allows to exploit this feature. A discussion of possibilities of separating the header of an optical packet from the payload that is to be transmitted untouched through the network leads to a physical architecture for a node to access the multi-ring network. The general concept assumes an electronic evaluation of the header and the setting of a simple, but fast, 2×2 optical switch, while the payload is temporarily buffered in an FDL. The basic access mechanism is a Carrier Sense Multiple Access (CSMA). A node is only allowed to start transmitting as long as the FDL is empty. The FDL architecture leads to a fixed packet size. Destination stripping is employed to make use of the low mean hop distance in the network.

A evaluation of the access mechanism performed in section 10.3 shows the need for a fairness mechanism. Without it, nodes suffer from traffic that is generated by upstream nodes. The mechanism that is employed is similar to the SRP that was the basis for the current development of the IEEE 802.17 Resilient Packet Ring (RPR) standard. It is chosen because of its aggressive way of bandwidth acquisition without a reservation process. The idea is to throttle upstream nodes using special packets whenever a downstream nodes detects congestion.
While SRP determines congestion by a threshold in the insertion buffer, this is not possible here because of the FDL (an insertion buffer of length 1). Therefore, a first approach is to monitor the transmission queue in a node. Whenever this queue is filled above a certain threshold, a *usage packet* is sent upstream that reduces the traffic of the upstream nodes.

It is shown in simulations using artificial Contant Bit Rate (CBR) and IP traffic that this mechanism leads to fair access on the rings.
However, modeling the packet length distribution of real traffic may not be enough to evaluate the dynamic behavior of a system. Therefore, "real" TCP connections are simulated over the ring network. Much to our surprise, there are situations where fairness can not be achieved using the above mechanism of defining congestion in a node. Therefore, a Head-Of-Line (HOL)-timer is introduced that leads to a timeout whenever the first packet (here: TCP segment) in the transmission queue waits for too long. This timeout is then taken to signal congestion. The decision about the start value of the HOL timer is made depending on the mean load the node was allowed to source onto the ring within the last period of time. Under certain network and traffic topologies fair access to the ring is achieved also for TCP connections using the modified congestion detection mechanism.
Another feature of the MAC protocol developed earlier is the possible reordering of packets that belong to a certain TCP connection. It is shown how this affects TCP, again in contrast to User Datagram Protocol (UDP)-like unidirectional traffic. A discussion of possible strategies to limit the effect of the reordering concludes this chapter.

An outlook is given in chapter 11 on optical multihop networks based on Cayley graphs. This family of graphs is based on permutation groups and shows a number of desirable properties like vertex transitivity and a closed form for the routing. PRIMENET itself can be considered a Cayley graph. Some Cayley graphs that interconnect different PRIMENETs are introduced and compared to known graphs like the ShuffleNet.

2. WDM - Wave Division Multiplexing - Physics and Components

2.1. Introduction

This chapter is intended to give an introduction into the the basic concepts of optical transmission and wave division multiplexing. For more details see [Muk97],[RS98] and [BJB+97], although we will give a survey of the physical phenomena and the basic building blocks of optical networks here, since it is necessary for the fluent understanding of the following.
For the transmission of optical signals three wavelength bands at around 850 nm, 1300 nm and 1550 nm are being used, where the attenuation is about 0.5 dB per kilometer. The peak in the attenuation around 1400 nm occurs due to water impurities (OH-ions) in the fiber (see Fig. 2.1).
Each of the frequency bands offers approximately 15 THz of bandwidth. Thus, we have a total of 50 THz of bandwidth in a single fiber (which today corresponds to around 50 Tbit/s, depending on the modulation scheme) compared to the electronic transmission speed of currently around 10 Gbit/s. This situation is called the electro-optical bottleneck. The most popular way to deal with it is the subdivision of the optical spectrum into a number of wavelength channels. This is called Wave Division Multiplexing (WDM). The International Telecommunication Union (ITU) has standardized a frequency grid which defines the central frequencies of the WDM channels. Their spacing is either 50 GHz, 100 GHz or 200 GHz.
In the following chapters we will explain the basic principles of optical signal transmission. After a short introduction into properties of lightwaves, the components in the optical path will be explained. Special attention is then paid to the components that are needed for a packet switched WDM system. Fast optical filters, directive switches and the arrayed waveguide grating(AWG) will be explained shortly. Chapter 2.11 is particularly devoted to the latter, since most of the following work is based on AWGs.

2.2. Some Phenomena of Optical Transmission in Fiber

Whenever we speak about optical data transmission, a few fundamental principles and features of lightwaves apply. These are:

- Optical interference – As all electromagnetic waves do, lightwaves interfere with each other. Depending on their phase difference, this interference can be either destructive or constructive. This interference, however, appears only at the receiver. Within

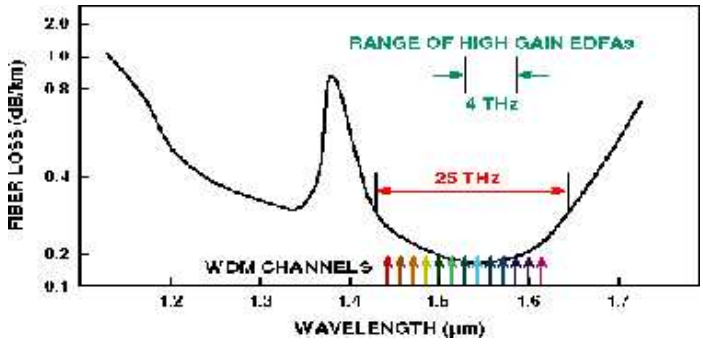

Figure 2.1.: The optical windows at 1300 and 1550 nm. (the 850nm is not shown here.) from [Con96]

the fiber, light waves travel just like any other electromagnetic wave travel without influencing each other.

- Stimulated Emission – Each atom has a discrete number of energy levels that an electron can reside on. When it absorbs energy (by the means of light, microwaves or electrical current), the atom becomes excited, i.e. the electron moves to a higher level. When going back to the ground level, the electron releases a photon. There are chemical elements whose energy levels are quasi-stable, and the phenomenon of *population inversion* occurs, when energy is applied. This means that there are more electrons in the excited state than in the ground state and consequently, that these elements are able to emit more light than they absorb. Stimulated emission occurs when a photon passes very closely to an excited electron. This causes the electron to release another photon which is of the same direction and coherency (frequency) than the first.

- Evanescent coupling – The part of a propagating wave that travels along or outside of the waveguide boundary is called the evanescent wave. If two waveguides are arranged in close proximity over a characteristic length, the lightwave moves from one waveguide into the other and then back. The amount of energy from one input that appears on the output of the same fiber depends mostly on the coupling length.

- Nonlinear effects – A number of different effects are summarized under this term. Many of these are caused by the fact that the attenuation and the refractive index of a fiber are a function of signal power. This means that a fiber can be seen as a linear system as long as the injected power is low. To achieve high bit rates above 10 Gbit/s, however, high bit energies have to be transmitted. Therefore it becomes necessary to consider these effects.

2.3. Important parameters for Optical Transmission in Fiber

- Optical Power.
  The optical power of a transmitter is usually given in dBm, that is the power of the signal normalized by 1 mW.

  $$P(dBm) = 10 \log \frac{P_{out}}{mW}$$

  That is, for a typical laser output power of 1 mW we have 0 dBm, 50 mW equals 17 dBm.

- Attenuation.
  The loss of optical power in any component is described using the notion of attenuation. It is defined in dB.

- Dispersion.
  Different components of the light signal travel at different speed through the fiber which leads to a widening of the pulses. It leads to Inter-signal Interference (ISI) and limits the possible bandwidth and the transmission distance without regeneration. There are three elements of dispersion:

  - Chromatic dispersion
  - Modal dispersion
  - Polarization mode dispersion (PMD)

  The dispersion D is usually specified in $ps/(km \cdot nm)$.

- Crosstalk. The effect of other signals on the desired signal. Almost every component in a WDM system introduces some crosstalk, especially filters, switches, amplifiers and the fiber itself (by the way of nonlinearities) There are two forms:

  - Interchannel crosstalk.
    Imperfect filters or switches can cause a portion of the signal on neighboring wavelengths to be received by the photodetector. The adjacent channel suppression is the ratio of the output powers on the two neighboring channels. It is usually given in dB.
  - Intrachannel crosstalk.
    Intrachannel crosstalk is caused by imperfect switches or cascaded wavelength demultiplexers/multiplexers. A portion of the signal on the same wavelength, but from a different input than the desired one leaks into the desired signal. It is not possible to fight this effect using filters which makes the intrachannel crosstalk a harder problem in large networks.

2.4. Light Generation

The generation (emission) of light is mostly performed by LASERs (Light Amplification by Stimulated Emission of Radiation). Another possibility is the LED (Light Emitting Diode), but the amount of energy per wavelength a LED can emit is fairly low, so we will only consider lasers here. The most useful type of lasers in optical networks is the semiconductor laser. Here, the ground and excited level is equivalent to the valence and the conduction band, respectively. The laser itself is a p-n junction and light of a given frequency (again, determined by the cavity length) is emitted, when an electrical current is applied.
A Fabry-Perot laser consists of two mirrors and a cavity in between. One of the mirrors only partially reflects the light. The cavity is filled with a quasi-stable lasing medium. An excitation device applies electrical current to it. Photons that are emitted stimulate the emission of others. Photons for which the frequency is an integral fraction of the cavity length interfere constructively and build up light of the given frequency between the mirrors. Thus, the length of the cavity determines the frequency of the light that the laser emits through the semipermanent mirror.
The Distributed Feedback (DFB) Laser is able to emit only a single wavelength instead of all integral fractions of the cavity length. This reduction in the number of wavelengths leads to a higher resolution and a lower linewidth[1] of the lasers. This decreases the chromatic dispersion and the crosstalk in the fiber and thus enables a transmission over a longer distance.

Type of transmitter	LED	Fabry-Perot	DFB
Linewidth	35nm	10nm	20MHz
Output power	-20dBm	0-6 dBm	0-10 dBm
Price	$	$$	$$$

Table 2.1.: Important parameters for Lasers

2.5. Light Modulation

Information can only be transported on the fiber if it is encoded properly. Lasers are modulated either directly by varying the injection current or externally by passing the light through a controllable device that changes the amplitude and/or the phase of the outgoing light. Signals can be modulated either analog via amplitude modulation (AM), frequency modulation (FM) or phase modulation (PM).
Direct modulation introduces problems when large signals are to be modulated onto an optical channel. This is the case for digital signals that are of a rectangular shape. Here a so called chirp arises due to a change in the refractive index of the lasing material, which results in a phase and frequency modulation in addition to the intended amplitude

[1] The spectral width of the emitted light

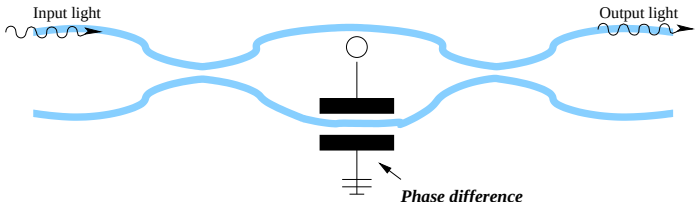

Figure 2.2.: Schematic of a Mach-Zehnder Interferometer

modulation and therefore a significant broadening of the pulse spectrum. However, direct modulation is used for multichannel sub-carrier applications, where analog subchannels (of a small amplitude) are multiplexed onto the optical channel.
Amplitude shift keying (ASK) is currently the preferred modulation technique for digital channels in the optical domain. It is also called on/off keying (OOK), because the signal level changes between 1 (light on) and 0 (light off). A Mach-Zehnder interferometer can be used as a modulation device here. The basic principle of it can be seen in 2.2. Light is led into a waveguide that is split up into two parallel tracks of equal length. When no voltage is present, the light recombines at the end of the interferometer without loss. But when a voltage is applied on one of the parallel waveguides, it produces a phase shift in the lightwave that is led through it. If this phase shift equals π, no light recombines at the end. Thus, by applying an appropriate voltage this device can act as an on/off switch. Data rates up to 40 Gbit/s have been demonstrated using external modulators of this kind.

2.6. Light Transport

Optical fibers are thin filaments of glass which act as a waveguide. Light travels in the fiber due to total reflection. This phenomenon occurs when the refractive index (the ratio between the speed of light in a certain medium to the speed of light in the vacuum) of the inner waveguide (the core) is higher than that of the cladding. This means that light in the outer regions of the fiber travels faster than in the inner regions, an effect that is used in gradient index fibers. In general there are two types of fiber, depending on the diameter of the core: multi-mode fibers, where the core diameter is around 50 to 100 μm and mono mode fibers (10 μm). The difference between both is the number of modes that travel along the core. A mode is a solution of the wave equation that is derived from Maxwell's equations. Less formally, we can say that there are many angles at which the light can be coupled into the fiber. At most of these angles the light that is reflected at the border to the cladding interferes with the incident light destructively. Only at a small number of angles that is proportional to the square of the core diameter the light interferes constructively. Although multi-mode fibers have low insertion loss due to their large core diameter, inter-modal dispersion limits the range a signal can travel. It is possible to reduce the core diameter so that only one mode (called the fundamental mode) passes through. The resulting single mode fiber has superior properties over the multi-mode fiber in that

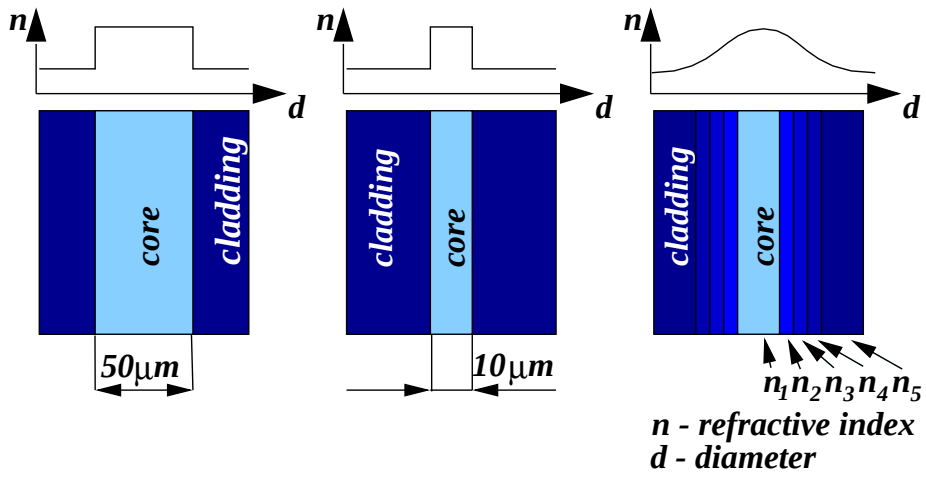

Figure 2.3.: Single, multi, and graded index mode fiber.

there is no inter-modal dispersion and the data rate and transmission range can be much higher. One disadvantage of single mode fibers is that semiconductor lasers are needed to concentrate enough power to couple into the small core. A compromise between both kinds of fiber is the graded index fiber, where the refractive index decreases from the core to the cladding in many small steps. That way, as mentioned above, the different modes travel with almost the same speed through the fiber and the inter-modal dispersion is decreased. Typical values for the attenuation in a standard SMF are reported to be 0.2 db/km. The chromatic dispersion is usually given around 16 ps/nm-km. A dispersion shifted fiber (DSF) like the "MetroCor" fiber [Inc00] has a D=-8ps/nm-km in the 1550nm wavelength band and is thus better suited for long range transmission.

2.7. Light Amplification

Currently there are two types of optical amplifiers: semiconductor optical amplifiers (SOA) and doped amplifiers (PDFA or EDFA). SOAs are based on a forward-biased p-n junction. They amplify over a wide range of wavelengths (100 nm), but suffer severe crosstalk problems.[2] Important parameters for Amplifiers are the achievable fiber-to-fiber gain and the noise figure. While an EDFA is typically able to deliver at least 25 dB gain at a noise figure F=5dB, the values for SOAs are slightly worse (G=20-25dB, F=6dB). The range of amplification is higher for the SOA than the EDFA (around 45 nm). Because of the lower crosstalk that is introduced by EDFAs, these are preferred for long range transmissions. In addition, the point of amplification can be remote for EDFA that are powered by a pump

[2] An optical signal that is amplified lets many electrons leave the conduction band and fall back to the valence band. Thus, the signal reduces the population inversion seen by other signals. The result is a negative imprint of the signal on all other signals. This is exactly what we call *crosstalk*.

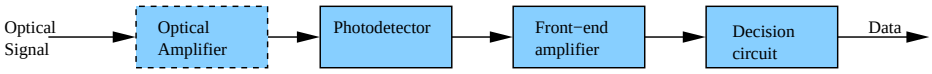

Figure 2.4.: Basic components of an optical receiver (after [RS98]).

laser on a lower wavelength (eiter 980 or 1480 nm). This way it is possible to amplify a signal inside an under-sea cable.

2.8. Light Detection

A receiver converts an optical signal into a usable electrical signal. Fig. 2.4 shows the basic components of a optical receiver. Optical receivers suffer three sources of noise: *thermal noise* which adds to the total photocurrent, *shot noise* which is a representation of the variation of the received current, and spontaneous emission from optical amplifiers. There are four basic principles of optical receivers:

2.8.1. Direct Detection

Direct detection in principle works like an inverted semiconductor laser or amplifier. Different semiconductor materials reveal different so-called cutoff frequencies, under which the material becomes transparent. For instance, silicon has a cutoff wavelength of 1.06 μm, so that it is only usable as a photo-detecting material in the 850 nm band.[3] A photodiode is a reverse biased p-n junction. Through the photoelectric effect, light incident on the p-n junction will create electron-hole pairs in both the "n" and the "p" region. The electrons created in the "p" region will cross over to the "n" region and the holes created in the "n" region will cross over to the "p" region, thereby creating an electrical current, that is referred to as *drift*.
This current is led through a threshold device, where it needs to be above or below a certain threshold for a bit time to determine a logical "1" or "0", respectively.
If the electron-hole pairs are generated far away from the p-n junction, they can only move to the other side by *diffusion*, which creates another current that only very slowly reacts to the incoming light and therefore limits the frequency response of such a device.

2.8.2. PIN Photodiode

The I stands for "intrinsic", which means that there is another semiconductor material in between the p and n regions, respectively. An example for such a PIN diode is a combination of InP for the p and n regions and InGaAs for the intrinsic region, which is usually much wider than the other regions. A schematic of this can be seen in Fig. 2.5. While InP is transparent in the 1.3 and 1.5 μm band, respectively, InGaAs is highly absorbant. This way the diffusion component of the photocurrent is totally eliminated.

[3] Light of a longer wavelength has a lower energy that may not satisfy the bandgap energy of that material. Thus, no electron-hole pairs can be produced.

Figure 2.5.: Schematic of a PIN diode

2.8.3. Avalanche Photodiode (APD)

If the reverse biasing voltage is further increased, one photon no longer generates just one electron-hole pair, but the resulting electrons themselves collide with other and thus create a so-called avalanche multiplication. The multiplicative gain of such a photodiode is an important parameter, since the variation of the resulting current increases with the gain and therefore leads to an increased noise level.

2.8.4. Coherent Detection

Another form of light detection is the coherent one, where a local oscillator is used to limit the effect of the thermal noise. Thus it allows the reception of weak signals from a noisy background. Depending on the frequency of the local oscillator we can differentiate between homodyne and heterodyne coherent detection. While homodyne detection requires the local oscillator to be of the same frequency, phase and polarization, heterodyne detection uses a local oscillator of a slightly different frequency (typically a few GHz away). The latter produces an intermediate frequency (IF) that is electronically processable. This feature gives rise to a number of interesting problems (and solutions). For instance, the IF could be filtered with much better accuracy than using optical filters. This might enable for a tighter channel spacing and for a fast (packet) switching between those channels.
Another, even more interesting point is the combination of optical and wireless transmission. Grosskopf et al.[BGRS98] proposed a heterodyne coherent detection in base stations of a wireless LAN operating at 24 or 60 GHz. In the proposed architecture, the local oscillator is not local to the base station anymore, but itself resides in a central node that coordinates a large number of base stations. That way, the problems of phase noise that usually arise with coherent receivers are avoided and additionally, the base stations of such a wireless-over-fiber LAN could be totally passive and therefore, inexpensive.
The responsivity R is a measure of the photocurrent that is produced by the receiver per received input power. It is takes values around R=1 A/W for PIN diodes and R=8 A/W for an APD.

2.9. Optical Switches

Optical switches that are used today usually are wavelength insensitive, in other words, they switch all wavelengths from one input fiber to the destined output fiber. Generally we can

divide switches into two classes: relational and logic devices. While in the first architecture the connection between input and output fiber is made as a function of a control signal, logic devices make this decision on the basis of the incoming data (e.g. packet headers). The basic difference between the types of optical switches that are introduced in the next chapters is their speed. As usual, there is a tradeoff between insertion loss, polarization sensibility, crosstalk and switching latency. We start with the most common technology, namely the mechanical switches.

2.9.1. Mechanical Switches

Tuning times of mechanical switches are usually in the order of 10 ms. This makes them improper for packet switching, but their crosstalk suppression (55dB) and low insertion loss (3dB) makes them favorite candidates when it comes to circuit switched networks.

- WaveStar/MicroStar© technology developed by Lucent: MicroStar technology is used to attain relatively large switching fabrics with sub-millisecond switching speed and a small product footprint. MicroStar relies on an array of hundreds of electrically configurable microscopic mirrors fabricated on a single substrate to direct light. The switching concept is based on freely moving mirrors being rotated around micro-machined hinges.

- BubbleJet© technology by Agilent. Works like an ink printer, but the bubbles are used as mirrors that change the way of the incoming light into a new (e.g. 90° rotated) direction. In Agilent's way of doing things, the basic building block is a 32-by-32 port switch on a chip. Inside the chip, there's a matrix of microscopic channels filled with a special liquid, through which light travels. At each intersection point, a bubble jet pen can heat up the liquid so that it boils and creates a tiny bubble. This acts like a mirror, reflecting light onto the intersecting path. These 32-by-32 port modules can be linked together to create large-scale switches.

2.9.2. Thermo-Optic Switches

Thermo-optic switches are MZIs that can be thermically influenced. Switching times are in the order of 2 milliseconds[RS98].

2.9.3. Electro-Optic Switches

Electro-optic switches are directional (3 dB) couplers whose coupling ratio is changed by changing the refractive index of the material in the coupling region. Switching times are less than 1 ns, but electro-optic switches have modest crosstalk and polarization properties.

2.9.4. SOA switches

The semiconductor optical amplifier (SOA) can be used as a on/off switch by varying the bias voltage. If it is low, no population inversion occurs and the incoming signal is absorbed.

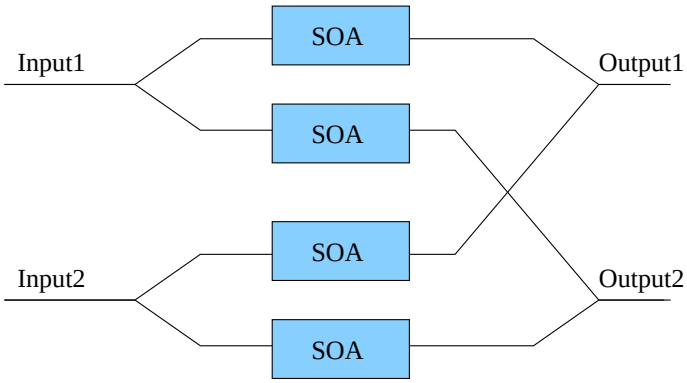

Figure 2.6.: Schematic of a 2x2 amplifier gate switch

If the bias voltage is high, the incoming signal is amplified, thereby compensating for the insertion loss of the amplifier and leading to high extinction ratios.[4]

2.9.5. Important parameters for Switches

Crosstalk Switching time

2.10. Tunable Filters

Tunable filters are used in optical receivers and in larger wavelength switch configurations to select the desired wavelength out of the pool of WDM channels. Similar to optical switches they offer a wide variety of switching times and wavelength ranges and unfortunately, both properties seem to be proportional. Until recently, there was no tunable filter available that would be fast enough to tune in between packet arrivals. Nowadays, electro-optic filters seem to become promising candidates to accomplish this task, but they come at a high price and are still limited in their tuning range to a small number of channels (e.g. 10 [Bra96]). For that reason, tunable filters (as well as transmitters) were not the way we followed here, so they are just briefly mentioned with their typical tuning ranges and times listed in table 2.10.

- Mach-Zehnder Interferometer (MZI):
  The schematic of the MZI was already shown in Fig. 2.2. It is the basic element for a number of tunable filters, only the way to accomplish the delay in the second arm is different.

- Mach-Zehnder Chain:
  The idea is to cascade several MZIs with $i\Delta L(i = 1, 2, \ldots)$. The different FSR (Free

[4]The *extinction ratio* is the power ratio, usually in dB, between the outgoing signal for a bit "0" to a bit "1".

Tunable Receiver	Approx. Tuning Range (nm)	Tuning Time
Fabry-Perot	500	1-10 ms
Acoustooptic	250	10 μs
Electro-optic	16	1-10 ns
LC Fabry-Perot	30	0.5 - 10 μs

Table 2.2.: Tunable optical filters and their associated tuning ranges and times (after [BJB+97]

Spectral Range) of the MZI stages lead to the extraction of a single wavelength. (see [RS98], pp. 111) Such a device is easy to integrate, but has a slow tuning due to thermic change of the refractive index. In addition, the loss increases with every stage.

- Fabry-Perot Filter:
  Similar to the Fabry-Perot laser, wavelengths are selected by mechanically adjusting the cavity between two mirrors. Slow tuning, but huge tuning range.

- Acoustooptic Filter:
  A piezoelectric crystal is used that changes its refractive index whenever a sound wave is applied on it. The crystal can act as a grating and extract a single wavelength that depends on the sound wave applied. The advantage of the AOTF is that any number of wavelengths can be selected simultaneously. The speed of the sound waves limits the tuning speed of the AOTF.

- Electro-optic Filter:
  Similar to the AOTF, but the refractive index of the crystal is changed by electrical currents. EOTFs are very fast but limited in their tuning range to around 16 nm.

- Liquid Crystal:
  Similar to a Fabry-Perot Filter, but the cavity is filled with a Liquid Crystal (LC). Its refractive index can be modulated by an electric current. This technology currently offers the best proportion of tuning times and ranges up to now.

2.10.1. Fixed Filters

2.10.1.1. Bragg Gratings

Bragg gratings are widely used in optic communication systems. The basic principle of operation is a periodic perturbation of the refractive index in a waveguide. In its special form of a *fiber Bragg grating* this change of the refractive index is directly written into the fiber. This is accomplished by photosensitive properties of certain types of fiber. Silica fiber doped with germanium is exposed to ultraviolet light which causes a change in the refractive index of the fiber core. This change can be made periodic by letting two UV

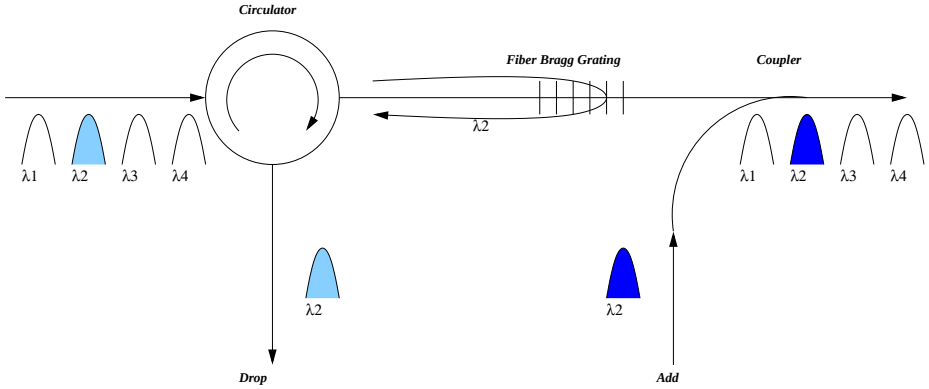

Figure 2.7.: A wavelength add-drop-multiplexer based on a fiber Bragg grating.

sources interfere. At the point of constructive interference the refractive index is increased while it is unchanged where the light beams interfere destructively. The length of the period Λ determines the so-called Bragg wavelength:

$$\lambda_0 = 2n_{eff}\Lambda$$

This wavelength is reflected in the fiber core while all other wavelengths are transmitted. Together with an optical circulator (cf. [RS98] p.88) simple optical add-drop multiplexers as shown in Fig. 2.7 can be build.

2.11. Arrayed Waveguide Gratings

The Arrayed Waveguide Grating can be found in the literature under different names: The terms Phased Array, PHASAR [Smi88], AWGM, Dragone Router and some more all refer to the same device. To our knowledge it has been parallely invented by Meint Smit [Smi88] and Corrado Dragone [Dra91]. Throughout this work we refer to it as AWG.
In principle it consists of two NxN' passive star couplers that are arranged on a single chip. (A typical size of these chips is 30x40 mm.) The N' outputs of the first star coupler are connected to the inputs of the second by a number of waveguides that is much larger than N ($N' >> N$). Neighboring waveguides show a constant difference in length ΔL. A light signal entering on one input in Fig. 2.9 is split in the first star coupler and recombined in the second. Due to the length differences of the waveguides there is a phase difference in the light that exits the waveguides. Thus, depending on where the input port is situated and which wavelength the optical signal resides on, the light recombines at exactly one output port of the second star coupler.[5] For another wavelength coming from the same input port, this point of recombination (constructive interference) will be slightly left or

[5] Actually, there is more than point of recombination, but the others are outside the scope of the output ports.

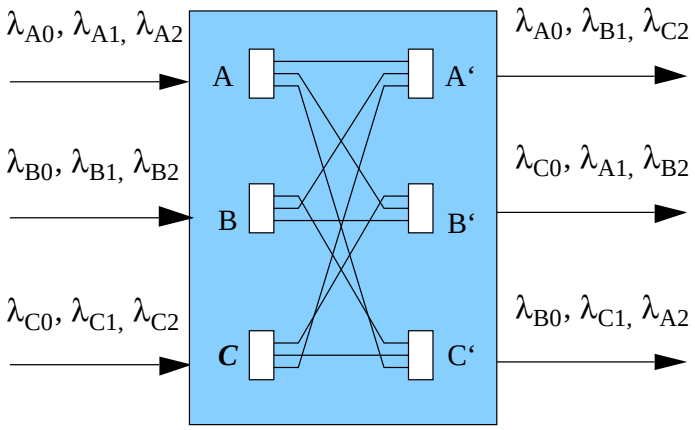

Figure 2.8.: The logical structure of a 3x3 Arrayed Waveguide Grating

right of the previous. Light from other input ports will recombine in the same way, but on different output ports for each wavelength. Interestingly, when there is a λ_1 that goes out the before-last output and a λ_2 that leaves through the last output then λ_3 will appear on the first output, it is somehow "wrapped around".[6] In principle all routing of wavelengths is done by the selection of the input port and the input wavelength. A signal on wavelength λ_1 from input A in Fig. 2.8 is routed to output B', while the same wavelength from input B is routed to output A' and from input C to output C'. One basic property of the AWGs is their periodicity. There is a so-called free spectral range that describes the difference between two wavelengths coming from the same input port and leaving through the same output port.

The order m of the AWG is defined by:

$$m = \frac{N_g \cdot \Delta L}{\lambda_c} \tag{2.1}$$

with λ_c for the center wavelength and N_g being the effective group index:

$$N_g = N_{eff} - \lambda_c \frac{d \cdot N_{eff}}{d \cdot \lambda} \tag{2.2}$$

The free spectral range then is:

$$FSR = \lambda^{(m)} - \lambda^{(m+1)} = \frac{\lambda^{(m)}}{m} \tag{2.3}$$

[6] Here we have our second point of recombination. While the first one moves out to the right, the next one moves in from the left.

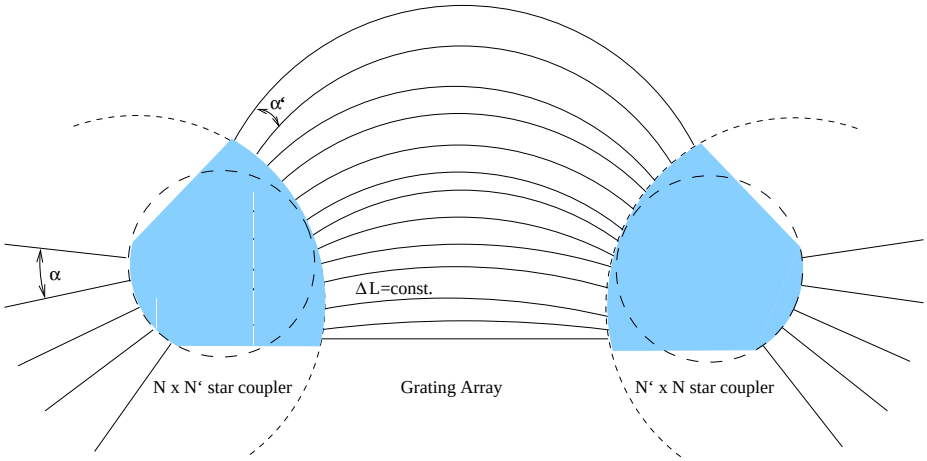

Figure 2.9.: Schematic of an Arrayed Waveguide Grating

2.11.1. Crosstalk in an AWG

The device can be logically seen as a combination of N demultiplexers and N multiplexers, even though it is essentially an analog grating based element with severe limitations in its size due to crosstalk properties. Besides other limiting factors like insertion loss there are three types of crosstalk in an AWG: interchannel, coherent intra-channel and incoherent intra-channel [PONJ99]. Interchannel crosstalk appears between light on different wavelengths leaving the same output port. (Thus, coming from different input ports.) Intra-channel crosstalk in general is the mixing of signals of the same wavelength coming from different input ports. In two special configurations of the AWG which will be explained in the next section there is also the effect of crosstalk between a signal on a certain wavelength that comes in from two input ports. In general, crosstalk in AWGs increases with the number of channels (inputs) and with decreasing channel spacing. Nevertheless there are 40x40 AWGs commercially available that provide an intrachannel crosstalk below -25dB and an interchannel crosstalk below -30dB. [VvdVTB01]

2.11.2. Configurations of AWGs

There are a number of possible configurations for such a beautiful device. First of all, it serves as wavelength de-/multiplexer. Therefore 1xN AWGs have been designed and are widely available as of today. A much wider range of applications is opened up by NxN AWGs as the one shown in Fig. 2.9. Also NxM devices have been designed. What is common to all of them is that they are symmetric in the sense that the direction of the light (is it incoming or outgoing) does not matter in the choice of the output port. We will stick to fully symmetric NxN devices in the following. By fully symmetric we mean that also the labeling of the in/outputs is symmetric, such that e.g. light on λ_x that goes from input port 2 to output port 4 also goes from output port 2 to input port 4.

Another popular application of the AWG is its use as a wavelength Add-/Drop-Multiplexer (WADM). There are two basic configurations when using it for that purpose: *Loop back* and *fold back*. Tachikawa et.al. [TIIN96] propose the use of a looped-back configuration for the processing of optical signals in a network node. Here, one input (usually in the middle) is being used for all incoming wavelengths (e.g. from a WDM ring or passive star coupler). All output ports except the one that is going back to the ring or coupler then have exactly one wavelength. This is fed into a signal processing unit and then back to the input port with the same label as the output port. Pires et.al. [PONJ99] show that the looped-back configuration suffers crosstalk much more than the so-called fold-back configuration, where signals are being fed back from the output port side which requires twice as much ports on the AWG as the looped back variant. When using the fold-back configuration, this paper showed analytically the feasibility of a 13 node WDM ring, compared to 11 nodes otherwise. This means that one can send an optical signal through a state-of-the-art AWG 13 and 11 times, resp., and still achieve a BER of 10^{-12}!

2.11.3. Notation of the wavelength routing

Here we use a similar notation to the one presented by Oguchi[Ogu96]. The output matrix O_m is a product of the Wavelength Transfer Matrix (WTM) $L_{m,n}$ and the input matrix I_n:

$$O_m = L_{m,n} * I_n \tag{2.4}$$

The product of the elements of the WTM and the input wavelengths is defined as follows:

$$\Lambda_k * \lambda_k = \lambda_k \tag{2.5}$$

$$\Lambda_l * \lambda_k = 0 \qquad (l \neq k) \tag{2.6}$$

For an AWG with m=n=5, i.e. 5 inputs and 5 outputs, the WTM is the following:

$$L_{5,5} = \begin{pmatrix} \Lambda_1 & \Lambda_2 & \Lambda_3 & \Lambda_4 & \Lambda_5 \\ \Lambda_2 & \Lambda_3 & \Lambda_4 & \Lambda_5 & \Lambda_1 \\ \Lambda_3 & \Lambda_4 & \Lambda_5 & \Lambda_1 & \Lambda_2 \\ \Lambda_4 & \Lambda_5 & \Lambda_1 & \Lambda_2 & \Lambda_3 \\ \Lambda_5 & \Lambda_1 & \Lambda_2 & \Lambda_3 & \Lambda_4 \end{pmatrix} \tag{2.7}$$

Numbering the inputs from A to E leads to the following input matrix $I_{5,5}$($A_k = \lambda_k$ on input A):

$$I_{5,5} = \begin{pmatrix} A_1 & A_2 & A_3 & A_4 & A_5 \\ B_1 & B_2 & B_3 & B_4 & B_5 \\ C_1 & C_2 & C_3 & C_4 & C_5 \\ D_1 & D_2 & D_3 & D_4 & D_5 \\ E_1 & E_2 & E_3 & E_4 & E_5 \end{pmatrix} \tag{2.8}$$

Equation 2.4 now gives the output matrix $O_{5,5}$:

$$O_{5,5} = \begin{pmatrix} A_1 & B_2 & C_3 & D_4 & E_5 \\ E_1 & A_2 & B_3 & C_4 & D_5 \\ D_1 & E_2 & A_3 & B_4 & C_5 \\ C_1 & D_2 & E_3 & A_4 & B_5 \\ B_1 & C_2 & D_3 & E_4 & A_5 \end{pmatrix} \tag{2.9}$$

We will come back to this notation in chapter 8.

2.12. Conclusions

This chapter explained some of the physical phenomena necessary to understand the promises and limitations of the term *optical networks*. Light generation, transport, filtering, switching and detection were introduced. Several basic elements of an optical infrastructure have been introduced. For the design of optical networks the price of a certain element will definitely be one major factor. But there are other criteria as well, for instance the number of components to achieve a certain network capacity. This problem will be delt with in chapter 9, where a comparison of two architectures requiring a different kind and number of components is done. Concerning the price of components, no precise figures can be given here, but instead some general "rules of thumb": We have seen that there is a tradeoff between the speed of tuning and the tuning range of filters and lasers. SOA switches are fast and modest in price, but reveal poor crosstalk behavior. The price of a laser or filter is proportional to its tuning speed.
But, compared to electrical networks, there is still one major component missing: random access memory (RAM). This would be needed to do an optical packet switching in the same way as it is done in the electrical network nodes of today. It has been reported recently that it indeed seems to be possible to slow down a light wave near the absolute zero temperature and release it afterwards [BP02], but products based on this finding will certainly not be available within the next decades. The only form of optical memory that is available today are fiber delay lines (FDL), single mode fibers of a certain well-defined length that add some delay to the transmission path. These FDLs may be cascaded to form some primitive queues, still without the possibility to actually do a "store and forward".
What results from this is that new concepts are needed when optical packets shall be transmitted and switched without a conversion into the electrical domain within each network node.

3. Optical circuit networks

3.1. Architectures of Optical Circuit Networks

In the previous chapter the main building blocks for the optical transmission have been introduced. These can be used to transmit either analog or digital information. Analog transmission is limited in the distance because the amplification of an analog signal adds noise and a *regeneration* of the signal is impossible. Therefore any transmission over a distance of more than a few kilometers will be digital.
The so-called first generation optical networks use the WDM link as shown in Fig. 3.1 to interconnect Digital crossconnects (DXC). These DXC offer *circuit switched* services. This means that the network sets up or takes down *calls* upon request of the user. Because the requested user data rate will in most cases be only a small fraction of the available data rate of the fiber, some kind of *multiplexing* has to be performed. This may be done either in a *fixed* (time division) or *statistical* way. Traditionally, the big network providers relied on a fixed time division multiplexing because it was their primary goal to transport voice traffic. In the first part of this chapter we will introduce SONET/SDH, the worldwide standard(s) for the synchronous multiplexing and transmission of circuit switched data.
In the second-generation WDM network that is being established today, the DXCs use the lightpath service offered by Optical crossconnects (OXC), that are again connected by WDM fiber links like the one in Fig. 3.1. The term optical is somehow misleading because all of today's OXCs perform opto/electric conversion, regeneration of the signal and then electro/optic conversion, possibly on another wavelength. There is, however, the aim of a truly optical, sometimes called Photonic Crossconnect (PXC). This would allow for the establishment of an *optical circuit* or lightpath between the source and destination DXC using a wavelength *tunnel*. In result, the topology visible for the SONET/SDH layer will be different from the topology in the optical network. The question is how to do the mapping between the connections of the DXCs and the lightpaths. Right now, the lightpaths have to be established manually and are not distinguishable for the DXC. It is desirable to have the opportunity to dynamically set up (and tear down) lightpaths according to certain load or failure scenarios in the network.
In the second part of this chapter we will give an introduction into the problems that have to be solved in the process of RWA.

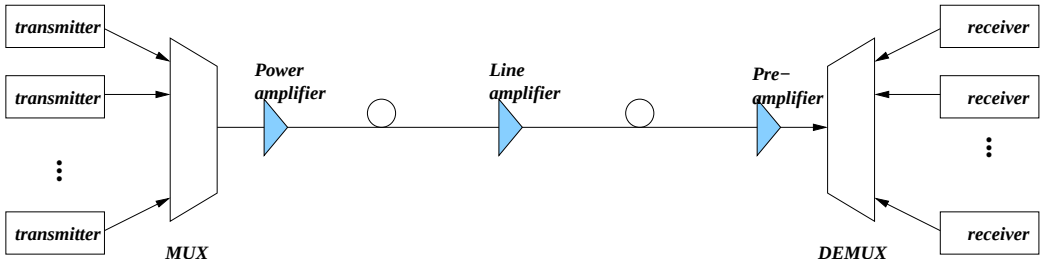

Figure 3.1.: Components of a WDM link (after [RS98]).

SONET	SDH	Data rate (gross)	Data rate (SPE)	Data rate (user)
OC-1	-	51.84	50.112	49.536
OC-3	STM-1	155.52	150.336	148.608
OC-9	STM-3	466.56	451.008	445.824
OC-12	STM-4	622.08	601.344	594.824
OC-18	STM-6	933.12	902.016	891.648
OC-24	STM-8	1244.16	1202.688	1188.864
OC-36	STM-12	1866.24	1804.032	1783.296
OC-48	STM-16	2488.32	2405.376	2377.728
OC-192	STM-64	9953.28	9621.504	9510.912
OC-768	STM-256	39813.12	38486.016	38043.648

Table 3.1.: Supported data rates in SONET and SDH. SPE=Synchronous Payload Envelope.

3.2. The Synchronous Optical Hierarchy

3.2.1. Historical evolution of SONET/SDH

The Synchronous Optical Network (SONET) was first standardized in the ANSI TX1 group in 1985 as an outcome of a work which had mainly been done at Bellcore. Soon the CCITT (later ITU) worked out an international telecommunications standard which based on SONET and was named Synchronous Digital Hierarchy (SDH). The differences between SONET and the SDH besides the naming of the transport modules and some management bytes mainly lay in the data rates supported. SDH is based on a three-fold SONET container, because of that, the basic data rate supported by SDH is 155.52 Mbit/s compared to 51.84 Mbit/s in SONET (see table 3.2.1).

The basic reason for the introduction of a fully synchronous multiplexing scheme like SONET/SDH was the synchronization overhead which was necessary in the Plesiochronous Digital Hierarchy (PDH). The proportion of this overhead grows with the overall data rate. Another reason was the need to demultiplex the whole data stream down to T1 or E1 lines at every multiplexer, since it was not possible to exactly locate a certain voice call (byte)

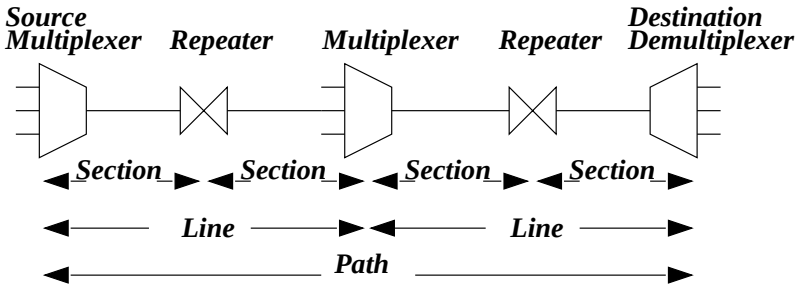

Figure 3.2.: Layer concept of SONET/SDH.

in a PDH stream.

3.2.2. The layer concept of SONET/SDH

SONET/SDH is based on a three-layer concept; it somehow resembles the OSI layering. The lowest layer is the SECTION layer, which controls the transmission of bits between two optical endpoints. The LINE layer controls the transmission between a pair of multiplexers. The highest layer is called the PATH layer. It implements an end-to-end control, at least in the sense of the SONET/SDH transmission. The layer concept is shown in Fig. 3.2.2.

3.2.3. The SONET/SDH frame format

Due to the background of SONET/SDH, which was developed as a common backbone for the old telephone network, samples are transmitted at 8000 Hz. This means that one byte in a SONET/SDH container forms a 64 kbit/s line. Multiplexing of SONET/SDH streams is performed byte-wise. (Some older implementations multiplexed bitwise.). This means that for instance a STM-4 stream is made up of four STM-1 frames (see Fig. 3.2.3) that are transmitted within 125 μs. The Path Overhead is transmitted as part of the payload, since it carries only information that is relevant for the endpoints of the SONET/SDH connection. Section and Line Overheads are recalculated and rewritten at every regenerator or multiplexer, respectively. The functions of the overhead bytes can be seen in table 3.2.3.

3.2.4. SONET/SDH Network Topologies

Although SONET/SDH basically defines a point-to-point connection, the topology of the network is arbitrary. Recent SONET networks in Northern America mostly employ a Bidirectional Line Switched Ring (BLSR) architecture, whereas in Europe a meshed net is preferred. The reason for this are the much shorter distances that have to be crossed between the main cities in Europe. In result, large SDH-crossconnects are being used in Europe compared to relatively simple SONET-Add-Drop Multiplexers (ADM) in Northern America.
APS (automatic Protection Switching) is responsible for a reconfiguration of the ring within

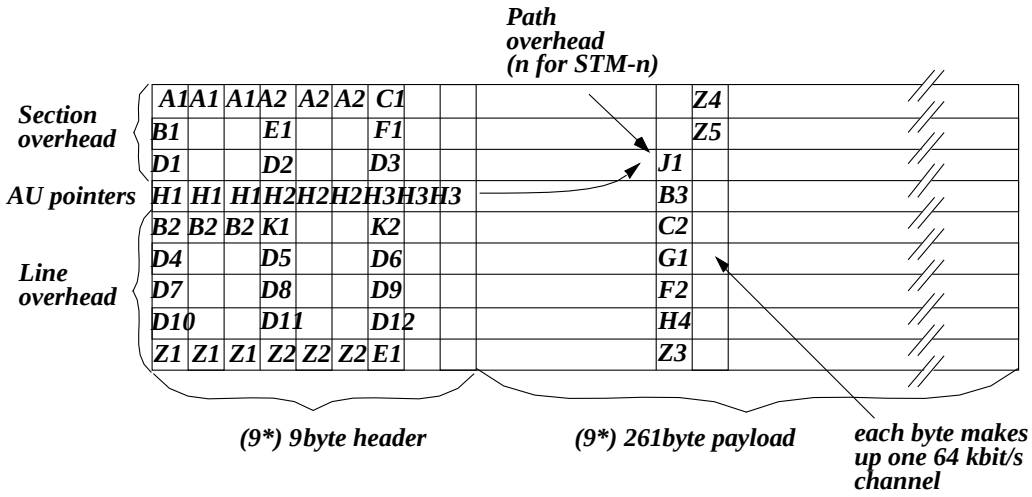

Figure 3.3.: Structure of an STM-1 frame.

Name	Function
A1,A2	Framing
AU Pointers	Administrative Unit Pointers
B1, B3	BIP-8 (Bit Interleaved Parity)
B2	BIP-24
C1, C2	STM Identifiers
D1 to D12	Data Communication Channels
E1, E2	Order Wire
F1, F2	User-defined channels
G1	Path Status
H4	Multiframe Indicator
J1	Path Trace
K1, K2	Automatic Protection Switching
Z1 to Z5	Growth (reserved as spare)

Table 3.2.: STM-1 header information

50 ms. This is done by a wrap in the two stations neighboring the failed one. One or both nodes will receive a LOS (Loss Of Signal) alarm within 100 μs. The huge amount of management information in the SONET/SDH header helps to spread the information about a failed link or node in the network such that the reconfiguration can be finished quickly. However, since SONET/SDH is circuit switched, there has to be 50% capacity reserved for APS, which can be used for unprotected traffic in error-free operation. This unprotected service is then preempted in the case of a link or node failure.

3.3. Wavelength routed networks

There is a good survey of the problems that have to be addressed in this contents in [Jue01].

3.3.1. The Routing and Wavelength Assignment (RWA) Problem

The problem of finding a route for a lightpath and assigning a wavelength to the lightpath is called the RWA problem. There are two objectives that have to be met: First that there are no two lightpaths sharing the same wavelength over a fiber link and second to minimize the network resources (nodes, links, wavelengths) used. Depending on the allowance of wavelength conversion in intermediate nodes there may be an additional constraint (the wavelength continuity constraint) under which the problem has to be solved [ZJM00]. There has been significant discussion in the literature [SAS96] to which extent wavelength conversion in intermediate nodes is useful. It has been stated that the gain of wavelength conversion in terms of reduced *blocking probability* and increased *utilization* depends on the routing and wavelength assignment algorithm and the load in the network[KA98]. *Sparse wavelength conversion*[SAS96] seems to be the tradeoff between the high cost of a wavelength converter and the gain that can be expected. It is an open question where to put the converters in the network. The assumption of full wavelength conversion in each node often serves as a lower bound in comparing different WA algorithms.
The RWA problem can be decoupled into its two sub-problems, namely the routing and the wavelength assignment, both of them are known to be NP-complete.

3.3.1.1. RWA for static wavelength assignment

In the static RWA, lightpath requests are known in advance. The objective is to route all lightpaths such that the number of wavelengths is minimized or, in a "dual" approach, route as many lightpaths as possible with a given amount of wavelengths. The problem can be formulated as an integer linear program (ILP). For large networks, heuristic methods have to be used that restrict the search space of the problem, such as in [BM96], where the set of links is restricted through which a certain lightpath may be established.

3.3.1.2. RWA for dynamic wavelength assignment

When lightpath requests are not known in advance but arrive dynamically, connections have to be set up at runtime. The objective is to chose a route and a wavelength that maximizes

the probability of setting up a certain connection while at the same time attempting to minimize blocking of future connections. The subproblem of routing can be categorized into being either fixed or adaptive and as utilizing either global or local state information.

Fixed Routing A fixed route is being given for every source/destination pair. If no (common) wavelength is available along this route, the connection is *blocked*.

Adaptive Routing based on global information Adaptive routing may be either centralized or distributed. If there is a central authority that keeps the knowledge about the global state of every link in the network, then lightpath requests can be routed by this entity. This approach is simple, but does not scale well with network size and is a potential single point of failure.
Distributed routing algorithms are alternate path routing, where a choice has to be made out of a given set of paths between source and destination (a variant of the fixed routing) and unconstrained routing. Here, all possible paths between source and destination are considered. Link state routing and distance vector routing, both also used in routing protocols of the IP world (OSPF and BGP[Ste94]) belong to this family. The problem here is to gather the global knowledge about the network state, which results in a significant control overhead. To make things worse, not only the "standard" link state information (link up/down, total number of wavelengths etc.) have to be broadcast to all nodes in the network, but also additional, "optical" information like polarization mode dispersion (PMD) and amplifier spontaneous emission (ASE), which limit the total length of a lightpath without O/E/O conversion are critical for a routing decision and have to be transmitted as well[SCT01].

Adaptive Routing based on local information To reduce the amount of state information that has to be broadcast and thereby improve the scalability of the network, it was proposed to use only the information about the first k hops along the desired lightpath[LS99]. A routing decision that is based on the local state of the network does of course not guarantee the availability of a wavelength along the whole path. Still, it may be a good estimate of the congestion.
Another approach to routing with only local information is *deflection routing*. Here, a node chooses the outgoing link on a hop-by-hop basis rather than on an end-to-end basis. For a lightpath request, the shortest path to the destination is considered first, if this is blocked, the request is deflected to another node that itself then tries to setup the shortest path to the destination and so on. As it can be guessed from this description, the problem of routing loops arises here. This may be solved using time-to-live (TTL) stamps in the requests.
For all routing algorithms, the lightpath may be routed either the shortest path or the least congested path (LCP). The latter generally distributes the traffic better over the network and performs better under high load.

3.3.1.3. Wavelength Assignment

The second subproblem is connected to graph-coloring problems, that are known to be NP-complete. Thus, a number of heuristics have been proposed[ZJM00]. A first approach is to select one of the available wavelengths along a lightpath at random, usually with a uniform probability. Another approach is called *First Fit*. Here, all wavelengths are numbered and the one with the lowest index is chosen first. This way, all lightpaths are "packed" onto the available set of wavelengths, leaving a higher probability for longer paths in the upper regions to be free. Other approaches are Least-Used, Most-Used and Min-Product. A review of these approaches can be found in [ZJM00].
The next question in the context of RWA is that of making the reservations for wavelengths in the lightpath. It will be covered in a later section (see section 5.6).

4. The Internet - Protocols and Traffic

4.1. Internet protocols

The Internet of today is a distributed and heterogeneous "network of networks". In spite of this heterogeneity all nodes that communicate over the Internet use a common language (with many dialects) called the Internet protocols. Being a whole protocol suite rather than a single layer protocol, these protocols cover the OSI layers above, say, layer 2.5 (see Fig. 4.1). Because the higher layer protocols strongly influence the traffic that is seen by the lower layers, this chapter briefly explains the characteristics of TCP, UDP, and IP.

4.1.1. IP – The network layer protocol

The IP protocol, first documented in RFC 791[Pos81b], is the primary protocol on the network layer of the Internet protocol suite. It provides a connection–less, best–effort delivery of datagrams through an internetwork. As of today, the version 4 (abbreviated IPv4) is used almost exclusively. It uses 32–bit long addresses for the nodes in the network. A node that is responsible for routing and forwarding of the datagrams is called an IP router. Whenever an IP datagram arrives at a router, its destination address is matched to all entries in the internal routing table to find the link on which the packet is to be forwarded. The routing tables, which consist of destination address/next hop pairs, are generated and updated by routing protocols. Depending on the function and location of the routers these protocols differ in the amount and frequency of the information that is being exchanged between the routers. IP routing protocols are dynamic and update the routing tables at regular intervals according to the information gathered from the neighbors. Because only the next hop of a datagram is known in a router and because of the dynamic change of the routing tables it is possible that a datagram loops infinitely in the network. To prevent this, the number in the TTL (time to live)–field of the IP header (see Fig. 4.2) is decremented in each hop until is reaches zero and the datagram is discarded. Datagrams may be delivered out of sequence or be dropped from overflowing queues in the routers. It remains the task of the higher layer protocols to correct this.

Since the mid-nineties, a new version of the IP protocol, IPv6, has been considered in the standardization process of the IETF[DH98]. Beside some simplifications in the header format, IPv6 differs in two main aspects: First, the address length grows to 128 bit, allowing a much greater number of addressable nodes and a simpler auto-configuration of addresses. Second, the capability of labeling packets is introduced, with a flow label of 20 bit length in the header. This label is – not by coincidence – to be found again in the MPLS header (see Fig.5.5), as we will see in the next chapter.

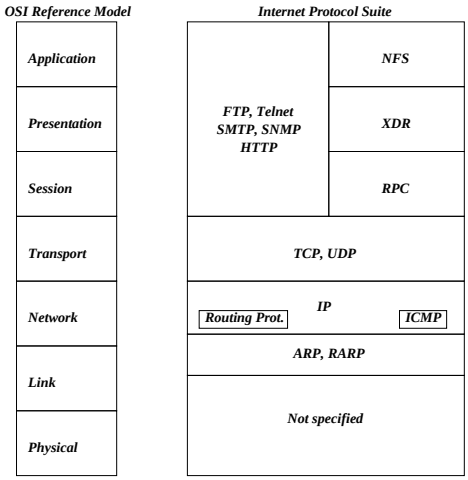

Figure 4.1.: The Internet protocol suite.

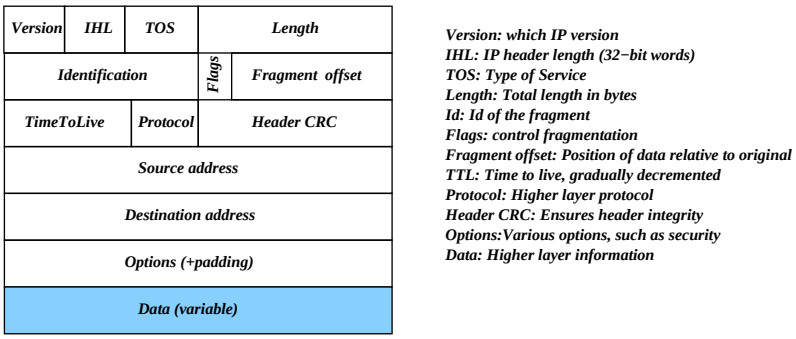

Figure 4.2.: The IPv4 packet format.

4.1.2. TCP - Transmission Control Protocol

TCP is by far the dominating transport layer protocol in the Internet[Pos81d]. It ensures a connection–oriented and reliable transmission over IP. It provides combined mechanisms for flow and error control based on a *sliding window*. But what made TCP so successful is the *congestion control* that consists of four intertwined mechanisms. Together these mechanisms prevent the network from breakdown under an increasing traffic load. The basic idea behind TCP congestion control is that the loss of a segment (a datagram that is part of a larger stream) indicates congestion somewhere in the network. After this congestion has been experienced by a connection, the data rate of this connection is first drastically reduced and then only slowly increased, again. Details to the mechanisms can be found in RFC 2581 [APS99]. The following definitions are taken copied here because they are necessary for the understanding of the following:

```
SEGMENT:
   A segment is ANY TCP/IP data or acknowledgment packet (or both).

SENDER MAXIMUM SEGMENT SIZE (SMSS):  The SMSS is the size of the
   largest segment that the sender can transmit.  This value can be
   based on the maximum transmission unit of the network, the path
   MTU discovery [MD90] algorithm, RMSS (see next item), or other
   factors.  The size does not include the TCP/IP headers and
   options.

RECEIVER MAXIMUM SEGMENT SIZE (RMSS):  The RMSS is the size of the
   largest segment the receiver is willing to accept.  This is the
   value specified in the MSS option sent by the receiver during
   connection startup.  Or, if the MSS option is not used, 536 bytes
   [Bra89].  The size does not include the TCP/IP headers and
   options.

RECEIVER WINDOW (rwnd) The most recently advertised receiver window.

CONGESTION WINDOW (cwnd):  A TCP state variable that limits the
   amount of data a TCP can send.  At any given time, a TCP MUST NOT
   send data with a sequence number higher than the sum of the
   highest acknowledged sequence number and the minimum of cwnd and
   rwnd.

FLIGHT SIZE:  The amount of data that has been sent but not yet
   acknowledged.
```

The reference given for the path MTU (maximum transmit unit) discovering algorithm is RFC 1191[MD90]. This algorithm determines the maximum size of a segment that can be

transported by the underlying IP network. This is done during connection setup to avoid the segmentation and reassembly of datagrams at the IP level.
All TCP implementations use a *slow start algorithm*. This means that the congestion window (`cwnd`) is gradually opened. At first a TCP sender transmits only 1 (or 2, depending on the implementation) segments and waits for the ACK packets to be received from the receiver. For every ACK that is received, the congestion window at the sender side is increased by at most `SMSS` bytes. This leads to an exponential increase of the `cwnd` in the first phase of the connection. After crossing a slow start threshold (`ssthresh`) the connection enters a second phase, *congestion avoidance*. Here, only a linear increase of the `cwnd` per round trip time (RTT) is allowed. Whenever a segment is lost, TCP assumes a buffer (queue) overflow due to congestion in the network and reduces its `cwnd` to 1 segment, i.e. `SMSS`. Loss of a segment is detected by the RTO (retransmission timeout).
Most of today's TCP implementations use in addition the *fast retransmit* mechanism [Ste97] to try to recover quickly from occasional packet loss. The fast retransmit algorithm is triggered by a series of duplicate ACKs. If the TCP sender sees three duplicate ACKs, it assumes that the data immediately after the byte being acked has been lost, and retransmits that data. The `cwnd` is only reduced to half of its original size to allow for a *fast recovery*. This way it tries to repair the loss of a packet before it causes the current pipeline of data to drain.

4.1.3. User Datagram Protocol

UDP is a connection-less transport-layer protocol that is basically an interface between IP and upper-layer processes[Pos80b]. The header is only 8 octets long and contains the source/destination port number, a length field and a checksum. UDP does not provide any of the error- or congestion control functions of TCP which makes it suitable for applications that either provide these functions on their own (like the Network File System(NFS)) or do not need a sophisticated error control (like audio or video streams).

4.2. Size of optical packets in the Internet and its influence on TCP performance

When considering optical burst switching (OBS) and optical packet switching (OPS) techniques, one of the fundamental questions is that of the size of an optical aggregate. It has a two-fold influence on the performance of such networks. First, the maximum throughput over an optical (WDM) channel depends on the ratio between the overhead (switching time, table lookup time etc.) and the payload (optical packet size). It is therefore desirable to have very short switching times and rather long packets. Of course, the switching time of optical switches is determined by fiber physics and is therefore not arbitrary. On the other hand, there is a lot of freedom in the choice of the optical packet size. Second, the throughput of a TCP connection over this link has an upper bound that depends on the error rate, the MSS (Maximum Segment Size) and the RTT (Round Trip Time).

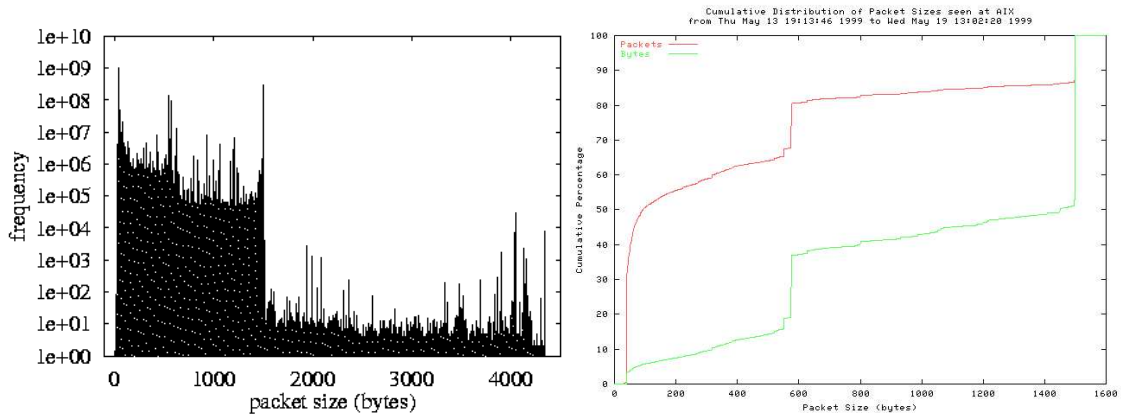

Figure 4.3.: Logarithmic-scale packet size histogram from 1997 (left) [TMW97] and packet size distribution in 1999 (right) [CAI].

4.2.1. What is the current packet size in the Internet?

There are a number of organizations that deal with measurements of internet traffic. One of the most important is CAIDA. The Cooperative Association for Internet Data Analysis is located at the San Diego Supercomputer Center on the campus of the University of California, San Diego. They have been performing measurements of internet traffic for about 10 years now. Most of these observations were done in the NSFNET and especially the vBNS in North America, which started as IP over ATM network in 1995 and is converging into a PoS (Packet over SONET) network right now.

Measurements in 1992 showed no packets above 1500 byte, and a mean packet size of 168 byte.[CBP93] 1998's mean was 347,[CMT98] 2000: 413 byte.[CAI] This means an almost linear increase of the mean packet size by roughly 30 bytes per year. There is, on the other hand, a clear 24-hour pattern and a directional asymmetry in the packet size.[TMW97] The statistical importance of this statements on the average packet size is questioned by evidence of strong modality in packet sizes. There is a predominance of small packets, with peaks at the common sizes of 40, 552, 576, and 1500. The small packets, 40-44 bytes in length, include TCP acknowledgment segments, TCP control segments such as SYN, FIN, and RST packets, and Telnet packets carrying single characters. Many TCP implementations that do not implement Path MTU Discovery use either 512 or 536 bytes as the default Maximum Segment Size (MSS) for nonlocal IP destinations, yielding a 552-byte or 576-byte packet size [Ste94]. While the dominating MTU (Message Transfer Unit) size of 1500 byte reflects the dominating LAN technology, IEEE 802.3, the MTU sizes up to 4200 byte in the left part of Fig. 4.3 stem from FDDI traffic. The portion of the latter traffic almost disappeared in the next two years, so that again 99% of the traffic has MTU sizes up to 1500 byte. Around 90% of the total traffic is controlled by TCP, which itself is dominated by WWW traffic.

4.2.2. WAN TCP performance issues

The performance of TCP over wide area networks (the Internet) has been extensively studied and modeled. Matt Mathis et al. [MSMO97] explain how TCP throughput has an upper bound based on the following parameters:

$$Throughput <= \frac{\sim 0.9 * MSS}{rtt * \sqrt{packet\_loss\_rate}} \tag{4.1}$$

with MSS = Maximum Segment Size, which is MTU minus TCP/IP headers
rtt = round trip time

That means: All other things being equal, you can double your throughput by doubling the packet size! An example is given in the paper: Consider the distance between New York and Los Angeles. The Round Trip Time (rtt) is about 40 msec, the packet loss rate is 0.1% (0.001). With an MTU of 1500 bytes (MSS of 1460), TCP throughput will have an upper bound of about 8.3 Mbps. This is only due to TCP's congestion control mechanism, no other bandwidth limiting factors are included here. With, e.g. 9000 byte frames, TCP throughput could reach about 51 Mbps.

5. IP transmission over connection-oriented optical networks

This chapter intends to give a survey over a number of possible solutions for IP transport over lightpaths. The protocols introduced here are not necessarily WDM oriented, but deal with a fast packet transport in a general way.
To transport IP packets over SONET/SDH, some kind of link layer protocol has to be employed. The most widely used protocol for that purpose is PPP. So we show the functionality and frame formats of PPP and two possible alternatives. The following sections deal with the use of ATM in this area, Multi-Protocol Label Switching (MPLS) and MPλS towards ever shorter lifetimes of a connection in the optical network. The chapter ends with an outlook on optical burst switching, an approach that can be seen as a predecessor of a true optical packet switching.

5.1. IP over SONET/SDH

Today's most popular method to transmit IP datagrams over SONET/SDH uses the Point-to-Point-Protocol (PPP)[Sim94]. This protocol has been developed to provide access to the next Internet Service Provider (ISP) over long and error-prone links (like a modem connection over the analog telephone line). Nevertheless PPP is flexible and unlimited in the data rate supported. It requires a full-duplex channel (which SONET/SDH provides). PPP consists of two protocols, the Link Control Protocol (LCP) and a specific Network Control Protocol, which is adapted to the layer three protocol, IP in this case. IP datagrams are being packed by PPP into HDLC-like frames. The HDLC-like frame is shown in Figure 5.1.

A special pattern - 01111110 (7e hex) - indicates the start and end of the frame. The use of such a flag requires a byte stuffing mechanism to exclude this pattern within the

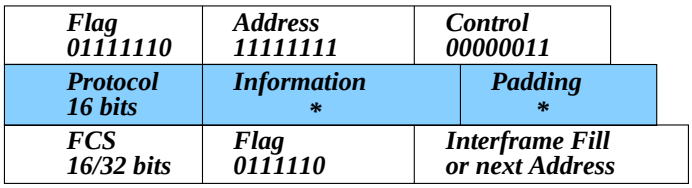

Figure 5.1.: HDLC-like framing, the PPP packet is shaded.

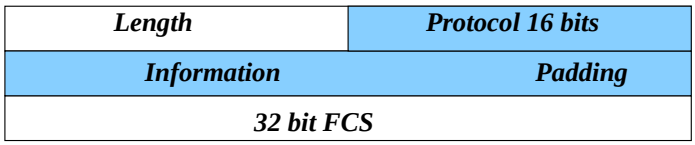

Figure 5.2.: Ethernet-like framing, the PPP packet is shaded.

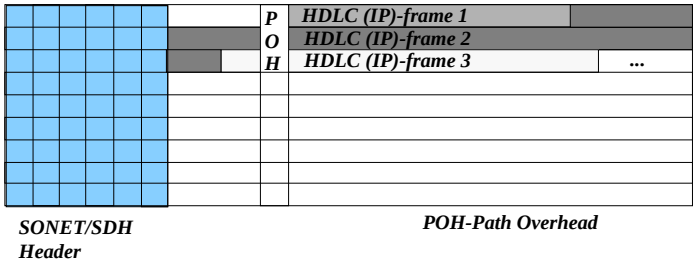

Figure 5.3.: PPP packets are written row-by-row into a SONET/STM frame.

payload. The address-field of PPP shows 0xff, which is the HDLC broadcast indicator. Because the control field of the HDLC header is always 0x03, [Sim99] defines an optional header compression where address and control are simply omitted. Because the first octet of the protocol field must not be 0xff, this is easily detectable by the receiver.
The byte stuffing mechanism introduces a speed limitation which shall be overcome by the introduction of an Ethernet-like framing starting with STM-4 (see figure 5.1, [MADD98] and [CLHVM00]). By the use of a length field special frame delimiter patterns and the byte stuffing becomes unnecessary. The Path Signal Label of the SONET/SDH-frame (C2-byte) describes which frame format is being used.

PPP uses SONET/SDH only as a byte-oriented transmission medium. HDLC frames are written into a SPE (Synchronous Payload Envelope) starting immediately after the POH (Path Overhead) (see figure 5.1). More than one frame may be written into one SPE, they are transmitted row by row. Due to the variable packet size of IP it is possible that one IP datagram continues over several SPEs.

An unexpected problem was the SONET/SDH scrambler. This device is responsible for the generation of enough 0/1 bit transitions in the data stream. It therefore uses a polynomial ($x^1+x^6+x^7$). It was shown experimentally in [MADD98], that a malicious user could easily introduce long all-zero patterns by sending appropriately manipulated (inverse pre-scrambled) IP datagrams. This continuous stream of zeroes then results in a LOS (Loss Of Signal) error which causes SONET/SDH alarms and a possible APS (Automatic Protection Switching) reaction. To avoid the possibility of the network being harmed by a user, a prophylactic octet stuffing is being proposed, where a long stream of “dangerous” bit patterns is interrupted by a 2-byte sequence. But the problem is of a general nature,

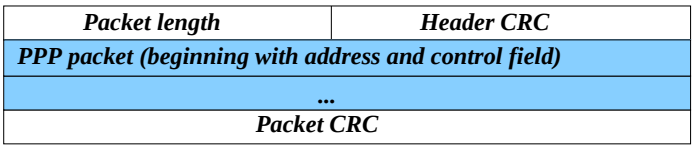

Figure 5.4.: SDL framing, the PPP packet is shaded.

in that the portions of the data stream assigned to a single user are becoming larger when going from statistically multiplexed ATM cells to large IP datagrams.

5.2. Simple Data Link Protocol (SDL)

A new proposal is the SDL protocol, which is designed to work on SONET/SDH connections as well as on other physical layers (e.g. dark fiber). The only header information is the packet length (see figure 5.2). This field is protected by a separate 16-bit CRC, which allows to correct 1-bit errors. That way, the information about the start and end of the packet will be lost only with a very small probability (10^{-8} at BER=10^{-4}). The packet length is variable and the packet is optionally secured by another CRC. For certain applications like audio/video transmissions SDL may allow the delivery of erroneous packets.
When the physical layer does not provide a byte synchronization, the start of the packet is determined by a continuous CRC calculation. SONET/SDH shall use the H4 pointer to point to the start of a SDL frame. When there is no payload to be transmitted, empty packets of a constant length are generated to keep the synchronization.

5.3. Multiple Access Protocol Over SONET/SDH (MAPOS)

MAPOS is a link layer protocol designed for the use in a LAN and provides a multiple access functionality on top of SONET/SDH connections. A so-called *frame switch* connects a number of SONET/SDH nodes in a star topology. IP datagrams are being transmitted in HDLC-frames. Every node has its own 8-bit HDLC address[1] within the LAN, which is assigned to it by the frame switch through a node switch protocol (NSP) [MM97]. Several switches may be cascaded, in this case the address of a node consists of ¡switch_address¿¡node_address¿. The matching between HDLC and IP addresses is done through an ARP (address resolution protocol) similar to an Ethernet. The whole protocol reminds of a switched Ethernet, apart from the dynamic address assignment, the framing and of course the underlying SONET/SDH. A connection to existing SONET/SDH WANs should be much easier using MAPOS than any other protocol.

[1]The remaining bits of the 32-bit HDLC-address field shall be 0. So there is room for a future development.

5.4. IP over ATM

It is not the intent of this chapter to explain ATM in detail, please refer to [WV96] for this. Instead, I will list the main features that are necessary for the transport of IP packets here. ATM is a connection oriented network. Prior to each transmission a connection setup has to be done where a switch assigns a pair of VPI/VCI numbers to the connection. During the connection cells are then forwarded (switched) rather than routed.
If the ATM switches are used for IP traffic (which is mostly the case) it is better to make use of the forwarding, because this can be done with less effort. Although IP routers are on the market now which can perform a full IP-longest prefix match at a link speed of STM-64, the proposers of IP switching argue that there is still an advantage in the complexity and consequently, there should be one in the price, too. [NML98]

5.4.1. Classical IP over ATM (CIP, CLIP)

Initially, there were two basic architectures to perform IP over ATM. The first, an IETF approach, is called Classical IP (CIP or CLIP)[LH98]. It is classical in the sense that all nodes attached to an ATM network view the attachment as a shared medium access forming a logical IP subnet (LIS). A LIS is characterized by a number of properties including:

- single IP subnet address for all hosts and routers
- same LLC/SNAP encapsulation (AAL5)
- same Maximum Transmission Unit (MTU)
- same routing architecture as in shared medium
- address resolution mechanism between IP and ATM address in the LIS

Several LIS can be overlaid on one ATM network. Every node in a LIS connects to every other by a VC (either switched or permanent). In the case of the SVC the ATM address of a LIS member is resolved by an ATM ARP request. In the PVC case a fully meshed interconnection of all nodes is needed. If an IP packet is to leave the LIS for another, it has to be expedited by a router which is a member in both LISs. This happens even if the LISs are both on the same physical ATM network. A solution to this problem is the Next Hop Resolution Protocol (NHRP). It is an extension of the ATM ARP in that the address resolution is done across LISs. A NHRP server answers the request either with the ATM address of the destination node (if that is connected to the same ATM network) or with the ATM address of the egress router nearest to it.

5.4.2. LAN Emulation (LANE), Multi Protocol Over ATM (MPOA)

The ATM Forum approach was initially called LAN Emulation (LANE). The main aim of LANE was to make ATM attractive for campus and enterprise solutions, where there already is installed a lot of equipment and the ATM network is just there to replace the old

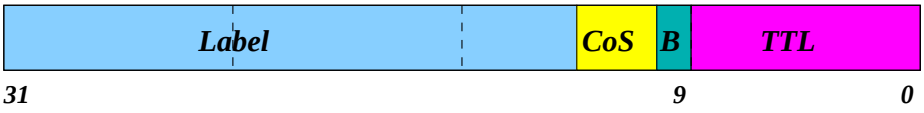

Figure 5.5.: The 4 octet MPLS shim header. The label itself is 20 bit long.

shared medium (e.g. Ethernet). Therefore the host interface card appears like a traditional interface card.

LANE is based on three servers: the LANE configuration server (LECS), the LANE server (LES) and the Broadcast-and-Unknown server (BUS). The LECS provides all clients with the ATM address of the LES, which is similar to the ATM ARP server. If the LES cannot resolve a certain LE_ARP request, a client sends the frame to the BUS, which has a direct connection to all clients.

LANE version 2 added LLC multiplexing for VC sharing, ABR (available bit rate), other QoS support and MPOA (Multi protocol over ATM). The latter is a combination of LANE and NHRP to overcome the same problem which forced the addition of NHRP to CIP.

Several problems remain, no matter if one uses CIP, LANE or MPOA:

- in maximum $n * (n - 1)$ VCs needed to interconnect a LIS, this leads to problems in the VC numbering and in the routing protocol complexity, when there is a large membership in the LIS.
- routing between LIS needed, the switching infrastructure is not fully exploited
- still mostly best-effort connections, the actual argument in favor of ATM was QoS, neither of the proposed architectures delivers more than ABR.
- Cell Size: ATM uses a fixed cell size of 53 byte, with only 48 carrying the payload. While a small cell size is useful for the transmission of data over error-prone channels, it becomes increasingly unnessecary when the BER is under 10^{-12}, as it is the case in modern SONET/SDH based fiber networks.

5.5. Multi Protocol Label Switching (MPLS)

A number of so called *IP Switching* architectures appeared in the late 90-ies that tried to address some of the problems above. Since they were all somehow similar and incompatible at the same time, the IETF decided to set up a new workgroup to standardize Multi Protocol Label Switching (MPLS) [AMA+99]. Work in this group is concentrated on a base technology that combines layer-3-routing with the label-swapping paradigm. The latter means that when a packet enters the MPLS network, not only a conventional next-hop decision is made, but the packet is associated with a Forward Equivalence Class (FEC). This FEC includes all packets that share certain properties, like

- IP Prefix - All packets going to a single IP destination are associated to one class.
- Egress Router - All packets that leave the MPLS network through a common egress router share the same FEC.
- Application Flow - All packets of a certain IP flow make up one FEC. This technique is the least scalable, since it requires the maintenance of states for every flow (detect an active flow and watch for flows that are timed out, delete dead flows).

All packets belonging to a FEC are somehow *labeled* this can be done by either encoding the label into the MAC or the network packet header or encapsulating the packet with a specific header (see figure 5.5). In the ATM layer labels are translated into VPI/VCI numbers and connections are set up. The packet which was assigned a certain label through one of the above mentioned strategies is now being switched (on the data link, e.g. ATM layer) through the entire network until it reaches the egress router. Brilliant idea, however, several open questions remain.

First of all there is the distribution of the labels using a Label Distribution Protocol (LDP). It is either possible to go for a control driven or for a data driven label exchange strategy. The control driven strategy results in the label exchange closely following other control protocol messages (like RSVP) or being piggybacked onto them. There is usually no additional delay for a setup of a Label Switched Path (LSP), when a new flow arrives. On the other hand, the setup of the LSPs is some kind of a worst case, since all possible paths have to be set up. In addition to this, the LSP can only be set up within one routing domain, otherwise a stacking of labels is needed, i.e. an encapsulation of the packet with an additional label for each control domain.

Data driven approaches like IFMP (Ipsilon Flow Management Protocol) [NEH+96] work the following way: The first packet of a flow is routed normally by IP longest prefix match and a conventional IP routing protocol like OSPF or BGP. If the number of packets from one flow exceeds a certain threshold, the IP switch decides to set up a LSP to the egress router. All remaining packets of this flow are then switched through the network. If the LSP is idle for a certain time, it is automatically deleted. Advantages of the data driven approach are that the LSP can cross routing domains and essentially be end-to-end and second, that the number of labels is determined by the number of flows and not by $(n^2 - n)$. The additional setup time for each LSP is the drawback of such an approach.

5.5.1. MPLS Signaling Protocols

The IETF working group decided to allow for two signaling protocols, CR-LDP and RSVP-TE. Both protocols originally had different purposes, but could be reused for label distribution and QoS reservation. Either of the protocols had to be extended therefore. The basic task for a signaling protocol in MPLS is to reserve the specified resources and to set up the forwarding tables (do the label mapping) in each of the nodes along the LSP.

CR-LDP Constraint-Routing LDP is the QoS extension of the basic LDP. Using TCP connections for reliable transmission of control messages the ingress router transmits a LABEL_REQUEST message that contains the route plus some QoS parameters similar to ATM (committed data rate, peak data rate, burst size ...) to the next hop in the LSP. Here, the reservation is being made before the LABEL_REQUEST message is being forwarded and eventually reaches the egress router. If all the reservations could be made up to this node, it will answer with a LABEL_MAPPING message that contains its own outgoing label and is used to set up the LSP when going backwards to the ingress router. This kind of reservation is called *forward reservation* and may result in an unnecessary blocking of requests. To avoid this, *backward reservation* is being used by

RSVP-TE . The Resource ReSerVation Protocol with Traffic Engineering is the second option for signaling in MPLS. Reservations (PATH messages) are here recorded in all intermediate nodes but actually being made only when the egress router sends back the RESV message. RSVP originally used IP routing and therefore needed the -TE extension that allows to set up an explicit path through the network.

To sum it up, the main features of MPLS are:

- *Forwarding Equivalence Classes.* These allow the merging of different IP flows with similar characteristics, thereby reducing the number of VCs (or labels).
- *Label Stacking.* By this, the number of labels in a routing domain can be independent of the number of labels used outside, thereby reducing the size of the forwarding tables.
- *Traffic Engineering.* The ability to set up explicit routes opens the possibility to set up protection LSPs and to compensate for the overload on the shortest path that is usually being produced by IP routing protocols.

After this short excurse into the world of non-WDM link layer protocols, the next section again comes back to the IP over WDM problem we described in section 3.3.

5.6. Multi Protocol Lambda Switching

It soon became obvious that the requirements for signaling in a WDM network are not much different from what is being done in MPLS. In addition, a wavelength in a fiber link can be seen as a special label as well. MPλS was therefore proposed as a framework for optical bandwidth management and real-time provisioning of wavelengths[AR01]. The aim is to integrate some of the functionalities of MPLS LSRs (Label Switch Routers) into programmable OXCs. This means that an OXC shall at least be able to cooperate in the LSP setup by reserving wavelengths.
There is some discussion going on in the IETF working group on how far this integration should go. Currently, there are two antipodes and some mixed model:

- *Overlay model:* Use of different (independent) instances of the control planes in the MPLS LSR and the OXC. The control of the optical network and the IP network is maximally decoupled, much like in today's networks. Static or manual setup of lightpaths.
- *Augmented model:* Or sometimes integrated model. OXCs are IP addressable (they get IP addresses) and are able to map LSPs to wavelengths. The control of the LSP setup is still in the LSR.
- *Peer model:* Only a single control plane spans LSR and OXC. This in effect means a router that is able to set up lightpaths.

Of course, there are pro's and con's for every of these models, please see [DY01] for a discussion of the failure isolation aspects and [BDL+01] for management aspects. When the notion of a label is even more extended, every form of multiplexing can be considered a label. In that sense, an SDH connection (TDM) may be label as well as a wavelength (WDM) as well as a whole fiber (SDM)[BDL+01]. This approach is called *Generalized Multi Protocol Label Switching (GMPLS).*
The aim of this evolution is to reduce the number of control planes in the network. In effect, the IP network providers want to get rid of the ATM, the SONET/SDH and the WDM control plane. On the other hand, the traffic engineering, QoS, protection and path surveillance functions should not get lost on the way. To achieve this, IP routing must be enriched with all the information about virtual circuits (aka LSPs) and QoS, protection paths, available wavelengths and even optical properties of these wavelengths. There are a number of Internet drafts addressing extensions to OSPF and IS-IS. The question is how complex routing decisions can become and still allow for a timely computation of an LSP. And even if the route could be computed in time, it is very unlikely that an LSP can be setup end-to-end (or almost end-to-end) in a fraction of a second. This at least makes the data-driven approach unlikely to happen in GMPLS.

5.7. Optical Burst Switching

Exactly the timing problem was the driving force for a totally different approach by John Turner [Tur99] and Chunming Qiao [QY99]. It was called Optical Burst Switching (OBS) and can be seen somewhere in between wavelength routed networks and optical packet switching. Just like in MPLS, IP packets are classified at the edge of the OBS network. Also, the constraint (QoS enriched) routing has to be done here. Then, instead of building up an end-to-end LSP, a burst reservation message is issued by the ingress router. After a "base" offset time T, the burst follows the setup message. T is the sum of all expected processing delays δ in the nodes along the LSP: $T <= \sum_{h=1}^{H} \delta(h)$. figure 5.6 is taken from [YQD01] and shows the case for H=3.

If the reservation cannot be made, the burst is simply dropped and has to be repeated at a later time. No buffering (e.g. in Fiber Delay Lines (FDL)) is necessary in the nodes, because

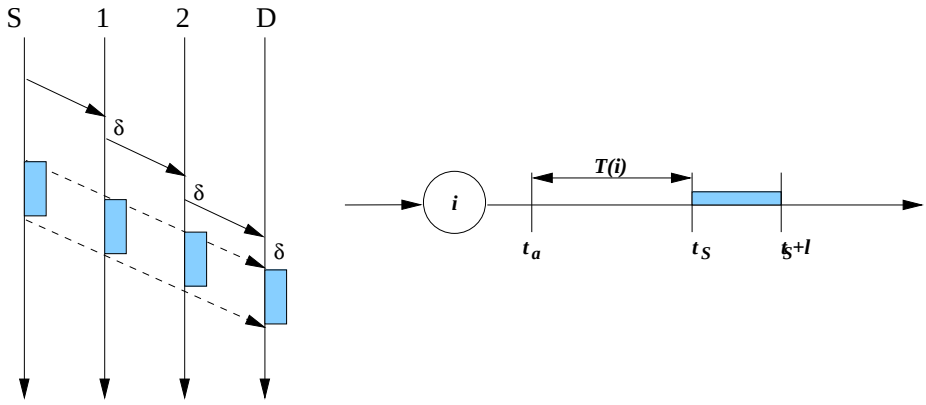

Figure 5.6.: Schematic of JET-based Optical Burst Switching.

the offset time is known in advance. FDLs can however be used to resolve contention in the nodes. The most advanced protocol that is responsible for the timing of the messages is called JET (for Just Enough Time)[QY99]. It differs from other protocols like JIT [BRPS02] and Horizon[Tur99] in that the start and end of the burst is being transmitted to all nodes in the path. By that, more reservations can be accepted than if only the beginning or the end of the bursts are known. By choosing appropriate offset times for different service classes, these can be isolated. The basic idea here is that an additional offset has to be given to the higher prioritized bursts. That is, the time between t_a and t_s for a higher priority class has to be significantly longer than for a lower priority class. That way it is made sure that the reservations for class 1 arrive before the reservation for class 0 (that is assumed to have no extra offset)[2]. For a discussion of several burst switching architectures and their separation of priority classes, please refer to [DGSB00].
There are, however, some potential problems with OBS. First of all, since it is a *tell-and-go* mechanism, there is a higher probability for burst blocking (dropping it somewhere in the network) under high load. Since not only reservations collide but whole bursts, the available bandwidth will be reduced, intuitively leading to some ALOHA-like instability. Unfortunately, no simulation results of a larger network using OBS have been published up to now. Second, the required additional offsets have to be some multiple of the mean burst length to lead to sufficient decoupling. Assuming that the maximum burst length is not fixed, it is hard to predict the end of a lower-class burst, especially with the self-similar traffic characteristics of WWW traffic today[CB97].

[2]This is very much like in real life when trying to make a reservation for dinner. When you know the Maitre d'hotel (that is, you are in priority class 1) and you call in 2 hours before (which is the estimated maximum length of a usual dinner of the people in priority class 0) then it is almost sure you will get a table.

6. Protocols of Optical Packet Networks

Within the next chapter a couple of protocols are introduced that were designed for optical packet networking in the LAN or MAN area. First of all, when speaking about LANs there is the Ethernet, or strictly speaking, its standardized version, IEEE 802.3x. Since the area of interest for the reader is optical gigabit networking, there are two families of Ethernet standards to deal with, namely IEEE 802.3z Gigabit Ethernet(GbE) and IEEE 802.3ae 10 Gigabit Ethernet(XGE). Work on the latter has been finished in March 2002, but except for the newer 64/66 bit encoding and the wide area interfaces (that allow a direct mapping of Ethernet frames into SONET/SDH OC-192/STM-64 containers) there are not so many novelties. The main development is that the actual *shared medium* that has been a synonym for Ethernet for a long time has been dropped for a full-duplex transmission and a purely electronically switched architecture now. This is the main reason why XGE is not being considered here in detail.
Instead, three ring access protocols are introduced in the following. MetaRing and CRMA-II were among the first attempts to guarantee fair and distributed access to the medium in *destination stripping* ring networks. They represent two of the main directions to assign transmission rights to nodes in a token-less ring: While MetaRing cyclically assigns a certain transmission quota to each node, CRMA-II basically relies on the cyclic reservation of bandwidth at a controller node. The third direction – a so-called backpressure mechanism – is represented by SRP, the access protocol of Cisco's Dynamic Packet Transport network. These three main directions – quota, reservation and backpressure – can be found in different hybrid forms and combinations in all of the currently proposed MAC protocols for the IEEE 802.17 RPR standard. Although none of the protocols introduced in the following implements either of the mechanism in a pure form, they serve as prototypes for their families here.

6.1. IEEE 802.3z - Gigabit Ethernet (GbE)

GbE is a part of the IEEE 802.3 family. The standard was approved in June 1998. Many of the features of the classical Ethernet are to be found here, too. Products are reported to be robust and inexpensive. [CL99] gives a good introduction into the standard. IEEE 802.3z defines two classes of connections:1000BASE-LX can work on monomode fibers over a distance of 5000 m and on multimode fibers over 550 m. 1000BASE-SX is only defined for multimode fibers. Additionally there are two copper-based GbE classes: 1000BASE-CX defines connections on 150W STP (Shielded Twisted Pair) up to 25 m link length. The working group IEEE 802.3ab recently finished the work on 1000BASE-T, which works on 4 pairs of Cat.5 Twisted Pair copper lines, all at a data rate of 250 Mbits. Except

for 1000BASE-T, which uses a 5-level PAM encoding, GbE uses an 8B10B coding which produces a raw transmission rate of 1.25 Gbaud at a data rate of 1 Gbit/s.

6.1.1. GbE frame sizes

Since GbE is based on the IEEE 802.3 standard, the frame format had to remain the same as in the classical (10 Mbit/s) 802.3. A minimum frame length of 64 byte was introduced there to allow for a 2 km size of the collision domain.[1] Increasing the data rate by the factor 10 resulted in a tenfold decrease of the collision domain (200 m). To overcome the 20 m limit (2 m for 10 GbE!) that would have been left if nothing except for the data rate would have been changed, some compensation had to be included into the standard.
GbE may be used in half or full duplex mode. The full duplex option means that a NIC (Network Interface Card) is point-to-point connected to a GbE switch via 2 fibers (or wavelengths). Hence, no collision detection is needed anymore and the segment length is only dependent on some PHY layer properties, resulting in the 5000 m mentioned above. In a half duplex configuration (i.e. in the real *shared medium*) a minimum channel occupation time is needed to enable the carrier sensing. At a transmission speed of 1 Gbit/s the packets therefore have to be longer than in the classical Ethernet. This results in a minimum packet size of 512 byte. The official minimum packet size is nevertheless kept at 64 byte and there are two possibilities to deal with this: Send large packets which are almost empty or send packets in *burst mode*. The latter means that after the channel has been acquired by some station it is allowed to send more than one Ethernet frame. In between these frames the channel is kept busy by sending IDLE patterns. There is a burst limit of 8192 bit times, after which the channel has to be released.
To have such a burst limit automatically raised the question of *Jumbo packets*.

6.1.2. Gigabit Ethernet - Jumbo Packets

There were two reasons for the 1518 byte maximum packet length of the classical Ethernet: Error-prone media did not allow for longer packets and the blocking of short packets by longer ones should be limited. Both these reasons are non-existent today anymore, with bit error rates under 10^{-12} and data rates of 1 or even 10 Gbit/s (which reduce the duration of packets). Jumbo packets were first implemented as a non-standard solution by Alteon, followed by 3Com and others. They have not become part of the IEEE 802.3z standard, but for Ethernet compatibility, not for performance reasons. The basic reason for the length of 9000 byte was that NFS uses 8192 byte packets and makes up a large part of the total traffic in local area networks. In addition, Ethernet's 32 bit CRC looses is failure detection capability above 12000 byte or so. Alteon claims a reduction of 50% of CPU utilization while increasing the throughput by 50% through the use of Jumbo packets [Lo98]. Recently

[1] 512 bit at 10 Mbit/s result in 51,2 μs frame length. At 200000 km/s (roughly the speed of the electromagnetic waves in copper) this means that a frame is about 10 km long. Collision detection required a round trip time less than the frame length (so only 5 km are left). Some additional delay is introduced by the (3 in maximum) repeaters, which limits the size of the collision domain to about 2 km and the length of a segment to 500 m.

the discussion about larger MTU sizes gained interest again in the context of 10 GbE. The reason for that is that the number of packets that have to be processed in a Network Interface Card or a switch is again ten-fold compared to GbE. No NIC is able up to now to fill such a pipe, just because of the enormous computing speed that is required to process $10^{10}/12000 \approx 0.83$ Mpackets/s.

6.2. MetaRing - an Insertion Buffer Protocol

MetaRing is a full-duplex fiber ring network operating at a speed of 1 Gbit/s and above that was proposed first in 1990 and in a revised version in 1993 [CO93]. It is an insertion buffer network with quota-based fairness, however there also exists a slotted ring variant. What distinguished MetaRing from its predecessors like TokenRing or FDDI was the *destination stripping* principle, which means that the destination of a data packet takes it off the ring. This leads to a possible *spatial reuse* in that every data packets traverses only the ring links between source and destination, leaving the rest of the ring for other transmissions. Together with the concurrent use of both rings for data and control information, this leads to a potential 8-fold capacity of MetaRing compared to FDDI or TokenRing.[2] The protocol provides two types of services: asynchronous and synchronous. The synchronous traffic has priority over the other. Using special packets that rotate around the ring (ASYNC-GR(een), ASYNC-Y(e)L(low) and ASYNC-R(e)D) the asynchronous traffic is enabled, stopped or prevented from entering the ring. In slotted operation, a signal called ASYNC-EN (asynchronous-traffic-enable) rotates around the ring freely as long as no node starts to store messages in its SYNC-QUEUE. Every node measures the rotation time of the ASYNC-EN signal. Whenever a node "starves", it holds and delays the ASYNC-EN signal for one time slot, thereby indirectly signaling the other stations to stop sending asynchronous traffic, because their rotation timer rises above the usual value.
Single (unidirectional) ring operation is optional. All control messages have to flow into the same direction then, which increases the time to converge.

6.2.1. Fairness Algorithms

To achieve global fairness, a special packet called SAT (for Satisfied) rotates on the counter-directional ring. When a node receives the SAT, it

- transmits data until its transmit queue is empty or quota exhausted
- updates its quota (e.g. by 20k)
- passes on the SAT packet

[2] This is due to the reduced average hop distance that a packet has to traverse to reach its destination. For a bidirectional ring and a uniform load pattern, the mean hop distance approaches N/4 for large N. Given that FDDI uses only one ring in normal operation instead of both for MetaRing, this leads to a factor of 8.

In [CCO93] a modification was proposed to introduce local fairness using REQ(uest) and G(ra)NT packets. The aim is to restrict the area where the SAT is applied to the *congested zone*. The algorithm works as follows:

- When a node *starves*, it transmits a REQ packet upstream.
- Doing so, it creates a *restricted area*, where quota and SAT apply. The node itself is the tail and the next idle node upstream is the head of this zone.
- When all nodes are congested (*starved*), the mechanism is global.
- When the tail node has reached its sending limit (is satisfied), it send a GNT packet upstream and removes itself from the restricted zone. The next upstream node becomes the new tail.
- When the GNT reaches the head of the restricted area, all nodes have moved to normal (unrestricted) operation.

6.3. CRMA-II - A cyclic reservation MAC protocol

The second version of the Cyclic Reservation Multiple Access protocol – CRMA II – was proposed in 1991 [vALSZ91]. Just as MetaRing, CRMA-II relies on a dual counter-rotating ring with the option for a unidirectional ring. Transmission in organized in slots whereby longer packets are taking contiguous slots from the insertion buffer. Two different markings of the slots show their availability: *gratis* (free) slots and *reserved* slots.
A central node (the so-called *scheduler*) cyclically issues *RESERVE* commands. Upon reception of a *RESERVE* message, each node inserts its reservation and waits for a reserved slot. After one round trip the *scheduler* computes the number of reservations and a mean of all transmit counts. It then sends out a *CONFIRM* message with that mean value. Nodes that have a transmission counter higher than that mean have to refrain from sending and let a number of free slots pass. In the next slot following the *CONFIRM* the scheduler sends the *END-OF-CYCLE* message followed by a pre-computed number of *reserved* slots. Each node that has not received as many *reserved* slots as it requested holds this message and releases it only afterwards. Whenever the *END-OF-CYCLE* message returns to the *scheduler*, the transmission cycle is completed and starts again with a new *RESERVE* message.
For a discussion of different fairness algorithms in CRMA-II, see [MCN97].

6.4. Dynamic Packet Transport (DPT)

In the second quarter of 1999 Cisco came up with the first products of a new IP transport technology. Dynamic Packet Transport shall be the next step on the way to a direct interconnection of IP routers. It supports different service classes and is a real *shared medium* optical network.
DPT is built upon two counter-directional fiber rings (mono or multimode). In contrast to

SONET/SDH BLSR/2, where 50% of the overall bandwidth has to be reserved for protection switching purposes, DPT uses both rings simultaneously. The initial products offer a data rate of STM-4/OC-12 on the rings. Network access cards implement the SRP (Spatial Reuse Protocol), which is a buffer insertion MAC protocol.
Additionally DPT provides a number of management functions. These are called IPS (Intelligent Protection Switching) and include:

- 50 ms Protection Switching time limit
  The ring will be folded in the case of a failure. A unidirectional ring is being set up without the need to reroute on the IP level.
- Multilayer Awareness: IPS registers and reacts on errors/alarms on the lower three OSI layers, not only on the physical layer. That way the DPT rings stay intact even if one of the IP routers attached to it fails. Packets to other routers are passed on.
- Plug-and-Play operation: IPS takes over the MAC address assignment and the acquirement of topology information. Special control packets rotate around the rings, gather and provide the topology information. Short-term changes in the topology due to a folded ring can be detected by the appearance of control packets belonging to the counter-directional ring.

DPT supports IP-CoS (Class of Service), in that SRP implements two priorities of data packets, multicasting and the use of the SONET-MIB [BT94] for the surveillance of the physical layer.

6.4.1. Spatial Reuse Protocol (SRP)

SRP is a buffer-insertion-ring-protocol [TS00]. Similar to its predecessors MetaRing [CO93] and CRMA-II it uses both rings and therefore offers at least twice the bandwidth of a SONET/SDH BLSR/2 ring. In contrast to SONET-ADMs the add and drop decision can be made here for each packet. This means, that the receiver of a unicast packet takes this off the ring. Multicast packets stay on the ring and will be stripped by the sender. This possibility to dynamically react on changing traffic patterns potentially offers a far higher gain in bandwidth over the fixed bandwidth assignment of SONET/SDH.

In the first version SRP uses SONET/SDH frames in addition to its own format. This is done to make use of the excellent link monitoring functions of SONET/SDH. More than one SRP packet can be written into one SPE.
Packets that control the transmission on one ring are always being transmitted on the counter-directional ring. Fig. 6.4.1 shows the basic architecture of a DPT ring and a station in the ring.[3] On the basis of the header information (see Fig. 6.4.1) of an incoming packet a station decides whether to take this packet off the ring. Packets that stay on the ring (e.g. multicast packets) are then electronically buffered in one of the parallel queues according

[3]Here only a unidirectional ring is shown.

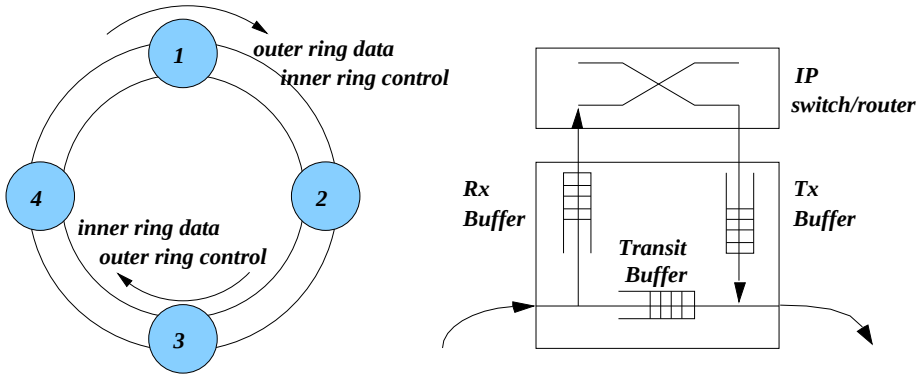

Figure 6.1.: Dynamic Packet Transport (DPT) - basic concept and station design (only one direction shown here).

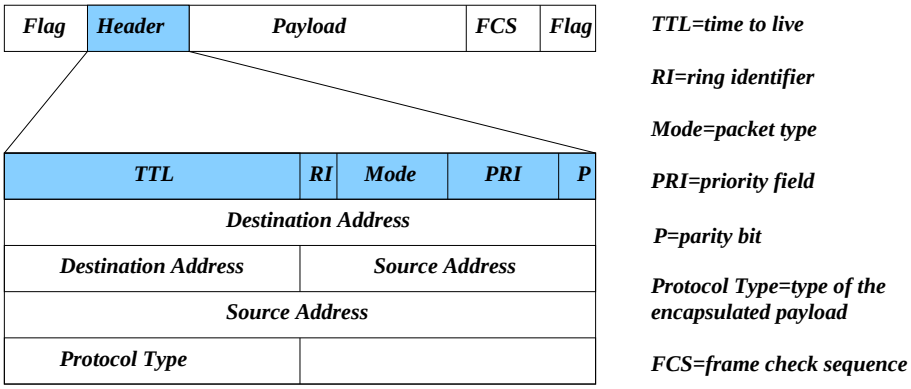

Figure 6.2.: Spatial Reuse Protocol (SRP) Version 2.0 frame format used in DPT.

to their priority.
The generic header size of a SRP version 2.0 is two octets. Data packets consist of the generic header and other fields including a four octets frame check sequence field (CRC-32). Control packets consist of the generic header fields and a one byte control type field. SRP version 2.0 has three types of control packets: The usage packet, topology discovery packet and the intelligent protection switching (IPS) packets.

6.4.1.1. SRP packet handling procedures.

Incoming packets are looked up to determine if they are bound for the Node. If the packet is bound for the Node it is received and passed to the host for processing. If the packet is not bound for the Node it is placed in the transit buffer for continued circulation. Transit Buffer packets and packets sourced from the Node are then scheduled for transmission on the outbound ring according to the SRP fairness algorithm (SRP_fa) (see also section 6.4.2).

SRP performs destination stripping of unicast packets leading to bandwidth gain on the other path of the ring that the unicast packet did not follow. Multicast packets are only stripped by the source.

Receive side packet handling Six things can happen to an incoming packet:

- Packet is removed from the ring i.e stripped.
- Packet is sent to host (layer 3) and removed from the ring.
- Packet is removed and forwarded.
- Packet is a multicast-packet. It is sent to the host (layer 3) and transit buffer.
- Packet is sent to the transit buffer.
- All packets are sent to the transit buffer including control packets.

Receive-side packet handling performs the following: First a node extracts the SRP control information from the incoming packet. Then it checks the *mode* field of the incoming packet to determine if it is a control packet. If a topology discovery packet or IPS packet is received, the packet is stripped and sent to the appropriate processing routine. If a usage packet is received, it is stripped and forwarded to the *mate*[4] which further passes it on to the SRP_fa routine for processing.
A check of the *ring_id* ensures that the packet was received on the appropriate ring. Packets for the outer ring should only be received on the outer ring. Whenever the *ring_id* has the wrong value, this indicates a ring wrap. Packets with a wrong *ring_id* shall not be received. They rotate until they reach the second wrapped node and are directed back on the original ring. That way it is made sure that a packet will not be received twice. If a node is wrapped, packets can be accepted regardless of the ring it is meant for as long as there is destination address match. At last the *destination address* is checked to decide whether to take the packet off the ring.

Transmit side packet handling Transmit side packet handling does the following: First a node determines the priority of locally sourced packets and places them in the appropriate high or low priority transmit queue. Then it selects the next packet to be sent on the ring by choosing between high and low priority packets in the transit buffer and high and low priority packets in the transmit queue. Manages the flow of packets via the SRP fairness algorithm (SRP-fa) that means determine if the node is forwarding or sourcing an excessive amount of traffic and asks upstream nodes to adjust their rates by originating and propagating fairness information or determining if the node is sourcing on excessive amount of traffic and imposing appropriate rate control.
SRP provides support for packet prioritization and expedited packet handling for the transmit queue and transit buffer. The purpose for this is to provide support for the real time

[4]The mate is the MAC instance on the counterdirectional ring.

applications, mission critical applications and control traffic which have strict delay bounds and jitter constraints and therefore require expedited handling.
The *priority* field in the SRP MAC header is set by the node sourcing the packet on to the ring. The value of the priority field is copied from the IP precedence bits in the type of service field. There are only two priority queues (high and low) in the SRP. The node utilizes a configurable priority threshold to determine if the packet should be placed in the high or low priority, transmit or transit queues. Based on the value of the configured priority threshold packets transiting a node can be placed in either the high or low priority transit buffer.
Output scheduling is determined by the transmit side packet processing algorithms. To choose the next packet to transmit the scheduler must choose between high and low priority transit packets and high and low priority transmit packets according to the following order:

- High priority transit packets
- High priority transmit packets from host
- Low priority transmit packet from host
- Low priority transit packets.

The packet priority hierarchy is modified by placing thresholds on the low priority transit queue depth to ensure that the transit buffer does not overflow while serving locally sourced traffic and the low priority transit traffic does not wait too long behind locally sourced low priority traffic.
High-priority transit packets are always sent first, if they exist in the transit buffer. As long as the low transit buffer depth is less than the *high threshold* (which means it is almost overflowing), high priority transmit packets are sent. Low priority transmit packets are sent as long as the low priority transit buffer depth is less than LPBT (Low Priority Buffer Threshold) and *my_usage* is less than *allow_usage* (variables of the fairness-algorithm, see next section). At last low priority transit packets are sent.

6.4.2. SRP_fa - The fairness algorithm

The fairness algorithm called SRP-fa does not use Tokens or SAT-packets like in FDDI or MetaRing, but instead it constantly monitors the number of packets which had to be forwarded to other stations and the number of packets originating from the station. Every station has a fixed maximum rate at which packets may be sent onto the ring. If an overload occurs the stations downstream uses the *usage* field of the SRP header to signal this to the station causing the overload on the counter-directional ring.
The fairness algorithm consists of two functions: A token bucket is used to shape the data rate that a node can emit onto the ring and special packets that control the size of the bucket.

Parameter	Value	Description
MAX_USAGE	594824000	The line rate (here: STM-4 user data rate)
DECAY_INTERVAL	8000	refresh period (number of bytes)
AGECOEFF	4	ageing coefficient
LP_MY_USAGE	512	low pass filter for own usage
LP_FD_RATE	64	low pass filter for forward rate
LP_ALLOW	64	low pass filter for allowed usage
MAX_LINE_RATE	(AGECOEFF *DECAY_INTERVAL)	bucket size
TB_LOW_THRESHOLD	1	low threshold of LP queue

Table 6.1.: Constant parameters of FDL_SRP

my_usage	count of octets transmitted by host
lp_my_usage	my_usage run through a low pass filter
my_usage_ok	flag indicating that host is allowed to transmit
allow_usage	the fair amount each node is allowed to transmit
fwd_rate	count of octets forwarded from upstream
lp_fwd_rate	fwd_rate run through a low pass filter
congested	node cannot transmit host traffic without the TB buffer filling beyond its congestion threshold point.
rev_usage	the usage value passed along to the upstream neighbor

Table 6.2.: Variables of FDL_SRP

6.4.2.1. Variables that are updated every clock cycle

- *my_usage* is incremented by 1 for every octet that is transmitted by the host (does not include transit data).
- fwd_rate is incremented by 1 for every octet that is passed on (for every octet in a transit packet)
- if $((my\_usage < allow\_usage)\&\&(fwd\_rate < my\_usage))\&\&(my\_usage < MAX\_ALLOWANCE))$ $my\_usage\_ok = true$
  true means OK to send host packets.

6.4.2.2. Variables that are updated every DECAY_INTERVAL

- $congested = (lo\_tb\_depth > TB\_LO\_THRESHOLD/2)$
- $lp\_my\_usage = \frac{((LP\_MY\_USAGE-1)*lp\_my\_usage+my\_usage)}{LP\_MY\_USAGE}$
- *my_usage* is decremented by $\min\left(\frac{allow\_usage}{AGECOEFF}, \frac{my\_usage}{AGECOEFF}\right)$
- $lp\_fwd\_rate = \frac{((LP\_FD\_RATE-1)*lp\_fwd\_rate+fwd\_rate)}{LP\_FD\_RATE}$
- fwd_rate is decremented by $\frac{fwd\_rate}{AGECOEFF}$
- *allow_usage* is incremented by $\frac{(MAX\_LRATE-allow\_usage)}{LP\_ALLOW}$

Note that lp values must be calculated prior to decrement of non-lp values.

To show how the algorithm behaves, the amount of octets that a node is allowed to transmit within the following $DECAY\_INTERVAL$ is calculated next as the difference of the newly computed $allow\_usage\prime$ and the new $my\_usage\prime$ Both *my_usage* and *allow_usage* are initially set to zero. The first observation is that *allow_usage* converges to MAX_LINE_RATE:

$$\lim_{t\to\infty} allow\_usage = MAX\_LINE\_RATE \tag{6.1}$$

The value of *my_usage* is between zero and *allow_usage*. When a node did not transmit for a long time (or never), $my\_usage = 0$.

$$allow\_usage\prime - my\_usage\prime = allow\_usage + \left(\frac{MAX\_LINE\_RATE - allow\_usage}{LP\_ALLOW}\right)$$
$$-my\_usage + \min\left(\frac{allow\_usage}{AGECOEFF}, \frac{my\_usage}{AGECOEFF}\right) \tag{6.2}$$
$$\tag{6.3}$$

Applying eq. 6.1 leads to

$$allow\_usage\prime - my\_usage\prime = MAX\_LINE\_RATE \tag{6.4}$$

In the case of a high load, *my_usage* converges to *allow_usage*. Therefore the resulting number of octets is

$$allow\_usage' - my\_usage' = \frac{MAX\_LINE\_RATE}{AGECOEFF} = DECAY\_INTERVAL \qquad (6.5)$$

Cisco stated its interest to make an open standard out of the SRP protocol and submitted this protocol to several standards working groups (like the Optical Internetworking Forum, the IETF and the IEEE). The aim is to create a standard for a packet-optimized data transport. In spring 2001, a new working group IEEE 802.17 was set up to standardize a so-called *RPR (Resilient Packet Ring)*, obviously a result of these efforts. Inside the IETF, a working group named *iporpr* deals with the transport of IP packets over resilient packet rings.

6.4.3. HORNET - An all-optical packet ring testbed

HORNET (Hybrid Opto-electronic Ring NETwork) [SSW+00] is a testbed of advanced packet technologies and protocols. It was developed at UC Stanford. It is a step further compared to the SRP and IEEE 802.17(RPR) in the sense that it realizes a real all-optical WDM ring meaning that the payload is not O/E/O-converted in each node. The main features of the network are:

- bi-directional fiber ring: As in SRP/RPR, 2 fibers transport data traffic counter-rotating. Shortest-path routing reduces the mean hop distance to $\frac{N+1}{4}$ for N nodes.
- TT/FR: Each node is equipped with a very fast tunable laser (tuning times of 15 ns have been demonstrated over a range of 30 nm) and a fixed receiver that drops the destination wavelength of that node.
- Sub-carrier multiplexing (SCM): Every packet that is transmitted onto the ring carries a tone on a subcarrier that is located outside the spectrum of the data packet. Every node is able to decide (by tapping a small amount of power from the ring) which of the wavelengths is occupied. After tuning its transmitter onto that wavelength a packet may be sent to the terminating node.
- CSMA/CA: The access control protocol is named CSMA/CA (CSMA with collision avoidance) and is derived from the CSMA/RN protocol. A node that has found a free space on the desired wavelength may start to transmit a packet there. If, however, a packet arrives while the node is still busy transmitting, it has to stop its transmission and continue at a later point in time.

The data on all wavelengths collides at baseband leaving the sub-carrier frequencies intact. Therefore, a missing sub-carrier at a given frequency indicates that the corresponding wavelength is empty and can be used for transmission. It is not necessary to evaluate every packet header in HORNET because the destination node drops the whole wavelength and no dropping of packets by intermediate nodes is possible.

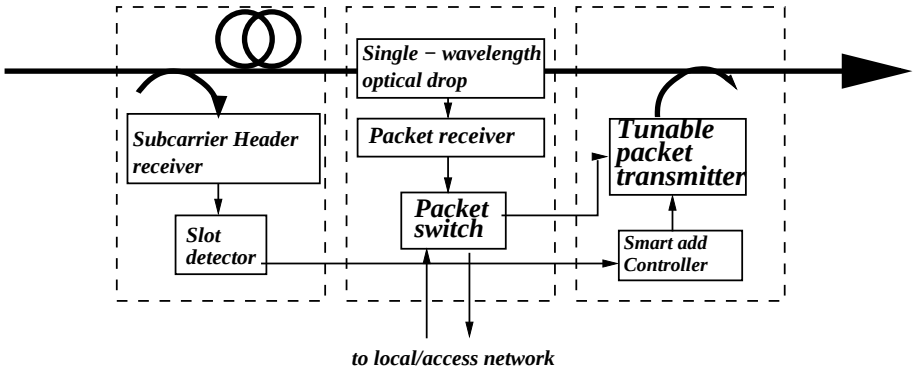

Figure 6.3.: Schematic of an Access Node in HORNET.

6.4.3.1. Node architecture

The architecture of a HORNET access node can be seen in Fig. 6.4.3.1. An optical splitter separates a small amount of power to detect the SCM headers. A Fiber Delay Line (FDL) is then used to store the packets on all wavelengths simultaneously until the decision about a free slot can be met. The middle part of the node terminates the home wavelength of that node and converts it opto-electronically. The right part is made up of the fast tunable transmitter and a controller that decides about wavelength and time slot to transmit the next packet.

6.4.3.2. Access Protocol

The small delay lines require a *cut off* (also called *backoff*) of packets in transmission whenever ring traffic arrives. Several ways to deal with this are proposed in [SSW+00]. All are based on the carrier sense mechanism described above, but differ in the length of the packets and hence, the length of the FDL.

- Slotted ATM cell transmission. The FDL has to be around 5 meters only to satisfy for the processing delay. The node listens at the beginning of the slot and then selects a free wavelength to transmit a cell.
- Unslotted ATM cell transmission. The FDL has to be around 40 meters (assuming 2.4 Gbit/s line rate) to ensure that a whole ATM cell may be transmitted before the incoming cell leaves the FDL.
- Slotted IP with multiple slot sizes Slot sizes of 40, 552 and 1500 byte are generated by a central controller node. These slot lengths are motivated by the packet length distribution in current Internet measurements (cf. Fig. 4.3).
- Unslotted IP with backoff 40 byte packets are transmitted without backoff

7. WDM packet networks

After providing the necessary information about the physical layer and IP networks based on wavelength routing, this chapter will go one step further on the way to ever shorter lightpaths or bursts. In other words, we are dealing with WDM packet networks now. Traditionally, the concept of WDM networks has been different for WANs and LANs. While in WANs, WDM has already entered the market, this technology is used as a simple point-to-point extension of conventional fibers between IP routers. The use of WDM as another dimension to share the medium was traditionally only considered in LANs. This has two main reasons:

- The IP routers in a WAN already do the multiplexing of different traffic streams, which makes it less necessary to have a shared medium
- The wavelengths can be controlled much better in the local area. In addition, common optical amplifiers are band-limited (around 45 nm for EDFAs), which limits the available number of wavelengths. So without the necessity of EDFAs, many more wavelengths can be used.

7.1. WDM Packet Local Area Networks

Mukherjee gave a classification of WDM LANs in 1992, which is basically being used in the literature until today [Muk92a], [Muk92b]. He distinguished between two classes of logical architectures for the local and metropolitan area networks, namely the *single–hop* and the *multihop* networks.

7.1.1. Physical architectures of WDM LANs

. . . can be arbitrary, but in most cases Passive Star Coupler (PSC) architectures are assumed. Physical ring or bus architectures are also possible, but provide generally worse signal attenuation figures. Fig. 7.1 shows how a 8x8 passive star coupler can be constructed out of 12 2x2 (3dB) couplers. The number of 2x2 couplers that an incoming signal has to traverse is $ld(N)$ for N input ports, whereas it is in average $N/2$ in a ring and $(N+1)/2$ for the bus. Since the attenuation of a signal directly depends on the number of optical splitters it has to cross, the number of stations that can be attached to an optical ring or bus is generally lower than for a PSC. Moreover, the signal power at the output of the PSC is independent of the position of the input and does not need to be adjusted.
Recently, physical star topologies have been proposed that use a AWG instead of a PSC

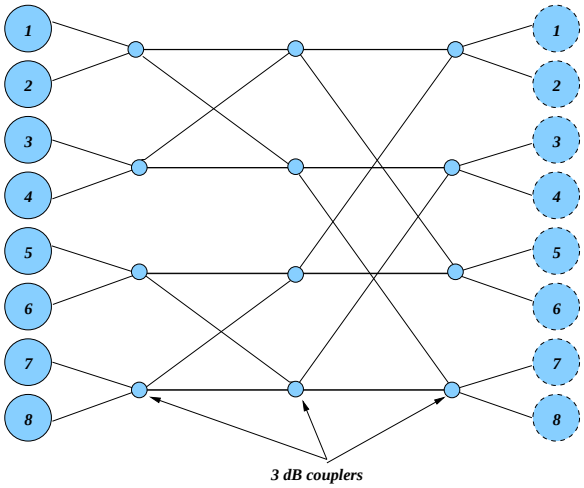

Figure 7.1.: A passive star coupler

[Woe97], [BJM99], and [MRW00]. These will be described in detail in section 7.2.2 and chapter 8.

7.1.2. Logical Architectures of WDM LANs

The number of hops in a network that a data packet has to traverse from its origin to its destination clearly has an effect on the design of such networks. We define a *link* as the physical connection between two nodes in the network. A *path* is the potential connection between any two nodes, thus there are $N(N-1)$ paths in a network of N nodes. In a fully meshed network, the number of links equals the number of paths (and is $N(N-1)$, as well). It is obvious that the mean number of hops (links that a packet has to traverse) is inversely proportional to the number of links in a network. So, to keep the number of possible links between senders and receivers high, in a single–hop network either transmitters or receivers (or both of them) have to be tunable. This way, all links are available, but *not at the same time*. In a multihop network, the number of links is $N \cdot d$ with d as the degree of the network. The links are availabe all the time, but multiple paths share a certain link.
Both classes of networks have their strengths and limitations, and we will take a closer look on them in the next sections.

7.2. Single Hop Networks

The basic principle of a single hop network is to first arrange for the transmitting and the receiving node to send and receive on the same frequency, respectively. In most cases this results in a cyclic operation of these networks consisting of an announcement (reservation) phase and a data phase. Whether or not there is a control channel is another issue in single hop networks. Mukherjee classified single–hop networks according to the number and type

of transmitters and receivers per node:

$$\begin{cases} FT^iTT^j - FR^mTR^n & \text{no pre-transmission coordination ,} \\ CC - FT^iTT^j - FR^mTR^n & \text{control channel based system} \end{cases}$$

where a node has i fixed transmitters, j tunable transmitters, m fixed receivers and n tunable receivers.

As it could be seen in table 2.10, the tuning times of existing optical filter technologies do not allow for a rapid packet switching right now, and there is doubt that they ever will. So most of the single–hop networks are essentially fast circuit-switched networks.

7.2.1. Access protocols for single–hop networks

There are a number of surveys on MAC protocols for single–hop WDM networks (e.g. [MRW02]) and it is not the intention of this chapter to repeat this work here. The general idea is to make the transmitter and/or the receiver of a node able to tune to another wavelength. The problem is to find a distributed algorithm that allows to coordinate both. As stated above, there are systems with and without pre-transmission control. If there is no dedicated control channel in a network, a fixed assignment of wavelengths and time slots to the $N \cdot (N-1)$ potential pairs of communication can be made, given that there are N nodes in the network. The main concern here is that there are no channel and receiver collisions.[1] Additional constraints may be the tuning range or time.[2] These schedules traditionally result in a rather low data rate per communication pair but in optimum throughput for a balanced load. They are easy to implement in firmware but require a precise synchronization.

Random access protocols have been developed that try to assign the bandwidth in a more dynamic way. Most of these protocols perform some kind of ALOHA protocol in both frequency and time. Although ALOHA's maximum throughput is known to be very limited ($1/2e$ or $1/e$ in the case of *slotted* ALOHA), no better access strategy seems applicable. A CSMA (Carrier Sense Multiple Access) mechanism is mostly not feasible because of the very high channel bandwidth these systems are designed for that limits the length of the fiber links. To reduce the complexity of the algorithm and to decrease the cost the random access protocols for broadcast–and–select networks like the PSC mostly $TT - FR^x$ architectures have been developed. This means that there is a so called *home channel* for each node that may be coded in its address such that every other node knows in advance on which wavelength to transmit to this node. The existence of a *home channel* limits the number of nodes in the network to the number of wavelengths available.

Systems that employ pre-transmission control via a common control channel may scale better, as long as the capacity of the control channel is large enough to accommodate for all reservations that are made by the nodes. In the slotted ALOHA/ALOHA protocol, the

[1] The term *receiver collision* in a WDM network means that a tunable receiver is busy receiving a packet on one wavelength while another packet for it arrives on another wavelength. This packet is lost.

[2] It may be faster to tune to a neighboring channel than over the whole tuning range.

node transmits a control packet in a time slot on the control channel and starts to transmit the corresponding data packet on a randomly chosen data channel afterwards. The number of the data channel is transmitted together with the destination address in the control packet.
Perhaps the most influential MAC protocol for single–hop networks was the DT–WDMA (Dynamic Time–Wavelength Division Multiple Access) protocol [RS98]. Here, each node transmits data on a dedicated wavelength, the architecture is a $CC - FT^2 - FR/TR$. One FT/FR pair of each node is tuned to the control channel while the remaining tunable receiver is freely tunable over the whole range of wavelengths. The control channel is divided into N minislots, each one assigned to a certain node (and channel, therefore). There are no collisions in the control channel and no collisions in the data channels, but still the maximum throughput is shown to be $(1 - 1/e) \approx 0.63$ because of receiver collisions.

7.2.2. Single–Hop networks based on AWG

The AWG offers potentially an N–fold capacity compared to the PSC. This results from the wavelength routing property that allows for the spatial wavelength reuse of wavelengths. It requires both the transmitter and the receiver to be tunable, because there is only a single wavelength that may be used to communicate between any pair of nodes. Fixed assignment schemes have been proposed by Borella et al. [BJM99]. Here, a number of M nodes is attached to a combiner/splitter that itself is attached to one of the N input ports of the AWG. The numbers M and N correspond to S and D in Figure 7.2.
A system architecture and a MAC protocol employing pre-transmission control is proposed in [MRW00] and [MRW02]. Because of the impossibility to have a dedicated control channel in an AWG network a broadband LED is used to transmit the control information through the AWG. The signal is spectrally sliced by the AWG such that a fraction of the power of the control signal appears on every output. To distinguish data from control traffic the control signal is spread using a direct sequence spread spectrum code (see section 10.1.1). The node consists of a $CC - TT - TR$, where the CC is made up by the LED. The MAC protocol employs a *reservation ALOHA* in the control channel. The receiver cyclically tunes to each wavelength to receive the control packets from each input port of the AWG. Just like in the previous example, a number (here: S) of nodes is attached to a passive combiner/splitter before going to one of the D input ports of the receiver. It is proposed to use more than one FSR (Free Spectral Range) of the AWG, thus exploiting its periodicity for parallel transmission.

7.3. Multihop Networks

Multihop networks usually consist of a small number of fixed lasers that set up paths between sets of stations that follow certain structural criteria. We can distinguish between *irregular* and *regular* multihop networks. Irregular networks are all networks that do not

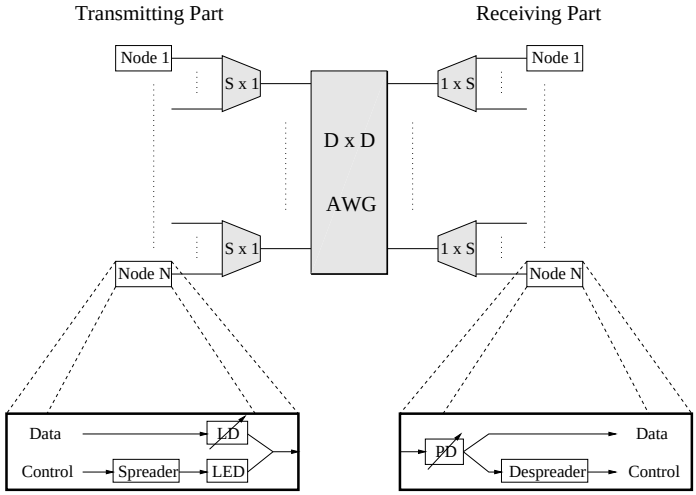

Figure 7.2.: Single hop network as proposed in [MRW00]

have an underlying node-connectivity pattern.[3] Although irregular networks can address certain optimization criteria (like differing load on certain links) directly, in general they do need sophisticated routing and path protection schemes. The process of designing an irregular multihop topology out of a given traffic matrix is called *virtual topology design*. A review of virtual topology design algorithms can be found in [DR00]. In fact, the issues regarding irregular multihop networks have already been addressed in chapter 3.3, since in this regard there is no structural difference between WANs and LANs.

7.3.1. Regular Multihop Networks

Traditional multihop networks were developed to both increase the number of possible paths between two nodes and to balance the (a priori unknown) load in a network. The first application field of regular multihop networks were multi-processor interconnection networks. A high number of links between physically close nodes was essential to these networks. It would for economical reasons not be feasible to actually build a LAN that would be meshed so densely. With the advent of WDM it became possible to avoid the cost of multiple physical ports and cabling per node in exchange for *fixed wavelength* channels. The first WDM multihop networks were seen as traditional *store–and–forward* networks and thus implicitly assumed an O/E/O conversion in each node and a switch that could decide about the way a packet had to take. In this sense, full wavelength conversion was assumed in each node. It is possible to embed every multihop network into a physical star or bus architecture by simple assignment of wavelengths to links. This way, an $S_{(2,2)}$ ShuffleNet as depicted in Fig. 7.6 could be implemented on a PSC using 16 wavelengths.

[3] It is not really clever to define irregular as not being regular, but in practice most larger networks are irregular.

Regular multihop networks follow certain patterns in the establishment of the logical connections. They usually employ much easier routing and protection schemes, but have their problems concerning the scalability of the network. Most of the regular multihop networks scale only very coarsely. For instance, when a ninth node shall be added to an existing $S_{(2,2)}$ ShuffleNet, the next step would be an $S_{(2,3)}$ net with 24 nodes. Most of these networks can be imagined as a three-dimensional geometric figure. Many of them incorporate rings.
Banerjee et al. gave a survey on regular multihop architectures in [BJS99]. To avoid redundancy, only a few of the more popular networks are mentioned in the following. A parameter to make the different multihop architectures comparable is the mean hop distance $\overline{h}$. This describes the number of nodes a packet has to traverse on its way from source to the point where it is removed from the network. This number is analytically tractable for many architectures and inversely influences the total capacity of a network. Because of that, we will make a statement concerning $\overline{h}$ wherever possible.
Since ring networks are a main accent of this work, a special section is devoted to them before the general introduction into multihop networks that is given in the following subsections. A generalization of many of the known regular multihop patterns is the concept of Cayley graphs. To conclude this chapter, the basic properties of Cayley graphs are introduced. We will come back to Cayley graphs in chapter 11.

7.4. Packet Ring Networks

The simplest form of multihop networks is the unidirectional ring. It provides full connectivity with only a single (fixed) receiver/transmitter pair per node. Consider a ring network of N nodes. Depending on the access strategy, the mean number of hops a packet has to be forwarded is either $\overline{h} = h = N$ or $\overline{h} = \frac{N}{2}$. The first means that every packet will be removed by its originator (also called *source stripping*) while the latter is the case for *destination stripping* networks. From now on we consider the latter exclusively.

7.4.1. Bidirectional rings

Adding one additional ring not only increases the possibility for the network to survive a node failure, but also decreases the mean hop distance. If this ring is counter-directional to the first, the mean hop distance drops to $\frac{N}{4}$ for even N and $\frac{N+1}{4}$ for odd N, respectively. Bidirectional ring are optimal in the case of wavelength continuity, that is where no wavelength conversion is possible in a node. The assumption of wavelength continuity simplifies the node structure. A node has only to decide if a packet is destined to itself. If so, the packet has to be taken off the ring, if not, it is left on the ring. Both rings can be operated independently. Because of its simplicity these rings are often used in practice (like SONET rings, MetaRing, SRP).

7.4.2. Multiconnected Rings

Assuming the capability of a packet to change the ring, other multi-connected ring architectures provide a lower mean hop distance and thus a higher capacity. The *Wheel*, proposed by Guo and Acampora in 1996[GA96], can be seen as a generalization of the Forward Loop Backward Hop (FLBH) concept. In the latter, a node is connected to the nodes one "forward" and S backward for some S (the Skip distance). It was shown in [PT94] that the optimum value for S concerning the maximum fault tolerance (number of disjoint paths in the network) is slightly less than $\sqrt{N}$, while it is exactly $\sqrt{N}$ concerning the mean hop distance. The mean hop distance in this case equals to $\overline{h} = \frac{N(\sqrt{N}-1)}{N-1}$.
The *wheel* allows for more than one skip distance, leading to a skip vector of

$$[1, \sqrt[r]{N}, \sqrt[r]{N^2}, \ldots, \sqrt[r]{N^{r-1}}]$$

with r being the degree of the node. A *wheel* of degree 2 is shown in Fig. 7.3. It can be implemented on a fiber ring using 3 wavelengths. The nodes have to be equipped with passive optical multiplexers that drop and add a subset of the available wavelengths, here, 2.

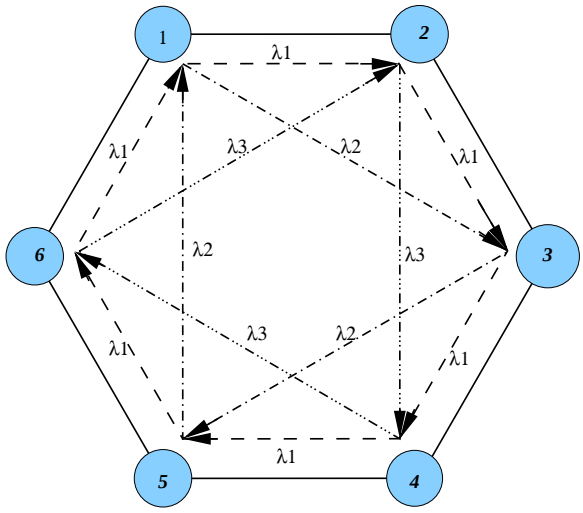

Figure 7.3.: The "Wheel" as proposed in [GA96].

7.4.3. DeBruijn Graph

In [SR94] the authors propose de Bruijn graphs as logical topologies for multihop lightwave networks. A de Bruijn graph $G(\Delta, D)$ has $N = \Delta^D$ nodes and diameter D. The nodes have labels or addresses of length D digits out of $0, 1, 2 \ldots \Delta - 1$. The connectivity in a de Bruijn graph follows the operation of a shift register. There is an edge from node i to node

j iff the state of a shift register that represents i can be transformed to state j by one shift operation to the left.[4] The degree of the node is Δ. There are also Δ nodes that show self-loops as it can be seen in Figure 7.4. Bounds for the mean hop distance are given in [BJS99] as:

$$D\frac{N}{N-1} - \frac{\Delta}{(\Delta-1)^2} + \frac{D}{(\Delta^D-1)(\Delta-1)} \leq \bar{h} \leq D\frac{N}{N-1} - \frac{1}{\Delta-1} \tag{7.1}$$

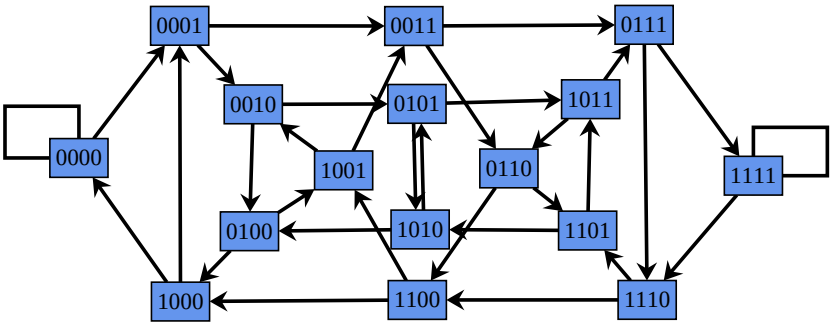

Figure 7.4.: A (2,4)-deBruijn graph.

7.4.4. Manhattan Street Network

The Manhattan Street Network (MSN) was developed by Maxemchuck in 1985 [Max85]. A two-dimensional MSN is a torus network. It consists of m rows and n columns. The direction of the links resembles the geographic topology of the streets and avenues of Manhattan. As can be seen in Fig. 7.5, there are actually m horizontal and n vertical rings. Every station has to be physically connected to two rings, one "horizontal" and one "vertical". There is no closed form for the mean hop distance of arbitrary m and n, but for large N and $m = n$, $\overline{h} \to \sqrt{N}$.

7.4.5. ShuffleNet

The perfect shuffle topology was proposed by Stone in 1971 for parallel processing, while Acampora in 1987 first proposed the ShuffleNet architecture for virtual WDM networks. [Aca87]. A (p,k) -ShuffleNet consists of k columns of p^k nodes each. It can be imagined as if the last column would we wrapped to connect to the first. A (2,2)-ShuffleNet is shown

[4]The direction of the shift is a convention. A shift to the left means to add one out of Δ digits from the right and drop the leftmost digit.

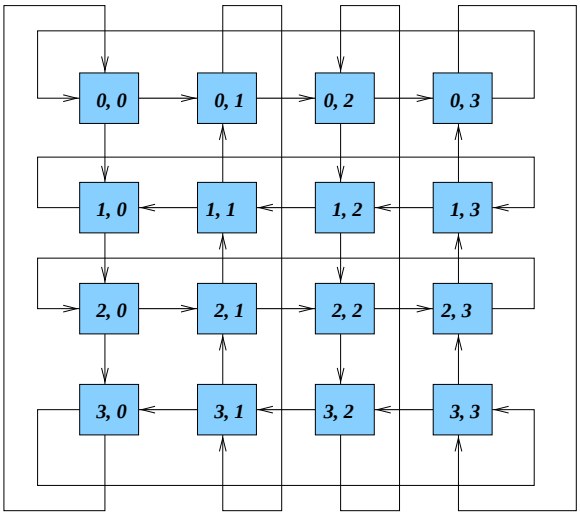

Figure 7.5.: 16 node (4x4) Manhattan Street Network

in Fig. 7.6. It has been shown[GGA95] that a S(k,p) ShuffleNet can be transformed into a Hypercube H (k,p) "multiplied" with a ring R(k). This means that a node in the Hypercube is replaced by a ring of k nodes with a constant degree p.

The average hop distance $\overline{h}$ between two arbitrary nodes in a (p,k)-ShuffleNet is given as:

$$\overline{h} = \frac{kp^k(p-1)(3k-1) - 2k(p^k-1)}{2(p-1)(kp^k-1)}$$

The diameter D is $2k-1$.

7.5. Optical networks based on Cayley graphs

7.5.1. Motivation

As explained in detail later on Cayley graphs have two major properties making them useful for researches on network topologies. First they cover (in a special way) the class of symmetric interconnection networks. These networks are of special interest as they naturally lead to uniformly distributed network loads. Secondly they connect graph theory with algebraic group theory, allowing thus the use of algebraic results on finite groups in graph theoretical problems.

This chapter reviews the notion and general properties of Cayley graphs mainly as presented in Akers/Krishnamurthy in [AK89].

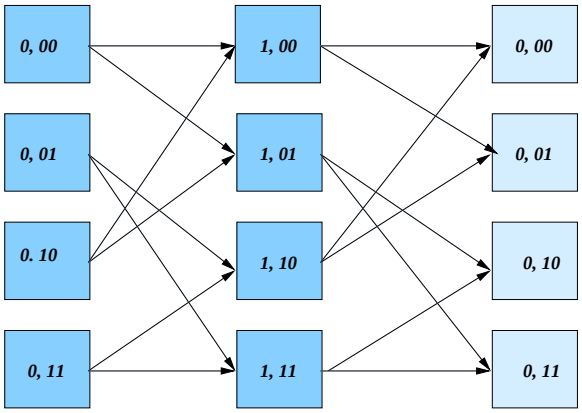

Figure 7.6.: A (2,2) ShuffleNet

7.5.2. Definition

A graph C =(V, G) is a (directed) Cayley graph with vertex set V if (V,*) is a finite group with $G \subset V \setminus \{I\}$ and the following condition holds for every two vertices (cf. [Big74]):

$$\text{Vertex } v_1 \in V \text{ is connected to vertex } v_2 \in V. \Leftrightarrow v_1 = v_2 * g \text{ for some } g \in G.$$

G is called the generator set of the graph. The set E of edges of the Cayley graph is given by

$$E = \{(v_1, v_2) \,|v_1, v_2 \in V, v_1 \text{is connected to} v_2\} = \{(v_1, v_1 * g) \,|v_1 \in V, g \in G\}\,.$$

The dimension of V is therefore given by the number of elements of G. The Cayley graph can be viewed as undirected iff $g^{-1} \in G$ for every $g \in G$. As (V,*) is a finite group it isomorphic to a subgroup of some permutation group. Thus every (V,*) may be described by a set of permutations, where the group product is naturally defined through composition. But it also may be convenient to use other representations, especially when restricting oneself to special groups (cf. e.g. [Tan94])

7.5.3. Vertex and edge symmetry

A graph is said to be *vertex symmetric* if for every pair of vertices v_1, v_2 there exists an automorphism of the graph that maps v_1 into v_2.

A graph is said to be *edge symmetric* if for every pair of edges e_1, e_2 there exists an automorphism of the graph that maps e_1 into e_2.

An automorphism of a graph (V, E) with vertex set V and edge set $E \subset V \times V$ is a mapping $f : V \to V$ together with the naturally induced mapping $F : E \to V \times V, (v_1, v_2) \mapsto (f(v_1), f(v_2))$ where f is bijective and F correspondingly maps E one-one onto E.

Every Cayley graph is vertex symmetric, since the mappings $f : v \mapsto v_2 v_1^{-1} v$ are automorphisms that map v_1 into v_2.

The condition for edge symmetry is as follows: Let the Cayley graph C = (V, G) be represented by a subgroup V of the group of permutations of n symbols and some adequate generator set G. C then is edge symmetric, iff for every $g_1, g_2 \in G$ there exists a permutation of n symbols that maps G into itself and g_1 into g_2.

7.5.4. General symmetric interconnection networks

As mentioned above (vertex) symmetric interconnection networks are of some importance for network research and design. All Cayley graphs are vertex symmetric. But there are vertex symmetric graphs, which cannot be modeled by a single Cayley graph.

To model all symmetric networks the notion of quotient graphs has to be introduced.

Let V be a finite group generated by a given set of generators G. Given a subgroup W of V, the Cayley graph (V, G) can be reduced to a graph called the quotient of V by W. This reduction is done by first splitting V in disjoint sets, the set W and its left cosets, i.e. the sets $\{wv|w \in W\}$ for any $v \in V$, then replacing these sets by a single vertex and connecting these vertices iff there has been an edge in the original Cayley graph (V, G) from one point belonging to the subset corresponding to the one of the new vertices to one point belonging to the subset corresponding to the other of the new vertices.

It is shown in [Sab68] that every vertex symmetric graph can be represented as the quotient graph of two groups V and W. With $W = \{I\}$ the resulting quotient graph is identical with the original Cayley graph.

7.5.5. Hierarchical graphs and fault tolerance

Some Cayley graphs can be decomposed step by step in identical subgraphs which are connected via edges all corresponding to the same, fixed generator in every step. These graphs are called *hierarchical*. A graph is hierarchical iff the generators can be ordered in a way, so that no generator is included the subgroup generated by its predecessors. If this condition holds for every order of the generator the graph is called *strongly hierarchical*.

As shown in [AK89] hierarchical graphs are maximally fault tolerance,i.e. the number edge which can be cut in any case without destroying the connectivity of the graph is degree of the graph minus one.

7.6. Multiconfiguration Multihop Protocols (MMP)

To overcome the limitations of both, single-hop and multihop approaches, Jue and Mukherjee proposed MMPs in 1998[JM98]. The basic idea is that single-hop networks require a reconfiguration for every packet and thus suffer a large penalty for the tuning time. If the reconfiguration would be performed less frequently, the tuning penalty would decrease. Multihop networks in contrast suffer from the large mean hop distance that is a function of the degree of the node and the interconnection pattern. A possible way to take advantage of both approaches while avoiding the problems is to switch the network configuration between several multihop patterns.

An easy way to illustrate this is presented in the paper: Assuming that every node is equipped with one FT–TR pair, it would be possible to arrange all nodes in a unidirectional ring. After a certain period it is now possible to re-tune all receivers to another configuration, namely the counter-directional ring. In result, a logical bidirectional ring would appear with almost twice the capacity. This cycle of two configurations is then repeated infinitely. Compared to a single-hop network the tuning would be performed only 2 times instead of N times, thus reducing the penalty. Simulations showed a tradeoff between the tuning time and the length of the cycle. Neglecting tuning time, assumimg an ideal MAC protocol and a uniform load, a single-hop network is optimal. However, the larger the tuning time is, the less reconfigurations should be performed in the network.

8. PrimeNet - A ring network based on AWG

8.1. Introduction

In this chapter a novel network architecture based on an Arrayed Waveguide Grating is proposed. The previous chapter introduced the two directions in local and metropolitan area WDM networking. We have seen that single-hop networks employ complicated MAC protocols to arrange for the pre-transmission coordination. Multihop networks mostly suffer from the large mean hop distance. We show that the AWG enables both, a simple MAC protocol and a low mean hop distance. The main reason for this is the potential n-fold capacity of such a network compared to a passive star coupler. Due to the spatial wavelength reuse that is possible in an AWG it is possible to use one wavelength for many parallel transmissions that do not share the same sender and receiver, respectively.
The basic element of the network has been introduced in chapter 2.11. The next section shows how to set up a logical ring on each wavelength that can be operated independently. An interesting phenomenon that results from the cyclic permutation of the wavelengths in an AWG is the need for a *prime number* of in– and output ports at the AWG. It is explained in short why prime numbers are advantageous to balance the load in the proposed network. The resulting nodal design is shown in section 8.3. To give an estimation about the possible physical size of such a network the transmission line is analyzed afterwards using a linear attenuation-based model with additional noise terms from the necessary amplifiers and receivers in the system. It is concluded that the architecture may be used in the local or metropolitan area, provided that additional amplifiers compensate for the loss of optical power in the network.

8.2. Basic Network Structure

In the architecture proposed here the AWG is used in a physical star topology. The network structure is a set of virtual rings on the underlying physical star topology. These rings may be used independently from each other in the sense that a packet does not change the wavelength (=ring) on its way from source to destination.
As explained in section 2.11, the AWG is a wavelength selective device, that is, a wavelength on an input of an $N \times N$ AWG appears only on one output. The advantage of this is that this passive device offers N times the bandwidth of a passive star and it is completely collision free.
We will now come back to the notation of the wavelength transfer matrix introduced in equation 2.4 in section 2.11. Again, without loss of generality we assume a (5×5) AWG. After multiplication of the wavelength output matrix $O_{5,5}$ with an appropriate selection

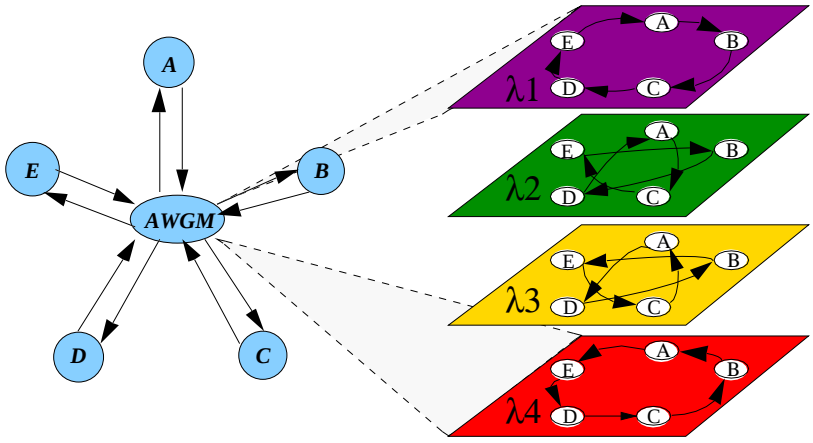

Figure 8.1.: Connections in a network of 5 nodes using 4 wavelengths.

matrix $S_{5,5}$ from the left (which means a simple exchange of the rows) the resulting output matrix looks like:

$$S_{5,5} * O_{5,5} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} A_1 & B_2 & C_3 & D_4 & E_5 \\ E_1 & A_2 & B_3 & C_4 & D_5 \\ D_1 & E_2 & A_3 & B_4 & C_5 \\ C_1 & D_2 & E_3 & A_4 & B_5 \\ B_1 & C_2 & D_3 & E_4 & A_5 \end{pmatrix}$$

$$= \begin{pmatrix} A_1 & B_2 & C_3 & D_4 & E_5 \\ B_1 & C_2 & D_3 & E_4 & A_5 \\ C_1 & D_2 & E_3 & A_4 & B_5 \\ D_1 & E_2 & A_3 & B_4 & C_5 \\ E_1 & A_2 & B_3 & C_4 & D_5 \end{pmatrix} = O'_{5,5}$$

The last step means nothing but an exchange of the outputs of the AWG, but it leads to an interesting conclusion, assumed that a station A is connected to the first pair of input/output ports, station B to the second and so on. Wavelength λ_1 is always routed back to the station where it came from, so it can not be used for the transmission to other stations, but the other four wavelengths now form unidirectional rings with all stations connected to all rings. This is shown in figure 8.1.

Note that the rings on wavelength λ_2 and λ_5 are counterdirectional as well as λ_3 and λ_4. The resulting connectivity pattern can be viewed as a fully meshed interconnection, too. It can be seen that potentially all of the wavelengths can be used for a transmission between a given pair of stations. Therefore the overall user data rate for an AWG with N inputs (that is, N stations in maximum) is $N-1$ times the bandwidth of a single channel.
The shortest path (output wavelength λ) from a sending node to the destination of the

packet is determined by equation 8.1:

$$\lambda = \frac{x \cdot N + distance}{h} \qquad \begin{array}{r} \text{for integer numbers h, N and x with:} \\ h = \text{hop number, } 1 <= h < N \\ x = 0 \ldots N \\ distance = (N + n_{out} - n_{in}) \mod N \\ \text{distance between receiver and sender at the input} \\ n = \text{port numbers} \end{array} \tag{8.1}$$

The number of hops should initially be set to $h = 1$ and $x = 0$. If the wavelength $\lambda = distance$ should not be available for any reason, h is to be incremented and x to be varied until λ is an integer. In a case where N is an integer multiple of h or λ, there are wavelengths which can not be used for transmission to a certain node. This is shown in figure 8.2, where λ_2 makes up two separate rings (A-C and B-D) which are not connected to each other, i.e. they do not share a common node. A variation of eq. 8.1 in the form of

$$(\lambda \cdot h) \mod N = distance$$

illustrates the problem better. With $N = \lambda * x$, h can only go up to $h = (x - 1)$. For $h = x$, the $distance = 0$, which means that the node reaches itself and the sub-ring (of size x) is completed. This feature could be used to set up subnetworks, but in our approach it is considered unwanted. Therefore the conclusion is that the number of nodes N in the network and hence the number of inputs of the AWG has to be a prime number[1]. With N being a prime number the network consists of $N-1$ parallel rings with all nodes connected to all rings. The AWG has the property of being periodic, i.e. for an $N \times N$ AWG wavelengths λ_x, λ_{N+x}, $\lambda_{2N+x} \ldots$ are all being routed to the same output. The number of periods is limited by the higher attenuation of frequencies far away from the center frequency. This feature allows for the use of more than N wavelengths and may actually enable the parallel transmission of more than one bit[2]. As can be seen later on, packet headers could be transmitted within the next period as well. We do not consider the periodic nature of the AWG in the above equation. To take this into account, x would have to go up to $2N$, $3N$ or higher, depending on the number of periods (FSR).

In general, there is no need to have as many Transmitter/Receiver-pairs as wavelengths in the system. It is possible to start up with only one fixed Tx/Rx pair per node, which results in a unidirectional ring (e.g. only using λ_1 in figure 8.1). Adding additional Tx/Rx pairs increases the possible throughput of each node. Thereby the available bandwidth between two endpoints can be scaled to the actual needs. An analysis of the available capacity follows in chapter 9. If N is a prime number, the wavelengths λ_x and $\lambda_{N-x} (x = 1..(N-1)/2)$ form counterdirectional rings.

[1] Hence the name!

[2] There would be problems with the chromatic dispersion over longer distances when transmitting a number of bits in parallel. To realign the octets, some kind of signal processing would have to be done in the receiver. We will not follow this line of discussion here.

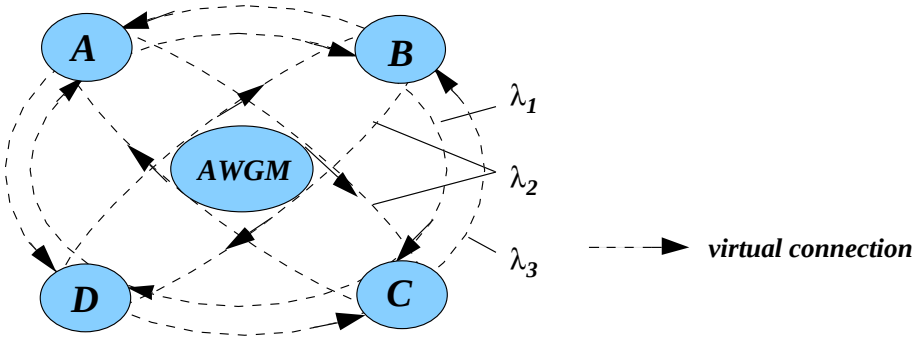

Figure 8.2.: Basic topology of a network made up by a 4x4 AWG. For better visibility only virtual connections are shown.

8.3. Node design

The node should be as simple as possible, but allow for the switching of (preferably IP size) packets. This means that the components used in the node should be capable of switching times in the order of a few ns. The basic functions a node in any non-broadcast network has to fulfill are:

- address recognition: the node should be able to recognize at least its own address in a packet.
- at least 2x2 switching (in and out)

These requirements led to a general node architecture in all-optical networks, that is widely agreed upon [Gre92]. Here, the processing of the header is separated from the payload. In figure 8.3 a fraction of the optical signal is extracted using an optical splitter, which should be enough to detect the header information. After that, the packet is delayed in a loop of standard SMF (single mode fiber) before entering a 2x2 optical switch. The decision about the way the packet has to go (receive it or leave it on the ring) should be made by the time the packet is leaving the delay line. The switch is then set to either *cross* or *bar* and the packet follows its destination. SOA switches like the one discussed in section 2.9.4 will be fast enough for this task. For WDM networks, there have to be wavelength demultiplexers and multiplexers before and after this switch architecture, respectively. It is possible to again use a single AWG do fulfill this task. One way to do this is shown in figure 8.4. As one can see, the incoming wavelengths are demultiplexed and then processed separately. Because of the symmetric nature of the AWG, the re-multiplexing can then be done from the other side while keeping the input and output numbering in contrast to the central AWG. This application of the AWG is similar to the one shown in [TIIN96]. For the reduction of inter- and intrachannel crosstalk it may however be desirable to use two separate devices for the demultiplexing and multiplexing of the wavelengths.

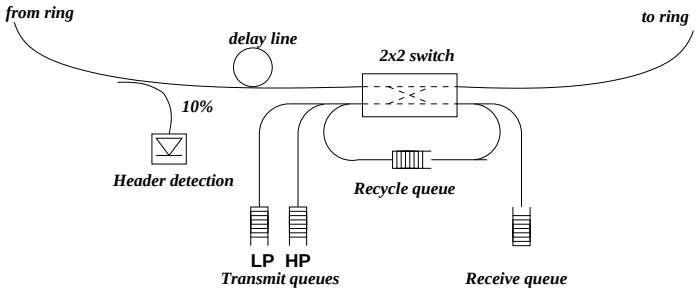

Figure 8.3.: Simplified nodal design for a single wavelength. The wavelength mux/demux is not shown here.

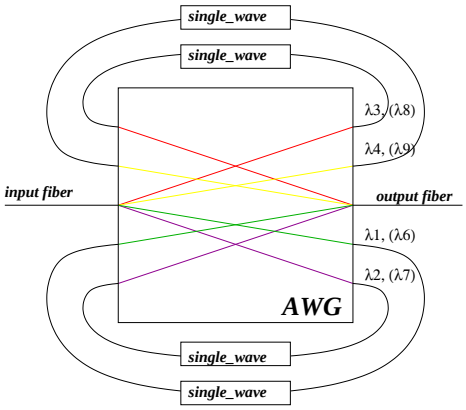

Figure 8.4.: Simplified nodal design using a 5x5 AWG as wavelength demux/mux. The small "single_wave" boxes have the design of figure 8.3.

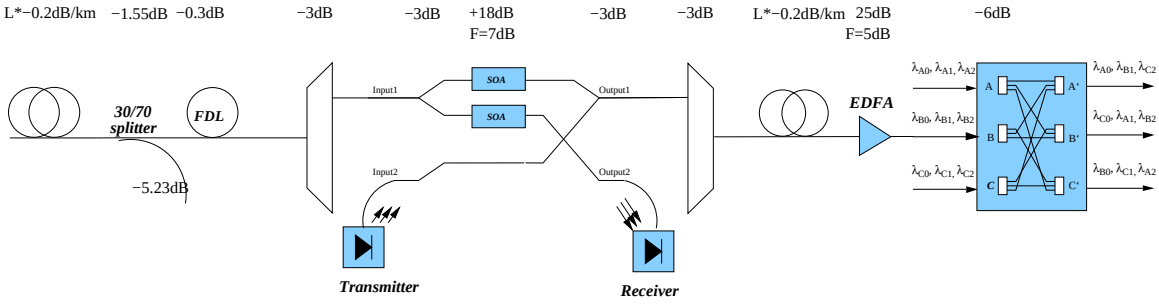

Figure 8.5.: Sketch of a complete transmission segment. The assumed gain and noise figures are printed above.

8.4. Feasibility aspects

To assess the general feasibility of such a network, a simplified calculation of the achievable bit error rate (BER) is performed next. We therefore neglect the influence of signal dispersion and the crosstalk within the Semiconductor Optical Amplifier (SOA) and the AWG. Thus, the result of the calculation can only be taken as a rough estimate for the number of hops that an optical packet can traverse without an electrical regeneration.
We start with a transmission segment shown in Fig 8.5. This figure shows one hop a data packet has to traverse. The signal is generated in the transmitter, which might be a Fabry-Perot or DFB (Distributed fiber Bragg) laser as mentioned in section 2.4. After that it enters a 3-dB combiner and the wavelength multiplexer, which is assumed to be a separate device here. A fiber of length L connects the node to the AWG and another one of the same length serves the opposite direction. Before entering the node, 30% of the optical power are separated in a so-called 30/70 splitter to detect the header information. After that, the signal enters the FDL and the wavelength demultiplexer. Finally, it enters a SOA where it is amplified or blocked. The lower SOA serves as a pre-amplifier for the detection of the signal in e.g. an PIN (Photo Diode, see section 2.8). To reduce the cost of the equipment, let us first consider the case without the EDFA shown before the AWG in the figure.

To calculate a BER that can be expected at the receiver, a parameter Q describing the quality of the signal is introduced.

$$BER = \frac{1}{2} erfc\left(\frac{Q}{\sqrt{2}}\right) \tag{8.2}$$

$$Q = \sqrt{SNR_{el}} = \frac{I_1 - I_0}{i_1 + i_0} \tag{8.3}$$

with I_1 and I_0 being the photo current of a logical 1 and 0, respectively, at the receiver and i_1 and i_0 the noise current. Assuming an OOK (On/Off Keying) with a modulation between the optical power $P_0 = 0$ and P_1, an optimal setting of the photocurrent threshold in the receiver and an electrical bandwidth of $B/2$ [Pet02], the above equation can be simplified

to:

$$Q \approx 2800 \cdot \sqrt{\frac{P_1}{mW}} \sqrt{\frac{Gbit/s}{B}} \frac{1}{\sqrt{F_{total}}} \tag{8.4}$$

with P_1 being the received optical power for a "1" bit, B being the data rate in the fiber and F_{total} the *noise figure* of the whole transmission distance. Simple receivers do not have an adjustable photocurrent threshold and use instead the average of the received powers P_0 and P_1 which leads to a value of $\frac{Q}{2}$ compared to eq. 8.4.
It is therefore necessary to calculate both P_1 and F_{total} next. A passive device like a fiber delay line or an optical splitter is usually characterized by its attenuation. In decreasing the power of a signal it however decreases the optical signal–to–noise ratio (OSNR). Therefore, the noise figure of a passive device is inverse to its gain:

$$F_{passive} = \frac{1}{G} \tag{8.5}$$

When N devices are cascaded in a *segment*, the noise figure F_{seg} and the gain G_{seg} calculate to:

$$F_{seg} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_2 - 1}{G_1 \cdot G_2} + \cdots + \frac{F_N - 1}{G_1 \cdot G_2 \cdot \cdots \cdot G_{N-1}}. \tag{8.6}$$

$$G_{seg} = G_1 \cdot G_2 \cdot \cdots \cdot G_N \tag{8.7}$$

Using equations 8.2 to 8.7 and the parameters listed in Table 8.1 the curves in Fig. 8.6 were computed. It shows the BER for a 2.5 Gbit/s transmission that can be expected at the receiver as a function of the distance between the node and the AWG.[3] It appears that the amplification of the SOA is not enough to compensate for the loss in the fiber over a reasonable distance. A second hop is only feasible for very short fiber lengths of few kilometers. In addition, it is interesting to see the different curves for the BER of the header and the data packet. In the second hop the BER of the header is several orders of magnitude higher than the packet BER. For the first hop, both curves are rather close to each other.
While this effect is only disturbing here, a careful system design could make use of it in two directions: A simple approach could give more power to the header detection branch of the asymmetric splitter before the FDL to guarantee a lower header BER. An error in the header would then indicate a high probability of an erroneous packet (which should be discarded, then).
The other way in which a forward error correcting code (FEC) would protect the header could be more elegant. The amount of processing to recover a broken header (with one or more bit errors) could then be taken as a measure for the quality of the transmission channel. This would allow for a *non–intrusive* monitoring of the data path.

[3]For simplicity we assume that all nodes in the network have the same distance to the AWG

Device	Gain (G)	Noise figure (F)
Standard SMF	-0.2 dB/km	
Fiber delay line	-1.18 or -0.3 dB (2.5 or 10 Gbit/s)	
30/70 optical splitter	-1.55 / -5.23 dB	
50/50 optical splitter	-3 / -3 dB	
wavelength mux/demux	-3dB	
AWG	-6 dB	
SOA	+18 dB	6 dB
EDFA	+25 dB	5 dB
DFB laser	1 mW (0 dBm output power)	
Photo diode	R=1 A/W (responsivity)	

Table 8.1.: Parameters used for the calculation of F_{total}. The noise figures for the passive devices are trivial.

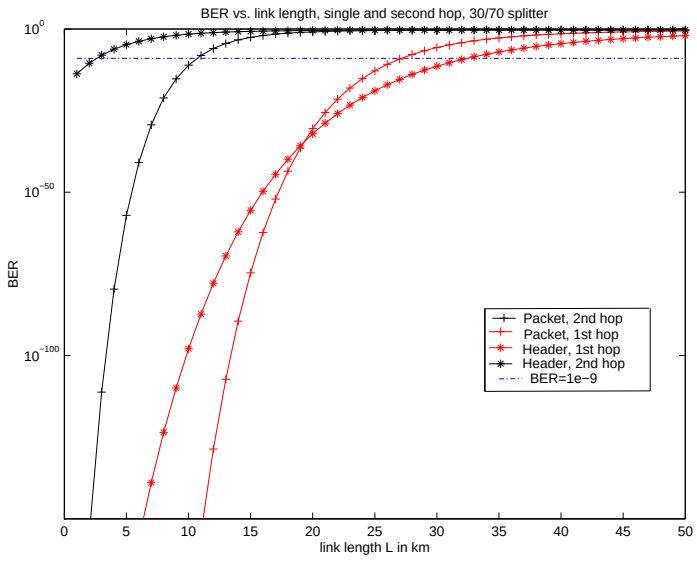

Figure 8.6.: BER vs. link length for the transmission segment without an EDFA. Only two hops seem possible, if at all.

8.4.1. Providing additional amplifiers

In the next step, the loss in the transmission segment is compensated by an Erbium-doped Fiber Amplifier (EDFA). A simple calculation gives a fiber length L of around 60 km, for which an off–the–shelf EDFA with a 25 dB gain would be able to achieve a total gain of 1, i.e. 0 dB attenuation per hop. Fig. 8.7 shows the resulting BER for the data rates of 2.5 and 10 Gbit/s, respectively. Both sets of curves (the header and packet BER curves are very close, now) show a remarkable increase in the number of possible hops in the network. For a 2.5 Gbit/s transmission, as many as 40 hops appear to be possible with a $BER = 10^{-9}$. Even with a high data rate of 10 Gbit/s, a reasonable network could be built, having a diameter of 10 hops or so.

8.5. Conclusions

In this chapter we proposed a new network architecture for multihop OPS networks. It is based on logical rings that may be set up on the wavelengths that are passively routed by an AWG. The transmission system has been modeled as a linear system concerning attenuation here. In addition, the thermal noise and the ASE noise (Amplified Spontaneous Emission) from the amplifiers have been considered. It has to be stated again that the considerations made in the previous section are very limited in their scope. Especially for data rates $\geq 10 Gbit/s$ nonlinear effects and dispersion problems (chromatic and polarization-mode) increase. No chromatic or polarization mode dispersion (PMD) is considered in the estimation made here. With standard SMF (showing a dispersion of $\frac{d\tau}{d\lambda} = 16 \frac{ps}{km \cdot nm}$) around 200km for 2.5Gbit/s and only 60 km for 10 Gbit/s is generally considered possible without dispersion compensation or regeneration. In longer transmission systems, dispersion compensating fiber (DCF) may be used. When doing so, dispersion is traded in for a higher attenuation. On the other hand, the parameter assumptions made in table 8.1 are rather conservative and allow for an additional penalty due to crosstalk and dispersion.
We can therefore conclude that it is indeed feasible to build such a network in a LAN or MAN size, however only if the attenuation of the fiber and the other passive elements in the network is compensated by additional amplifiers.

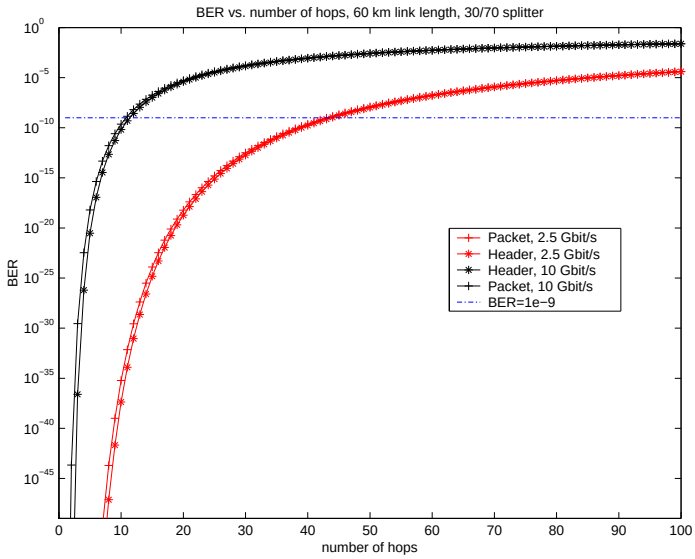

Figure 8.7.: BER vs. number of hops for a 60 km fiber length between the node and the AWG. For a transmission rate of 2.5 Gbit/s, 40 hops are possible with the BER still below 10^{-9}.

9. Performance analysis of the PrimeNet

In this chapter, we address the fundamental question whether single–hop or multihop AWG–based WDM networks provide a better performance for a given number of nodes and financial budget. For our comparison we consider a completely passive network which consists of a single AWG. Thus, the overall network costs are mainly caused by the structure of the nodes attached to the AWG. Note that not only the type of transceiver — tunable vs. fixed–tuned — and the number of transceivers used at each node determine the costs but also other aspects such as power consumption and management.
To compare both architectures we adhere to the terminology given in [GA96]. Let the total capacity of the net be the product of the nodal degree R (i.e. the number of transceivers per station), the number of nodes N and the data rate S of each transceiver divided by the average number of hops $\overline{h}$ between each station.

$$C = \frac{R \cdot S \cdot N}{\overline{h}} \tag{9.1}$$

9.1. Mean Hop Distance

As can be seen from the above equation the mean hop distance in the network is essential for the calculation of the total network capacity. Let one hop denote the distance between two logically adjacent nodes. The mean hop distance denotes the average value of the minimum numbers of hops a data packet has to make on its shortest way from a given source node to all remaining $(N-1)$ destination nodes. Note that in both single–hop and multihop networks the mean hop distance is the same for all (source) nodes. In the following we assume uniform traffic, i.e., a given source node sends a data packet to any of the resulting $(N-1)$ destination nodes with equal probability $1/(N-1)$.

9.1.1. Single–hop Network

Clearly, in the single–hop network each source node can reach any arbitrary destination node in one hop. Thus, the mean hop distance is given by

$$\overline{h}_S = 1. \tag{9.2}$$

9.1.2. Multihop Network

The capacity of the multihop network critically depends on the mean hop distance. Unfortunately we do not have the possibility to change the wavelength of a single packet in this

architecture. Instead, the source node decides according to the distance of the destination[1] which wavelength to use for transmission. In general form, the average hop distance can be represented as:

$$\bar{h} = \frac{1}{N-1} \sum_{i=1}^{N-1} dist_i \qquad \text{with} \tag{9.3}$$

$$dist_i = min(H(i, R_1), H(i, R_2), \ldots, H(i, R_r)) \tag{9.4}$$

$$H(i, R_i) = (i * R_i^{-1}) mod N \tag{9.5}$$

$$(R_i^{-1} * R_i) mod N = 1 \tag{9.6}$$

Each row of the matrix H in equation 9.4 contains the distance from a station 0 to any other station, indexed by the column. The inverse of r, r^{-1}, is used to calculate the distance. The values of r are all residue classes of N. So the calculation of r^{-1} yields another residue class. If we start with the computation of a matrix A of the form $A(i, j) = (i * j) mod N$ then the inverse of a given row $R_i = x$ is the row R_j, where $A(R_j, x) = 1$. Thus, the matrix H is calculated out of A by simply exchanging all rows r and r^{-1}. Note that if we know the inverse of r to be r^{-1}, then $(N - r)$ has the inverse of $(N - r^{-1})$.
For illustration, let us begin with the simple case where we use only one wavelength such that we obtain a unidirectional ring. The mean hop distance is then given by

$$\overline{h} = \frac{1}{N-1} \sum_{i=1}^{N-1} i = \frac{N(N-1)}{2(N-1)} = \frac{N}{2}. \tag{9.7}$$

The distance between an initial node and the other nodes is equal to $1, 2, \ldots, (N-1)$, respectively. That is, we walk around the ring. Next, let us deploy an additional wavelength. Adding another ring should decrease the mean hop distance as much as possible. To do so, the additional wavelength has to be chosen such that the second ring is counter–directional to the first one already in use. Consequently, for odd N we would walk $1, 2, \ldots, (N-1)/2$ hops in each direction. This case is illustrated next for $N = 11$ and wavelengths R_1 and R_{10}. The underlined figures in the matrix H (cf. eq. 9.4) represent the number of hops to each node. The lowest line is the distance vector of node 0 (and, due to the symmetry of the network, for every node).

[1] The decision could also depend on the load on this ring. This is not considered here.

node	1	2	3	4	5	6	7	8	9	10
λ										
1	1	2	3	4	5	6	7	8	9	10
2	6	1	7	2	8	3	9	4	10	5
3	4	8	1	5	9	2	6	10	3	7
4	3	6	9	1	4	7	10	2	5	8
5	9	7	5	3	1	10	8	6	4	2
6	2	4	6	8	10	1	3	5	7	9
7	8	5	2	10	7	4	1	9	6	3
8	7	3	10	6	2	9	5	1	8	4
9	5	10	4	9	3	8	2	7	1	6
10	10	9	8	7	6	5	4	3	2	1
$dist_i$	1	2	3	4	5	5	4	3	2	1

(9.8)

More generally, the resulting mean hop distance is given by

$$\overline{h} = \frac{1}{N-1} \frac{2(\frac{N-1}{2}+1)\frac{N-1}{2}}{2} = \frac{N+1}{4}. \tag{9.9}$$

The choice of the next wavelength(s) (= rings) to add depends on the resulting mean hop distance. The problem of the right choice seems to be NP–hard, although we can not prove this up to now. It seems to be of the family of knap–sack (or Rucksack) problems. Definitely, it is not the best idea to always go and look for counter–directional rings. For instance, in the case of $N = 13$ and $R_M = 4$, the combination of any two pairs of counter–directional rings like [1,4,9,12] leads to a mean hop distance of $\overline{h}_M = \frac{7}{3} = 2.3\overline{3}$ while $\overline{h}_M = \frac{9}{4} = 2.25$ for a combination of rings [1,4,6,11]. But a choice of the next ring to be counter–directional to the previous one seems to be a good heuristic and is in most cases near to the optimum value. Table 9.1 shows the mean hop distances of multihop networks for prime N up to $N = 17$. These were obtained through exhaustive search by taking the best (smallest mean hop distance) of all possible combinations of rings for a given value of R_M and N, respectively.

Since it was not possible to find a closed solution for $\overline{h}$, the next step was to find upper and lower bounds for it. For the generic calculation of the mean hop distance let the parameter $1 \leq R_M \leq (N-1)$ denote the number of simultaneously used wavelengths (transceivers) at each node. The mean hop distance $\overline{h}_M$ of the resulting multihop network is lower bounded by

No. of nodes N	3	5	7	11	13	17
No. of rings R_M						
1	1.5	2.5	3.5	5.5	6.5	8.5
2	1.0	1.5	2.0	3.0	3.5	4.5
3		1.25	1.67	2.5	2.92	3.75
4		1.0	1.33	2.0	2.25	3.0
5			1.17	1.5	1.75	2.37
6			1.0	1.4	1.5	1.94
7				1.3	1.42	1.75
8				1.2	1.33	1.5
9				1.1	1.25	1.44
10				1.0	1.17	1.37
11					1.08	1.31
12					1.0	1.25
13						1.18
14						1.12
15						1.06
16						1.0

Table 9.1.: Mean hop distances for optimum combinations of wavelengths in multihop networks

$$\overline{h}_M \geq \sum_{h=1}^{\left\lfloor \frac{N-1}{R_M} \right\rfloor} \frac{R_M}{N-1} \cdot h + \frac{(N-1)\mathrm{mod}R_M}{N-1} \cdot \left\lceil \frac{N-1}{R_M} \right\rceil \tag{9.10}$$

$$= \frac{1}{N-1} \left\{ R_M \frac{\left\lfloor \frac{N-1}{R_M} \right\rfloor \left(\left\lfloor \frac{N-1}{R_M} \right\rfloor + 1 \right)}{2} + [(N-1)\mathrm{mod}R_M] \left\lceil \frac{N-1}{R_M} \right\rceil \right\}. \tag{9.11}$$

To see this, note that the mean hop distance becomes minimum if (1) as many different nodes as possible are reached in each hop count starting with one hop and (2) the maximum hop distance (diameter) of the network is minimum. Applying this leads us to Eq. (9.10). Since a given source node sends on R_M wavelengths at most R_M different destination nodes can be reached for each hop count. Each time exactly R_M different destination nodes are reached up to a hop count of $\left\lfloor \frac{N-1}{R_M} \right\rfloor$, which corresponds to the first term of Eq. (9.10). The second term of Eq. (9.10) counts for the remaining nodes (less than R_M) which are $\left\lceil \frac{N-1}{R_M} \right\rceil$ hops away from the given source node.
Next, we compute the mean hop distance $\overline{h}_M$. For large N we assume a uniform distribution of the number of hops to every station over all rings. (Each station is reached only once

in a ring, and in a different hop number for every ring.) The probability of a certain hop count to be selected equals $1/(N-1)$. The probability of a certain hop number h to be the minimum of all selected rings R_M is the probability that the hop number is selected by one of the R_M rings times the probability that all the remaining $(R_M - 1)$ rings have a hop number between h and $(N-1)$. Of course, if the remaining area is smaller than the number of the remaining rings, the probability of this h to be the minimum is zero:

$$p(h_{min}) = \begin{cases} \frac{R_M\binom{(N-1-h)}{(R_M-1)}}{(N-1)\binom{(N-2)}{(R_M-1)}} & : \quad h <= N - R_M \\ 0 & : \quad h > N - R_M \end{cases} \tag{9.12}$$

The mean hop distance $\overline{h}_M$ is equal to the expected value of h_{min}:

$$\overline{h}_M = E[h_{min}] = \frac{1}{N-1} \sum_{N=1}^{N-1} \sum_{h=1}^{N-R_M} hp(h_{min}) \tag{9.13}$$

where the addition of the $\overline{h}$ over all stations can be omitted since we assume $\overline{h}$ to be the same for every station. Thus, we get

$$\begin{aligned} \overline{h}_M = E[h_{min}] &= \sum_{h=1}^{N-R_M} h \frac{R_M}{N-1} \frac{\binom{(N-1-h)}{(R_M-1)}}{\binom{(N-2)}{(R_M-1)}} && (9.14) \\ &= \frac{N^2\binom{N-1}{R_M}}{(R_M+1)(N-R_M)\binom{N}{R_M}} && (9.15) \\ &= \frac{N^2(N-1)!(N-R_M)!}{(N-1-R_M)!(R_M+1)(N-R_M)N!} && (9.16) \\ &= \frac{N^2(N-1)!(N-R_M)!}{(R_M+1)N(N-1)!(N-R_M)!} && (9.17) \end{aligned}$$

which surprisingly boils down to

$$\overline{h}_M = E[h_{min}] = \frac{N}{R_M+1}. \tag{9.18}$$

Equation (9.18) gives the mean hop distance for all possible combinations of R_M rings. Therefore, it is an upper bound for the mean hop distance of the *best choice* of the wavelengths.
Figure 9.1 depicts the lowest achievable mean hop distance as a function of R_M for $N = 17$. Apparently, increasing R_M, i.e., adding fixed–tuned transceivers to each node decreases the mean hop distance. The minimum mean hop distance equals one and is achieved for $R_M = N - 1 = 16$. Note that the lower bound of the mean hop distance is tight. For the other values of N presented in table 9.1 we observed that the upper bound is tight as well.

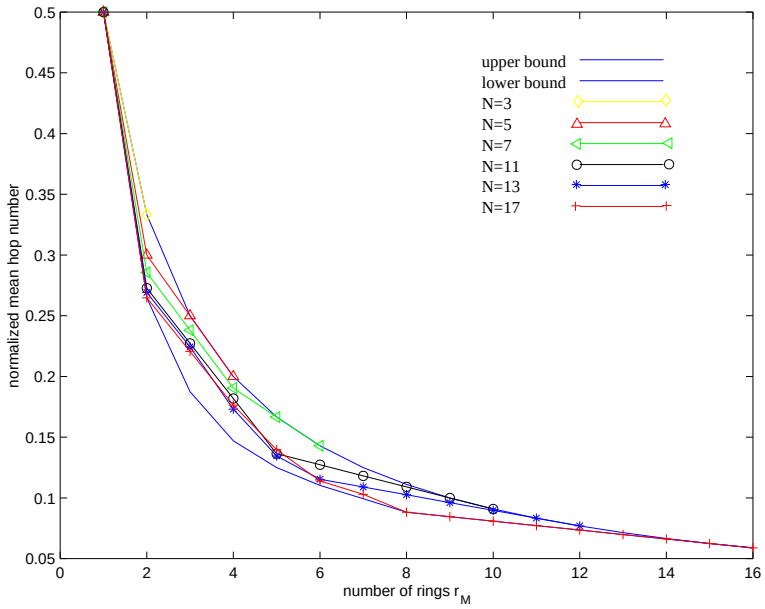

Figure 9.1.: Mean hop distance of multihop networks vs. R_M for $N = 3$ up to $N = 17$.

9.2. Performance Comparison

Beside the mean hop distance the single–hop and multihop networks are compared in terms of network capacity. According to [AS91], let the network capacity C be defined as

$$C = \frac{R_S \cdot S \cdot N}{\overline{h}} \tag{9.19}$$

where r denotes the number of transceivers per node, S stands for the transmitting rate of each transmitter, N represents the number of nodes in the network, and $\overline{h}$ denotes the mean hop distance of the network.

9.2.1. Single–hop Network

As mentioned in section 7.2.2, in the single–hop network each node is equipped with R_S tunable transceivers, where $R_S \geq 1$. We consider fixed–size packets and assume that the transceiver has to be tuned to another wavelength after transmitting a data packet (we thereby provide a conservative capacity evaluation). Due to the nonzero tuning time τ of the transceiver the effective transmitting rate is decreased as follows

$$S_S = \frac{L}{L+\tau} \cdot S \tag{9.20}$$

$$= \frac{1}{1+\tau_L} \cdot S \quad , \tau_L = \frac{\tau}{L} \tag{9.21}$$

where τ_L denotes the transceiver tuning time normalized by the packet transmission time L. With $\overline{h}_S = 1$ the capacity of the single–hop network equates to

$$C_S = \frac{R_S \cdot S \cdot N}{1+\tau_L}. \tag{9.22}$$

9.2.2. Multihop Network

In the multihop network R is equal to the number of used wavelengths (transceivers) R_M as explained in section 9.1.2, i.e., $r = R_M$ where $R_M = 1, 2, \ldots, (N-1)$. Since the transceivers are fixed–tuned there is no tuning penalty. Consequently, the effective transmitting rate equals S. Using the upper bound of the mean hop distance given in eq. 9.18 we get the lower bound of the capacity as follows

$$C_M \geq \frac{R_M \cdot S \cdot N}{\overline{h}_{M\,max}} = R_M \cdot (R_M + 1) \cdot S. \tag{9.23}$$

Similarly, using the lower limit of the mean hop distance given in eq.(9.11) conveys the upper bound of the capacity

$$C_M \leq \frac{R_M \cdot S \cdot N}{\overline{h}_{M_{min}}} \tag{9.24}$$

$$= \frac{R_M \cdot S \cdot N \cdot (N-1)}{R_M \frac{\left\lfloor \frac{N-1}{R_M} \right\rfloor \left(\left\lfloor \frac{N-1}{R_M} \right\rfloor + 1 \right)}{2} + [(N-1) \mathrm{mod} R_M] \left\lceil \frac{N-1}{R_M} \right\rceil} \tag{9.25}$$

Next, we want to calculate the proportion of fixed–tuned transceivers that has to be deployed to achieve the same network capacity as a single–hop network with a given number of nodes. Therefore, we start by equating the capacities from eqs. (9.23) and (9.22):

$$C_M = C_S \tag{9.26}$$

$$R_M \cdot (R_M + 1) \cdot S = \frac{R_S \cdot S \cdot N}{1+\tau_L} \tag{9.27}$$

$$R_M^2 + R_M - \frac{R_S \cdot N}{1+\tau_L} = 0 \tag{9.28}$$

$$R_M = -\frac{1}{2} \pm \sqrt{\frac{1}{4} + \frac{R_S \cdot N}{1+\tau_L}} \tag{9.29}$$

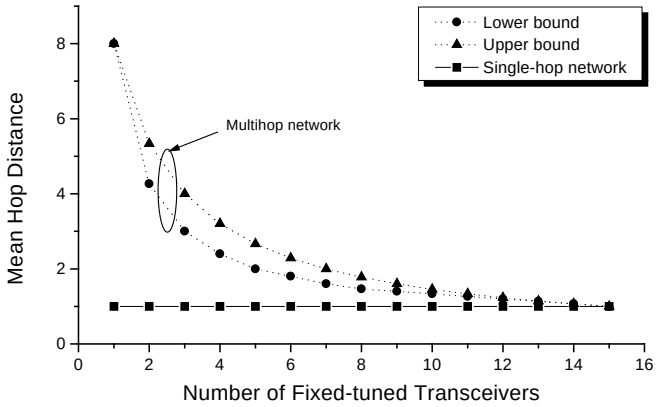

Figure 9.2.: Mean hop distance vs. R_M for $N = 16$.

Eq. (9.29) gives the number of fixed–tuned transceivers R_M per node in a multihop network whose capacity is equal to that of a single–hop network with R_S tunable transceivers at each node for a given population N.

9.3. Numerical Results

In all presented numerical results we consider fixed–size packets with a length of 1500 bytes and a transmitting rate of 10 Gbps. This translates into a packet transmission time equal to $L = 1.2$ μs. The channel spacing is assumed to be 100 GHz (0.8 nm at 1.55 μm). First, we consider a network with a population of $N = 16$ nodes. Figure 9.2 illustrates that the mean hop distance of the single–hop network is one, independent of R_M. We observe that both the upper and lower bound of the mean hop distance of the corresponding multihop network decrease exponentially with increasing R_M. As a consequence, a few fixed–tuned transceivers at each node are sufficient to decrease the mean hop distance of the multihop network dramatically and to get close to the mean hop distance of the single–hop network. Adding further transceivers has only a small impact on the resulting mean hop distance. For $R_M = N - 1 = 15$ both single–hop and multihop networks have the same mean hop distance, namely, one.

However, from the network capacity point of view equipping each node with as many fixed–tuned transceivers as possible is beneficial. This can be seen in figure 9.3 which depicts the network capacity (bounds) in Gbps as a function of r for both single–hop and multihop networks. While the network capacity of the single–hop network increases linearly the capacity of the multihop counterpart increases with more than the square of R_M. This is due to the fact that a large R_M not only decreases the mean hop distance but also increases

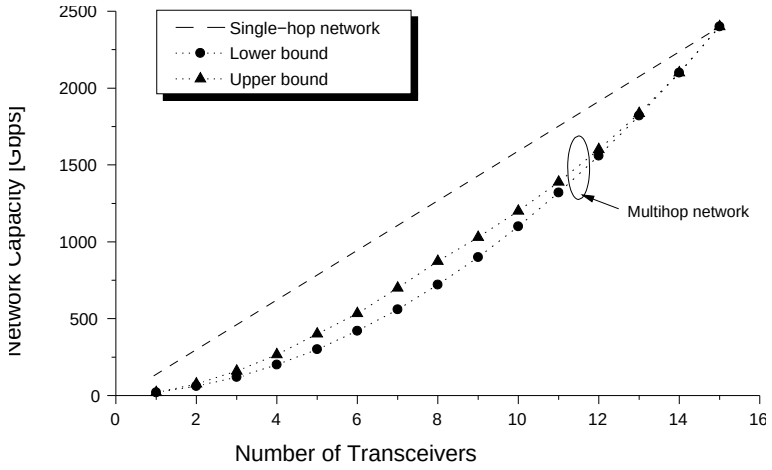

Figure 9.3.: Network capacity vs. R for $N = 16$.

the degree of concurrency by using all transceivers simultaneously. Note that the multihop network requires at least four fixed–tuned transceivers per node in order to outperform its single–hop counterpart with one single tunable transceiver per node in terms of capacity. For the illustration of this fact a dashed horizontal line is drawn in figure 9.3.

Recall from section 7.2 that for a given channel spacing the number of nodes N determines the required tuning range of the tunable transceivers used in the single–hop network. From table 2.10 in chapter 2 we learn that with a channel spacing of 0.8 nm we can deploy fast tunable electro–optical transceivers for up to $N = 16$ nodes, approximately. This translates into a negligible normalized tuning time $\tau_L = 8.3\bar{3} \cdot 10^{-3}$. In contrast, for $N > 16$ acousto–optic transceivers have to be applied which exhibit a three orders of magnitude times larger tuning time. Hence, we obtain a normalized tuning time $\tau_L = 8.3\bar{3}$. The impact of the transceiver tuning time on the network capacity is shown in fig. 9.4. For $N \leq 16$ the capacity of the single–hop network grows linearly with N.
For $N > 16$ acousto–optic transceivers have to be applied instead of electro–optic ones. The incurred larger tuning time dramatically decreases the network capacity. For increasing N the network capacity again grows linearly but the slope is smaller.
In addition, fig. 9.4 depicts the lower capacity bound of the multihop network. Interestingly, this bound remains constant for varying N. This is because with increasing N more nodes contribute to the network capacity but each node has to forward packets for a larger fraction of time due to the increased mean hop distance resulting in a lower netto data rate per node. Eq. (9.23) reflects this point more precisely; both the number of transmitting nodes and the mean hop distance are directly proportional to N such that the lower capacity bound is independent of N.

The dependency of R_M from N and R_S was calculated in eq. (9.29). It is shown in figure 9.5. The z–axis depicts the number of fixed–tuned transceivers R_M that must replace one tunable transceiver in order to achieve the same network capacity in both multihop and

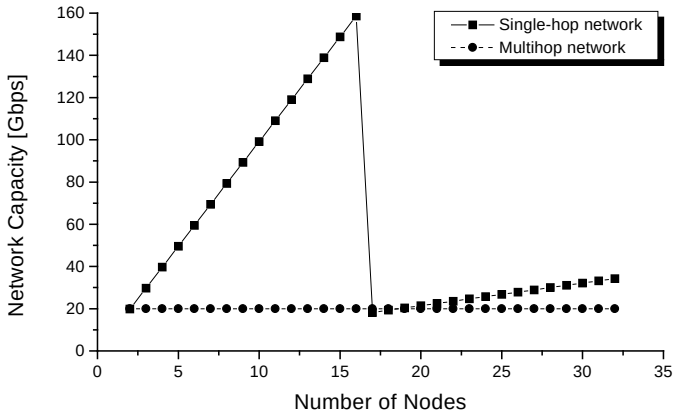

Figure 9.4.: Network capacity vs. N.

single–hop networks. Note that this graph can also be used to help decision makers design an appropriate AWG–based WDM network — either single–hop or multihop — for given population, capacity, and cost scenarios, as outlined in the concluding section 9.6.

9.4. Link Capacity, Access Delay and Throughput

Multihop networks in general offer a more flexible bandwidth assignment than single hop networks [Muk92a]. In the case of AWG-architectures this becomes even more critical, because there is only one direct wavelength between any pair of sender and receiver. The proposed PrimeNet architecture, in contrast, offers the possibility to use all available wavelengths for a certain flow between to nodes[2]. On the other hand, using a path other than the shortest will have a negative impact on other connections. Thus we try to estimate this impact in this section.

The following statements are made under the assumption of a Poisson distributed arrival process of λ packets per second of an exponentially distributed length with mean $1/\mu$ bits. Every link is assumed to be equally loaded. These assumptions are needed to model each link in the networks as a M/M/1 queue. The loading on a link L_i is defined as the number of flows that use this link to communicate. When using only one path – the shortest – there are $N(N-1)$ possible flows. Every flow uses in average $\bar{h}$ links. When we divide this by the total number of links in the network $R \cdot N$, the mean number of flows $\bar{L}$ depends on the

[2]The term flow is used here to describe the flow of packets from source to destination.

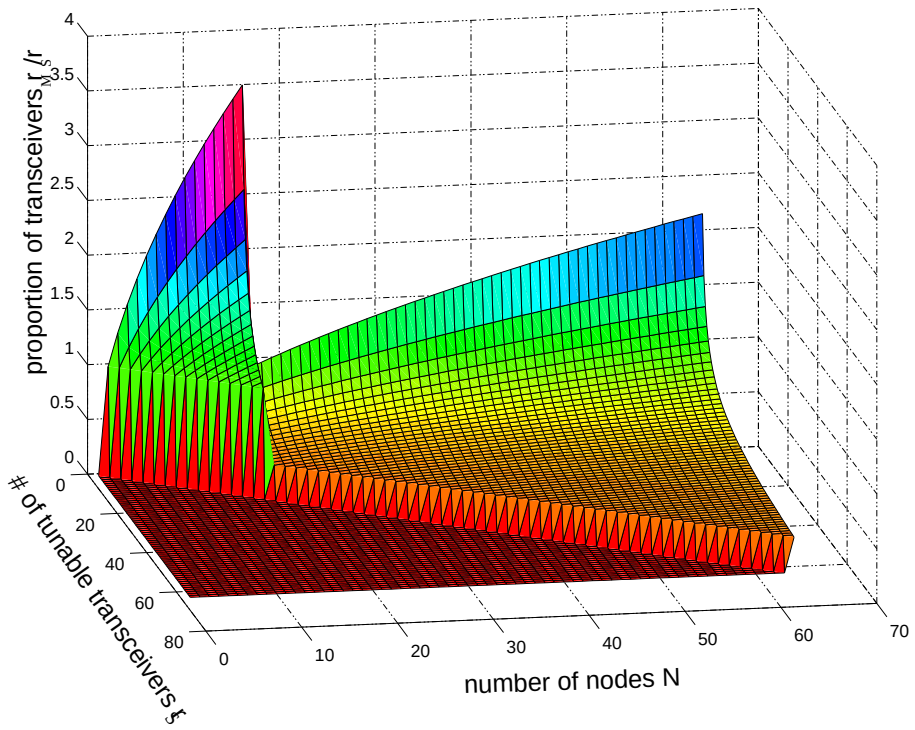

Figure 9.5.: Proportion R_M/R_S of fixed–tuned to tunable transceivers that is needed to achieve the same network capacity in a single–hop network with R_S tunable transceivers and in a multihop network with R_M fixed–tuned transceivers per node.

mean hop distance in the following way:

$$\bar{L} = \frac{\bar{h}N(N-1)}{R \cdot N} = \frac{N(N-1)}{R(R+1)} \tag{9.30}$$

According to [SR94] this loading determines the average queuing delay the following way:

$$f(L_i) = \frac{L_i}{\mu S - \lambda \cdot L_i} = \frac{1}{\frac{\mu S}{L_i} - \lambda} \tag{9.31}$$

This equation is true for a fixed assignment of the bandwidth of one link to each flow. When a variable assignment is assumed, the access delay is lower:

$$f(L_i) = \frac{1}{\mu S - L_i \cdot \lambda} \tag{9.32}$$

Introducing the normalized access delay for the shortest–path routing $d_{SP} = d\mu S$ that is computed using a normalized offered load per flow $\lambda_N = \frac{\lambda}{\mu S}$ we get:

$$d_{SP} = \frac{1}{1 - \lambda_N \cdot \frac{N(N-1)}{R(R+1)}} \tag{9.33}$$

Note that this access delay is given in the number of packets. The throughput per flow is the largest value for λ_N for that the above equation has a finite solution, that is: $\lambda_N < \frac{R(R+1)}{N(N-1)}$. This value also corresponds to the solution of eq. 9.23 when multiplying it with the number of flows $(N-1)$ that every station transmits.

In the single hop network, the throughput per station is derived from eq. (9.22), and in the next step we calculate the number of fixed transceivers that is needed to achieve the same throughput for a given R_M and τ_L:

$$\frac{R_S}{(N-1)(1+\tau_L)} = \frac{R_M(R_M+1)}{N} \tag{9.34}$$

$$R_S = \frac{R_M(R_M+1)(1+\tau_L)}{N} \tag{9.35}$$

9.4.1. Using multiple paths in parallel

Using only the shortest path between any two stations would exclude the main advantage of the architecture, namely the concurrent use of up to R rings (=wavelengths). When we want to make use of the other wavelengths, too, we have to define a strategy to do so. The simplest although probably not the most effective is the parallel transmission of an equal share of packets belonging to one flow over all R rings. Using this approach, the mean loading on the link is similar to eq. (9.30), but with a mean hop distance of $\frac{N}{2}$. It therefore

increases to:

$$L_i = \frac{N(N-1) \cdot R \cdot N}{R \cdot N \cdot 2} = \frac{N(N-1)}{2} \tag{9.36}$$

This is because the number of flows is to be multiplied by the number of outgoing links per station now. On the other hand, since every flow is divided over R links, the arrival rate decreases accordingly. The access delay per link is then:

$$d_{par} = \frac{1}{1 - \frac{\lambda_N}{R} \cdot \frac{N(N-1)}{2}} \tag{9.37}$$

As it can be seen from the above equation, the throughput per station is now $\lambda_N < \frac{2 \cdot R}{N(N-1)}$. When comparing this result to the throughput per station that can be achieved using the shortest–path routing we see that both are equal for $R = 1$. For every other value of R the achievable throughput is less for the parallel transmission. While this result seems to be discouraging at first, it is possible to find situations where it pays to parallelize the flows.
As mentioned at the beginning of the link capacity considerations, we assumed an equal load for every flow up to now. To see how the network behaves under unequal load, it is necessary to subdivide the traffic that is offered to a certain link into a λ_{own} and a λ_{others}. The equations 9.33 and 9.37 can now be rewritten as follows:

$$d_{SP} = \frac{1}{1 - \lambda_{others} \cdot \left(\frac{N(N-1)}{R(R+1)} - 1\right) - \lambda_{own}} \tag{9.38}$$

$$d_{par} = \frac{1}{1 - \frac{\lambda_{others}}{R} \cdot \left(\frac{N(N-1)}{2} - 1\right) - \frac{\lambda_{own}}{R}} \tag{9.39}$$

In the next step the maximum throughput for a certain flow is calculated depending on the offered load of the other flows:

$$\lambda_{ownSP} = 1 - \lambda_{others} \cdot \left(\frac{N(N-1)}{R(R+1)} - 1\right) \tag{9.40}$$

$$\lambda_{ownpar} = R - \lambda_{others} \cdot \left(\frac{N(N-1)}{2} - 1\right) \tag{9.41}$$

It can be seen here that may really be beneficial to go parallel as long as the the overall load is low. Setting λ_{others} to zero results in a potential throughput of R for the parallel case as compared to 1 for the sequential transmission over the shortest path. In the last step the above equations are transformed to lead to an expression for λ_{others} that is the point of intersection of both throughput curves as illustrated in figure 9.4.1:

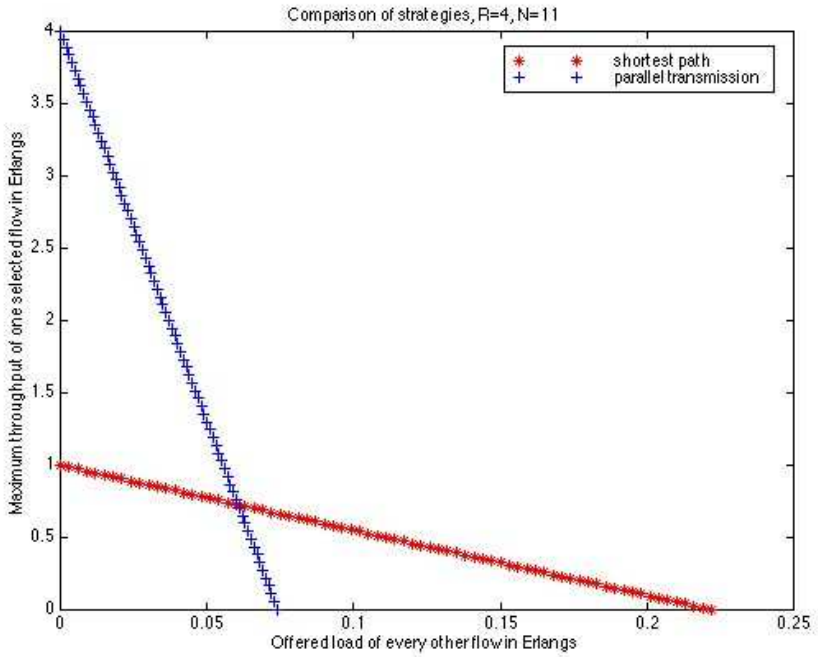

Figure 9.6.: Maximum throughput of a single flow vs. offered load of all other flows. Number of rings is R=4, N=11 node network.

$$\lambda_{ownSP} = \lambda_{ownpar} \tag{9.42}$$

$$1 - \lambda_{others} \cdot \left(\frac{N(N-1)}{R(R+1)} - 1 \right) = R - \lambda_{others} \cdot \left(\frac{N(N-1)}{2} - 1 \right) \tag{9.43}$$

$$\lambda_{others} = \frac{R-1}{\frac{N(N-1)}{2} - \frac{N(N-1)}{R(R+1)}} \tag{9.44}$$

$$\lambda_{others} = \frac{2 \cdot R \cdot (R^2-1)}{(R^2+R-2) \cdot (N(N-1))} \tag{9.45}$$

9.5. Comparison of the PrimeNet to other multihop architectures

To make the proposed architecture comparable to others that were introduced in chapter 7.3.1 it is necessary to change the paradigm of the all-optical node architecture. Most of the proposed multihop networks assume a full connectivity between all input and output ports of one node. In other words, when applying this to the world of WDM networks, full wavelength conversion is employed in each node.

This – rather expensive – assumption however leads to much lower mean hop distances in the other multihop architectures. So, to be competitive, we will derive the mean hop distance for a PrimeNet with full wavelength conversion next. We start with the common formula for the mean hop distance as the expected value of the number of hops:

$$\overline{h} \approx E(h) = \frac{1}{N-1} \sum_{i=1}^{N-1} i \cdot p_i \tag{9.46}$$

Now the value of p_i is of interest. Clearly, in the first step, $p_1 = \frac{R}{N-1}$. Since all rings have a different hop distance, R nodes can be reached from a given node N_0. Next we consider the problem of commutativity between the steps that are needed to reach a certain node. For example, let us consider two rings with hop distance a and b. Starting from a root node N_0, one can build a tree of all the nodes reachable from there. We can denote the nodes that have been reached in a certain step by a word that consists of i characters for step i. In the first step we thus build nodes a and b, followed by nodes aa, $ab = ba$, and bb in the second step. In a general form this is a combination of $R + i - 1$ elements taken i at a time and we can create $\binom{R+i-1}{i}$ nodes in every step. These nodes, however, are not necessarily distinct from nodes that were reached in the previous steps. On the other hand it is clear that $aa \neq a$ (therefore a had to be zero) and $aa \neq N_0$ (this is only valid for $N = 2$ and generally for $i = N$, which is not in the scope of the summation). A similar statement can be made for bb and for ab, which are all chosen out of $(N - 2)$ nodes. If we now keep in mind that the nodes that are generated in each step are distinct from each other, we can apply a hyper-geometric distribution to calculate the probability of the number of newly generated nodes in each step[3]:

$$p(X = x) = \frac{\binom{M}{x}\binom{N-M}{n-x}}{\binom{N}{n}} \tag{9.47}$$

with x being the number of events we are interested in (here: the number of new nodes), M being the number of potentially positive events (nodes not yet reached) in N, which is the total number of elements we are drawing n from. To justify the choice of the hyper-geometric distribution here, we assume again an equal distribution of nodes over all rings, just as we did in order to calculate eq. 9.18.

[3]The hyper-geometric distribution is often referred to as *drawing balls of two colors from an urn without putting them back.*

The expected value of this distribution is

$$E(H) = n \cdot \frac{M}{N} \tag{9.48}$$

$$= \binom{R+i-1}{i} \frac{M}{N-i} \tag{9.49}$$

$$E(0) = \binom{R-1}{0} \frac{N}{N} = 1 \tag{9.50}$$

$$E(1) = \binom{R}{1} \frac{N-1}{N-1} = R \tag{9.51}$$

$$E(i) = \binom{R+i-1}{i} \frac{N - E(0) - E(1) - \cdots - E(i-1)}{N-i} \tag{9.52}$$

$$\text{for} \quad N - E(0) - E(1) - \cdots - E(i-1) >= 0 \tag{9.53}$$

This calculation is not very exact for values of R being near to $N-1$, because here M may be exhausted while drawing from it. Thus, equation 9.48 over-estimates $E(i)$ for large R. Because of that,

$$E(i-1) = N - E(1) - \cdots - E(i-2)$$

whenever the whole difference (eq: 9.53) drops below 0. By the way, $D = (i-1)$ is the diameter of the graph. The probability of reaching a node in step i, p_i is of course the expected number of nodes that are i hops away from N_0 divided by $(N-1)$:

$$p_i = \frac{E(i)}{N-1} \tag{9.54}$$

9.5.1. Multi-connected rings

The *wheel* that was proposed in 1996 [GA96] has a very similar structure to a PrimeNet with wavelength conversion. It is described in section 7.4.2. Unfortunately, the calculation of the mean hop distance in a *wheel* that was carried out in the original paper had two little errors that made it necessary to change the formula for the E_{avg}. The first was the division by N instead of $(N-1)$ for the mean hop distance. The second is similar to the above mentioned over-estimation of M and leads to $E_{avg} > 1$ for $R = N-1$, which is also not useful. Although we were not able to derive a closed solution for the second problem, the first can be fixed by:

$$E_{avg} = \frac{N}{2} \cdot \frac{R \cdot (\sqrt[R]{N} - 1)}{N-1}.$$

This produces a correct calculation for the case of $R = 1$, namely $N/2$. A comparison of the mean hop distance of both networks is shown in figure 9.7. It can be seen there that for low R the curve for the *wheel* is between the lower and upper bound for the PrimeNet. This is no surprise since both networks are employing the same interconnection patterns. The main difference between both networks is the number of wavelengths that is needed to support R rings or *skip vectors*, respectively. It is $\frac{R(R+1)}{2}$ for the *wheel* compared to R

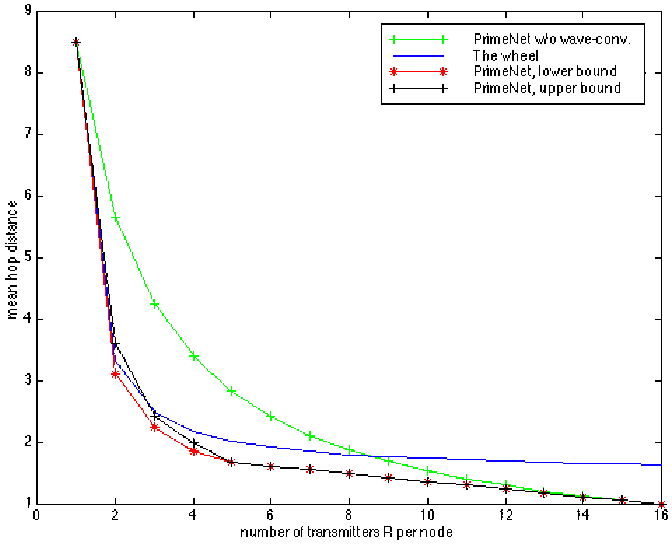

Figure 9.7.: Mean number of hops for a N=17 node network. The upper line represents the configuration without wavelength conversion.

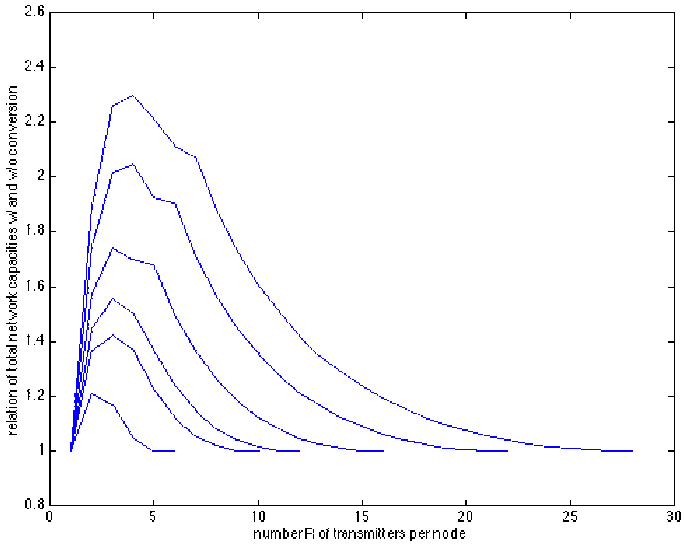

Figure 9.8.: Relation of the total network capacities between nodes with wavelength conversion and without wavelength conversion. Curves for prime numbers between5 and 29 are plotted here.

for the PrimeNet. In addition, the number of *fiber hops* (number of nodes that a packet traverses purely optical) is $\sqrt[r]{N^{r-1}}$ in maximum for the *wheel* compared to 1 in PrimeNet. These properties make the *wheel* much less scalable in the number of *scip vectors* and the number of nodes than the PrimeNet.

9.5.2. Other multihop architectures

A comparison of other multihop architectures like ShuffleNet, deBruijn graph or Kautz graph to the PrimeNet is found to be not really useful, because the main objective of PrimeNet was the establishment of rings to ease the medium access. Although all of the mentioned multihop architectures form cycles that connect subsets of their nodes, these cycles are not used as rings in the sense of a MAC protocol. On the other hand, the configuration of rings on the AWG has a huge potential for wavelength reuse. As an example, consider the (2,2)-ShuffleNet that was shown in figure 7.6. It consists of 4 rings of two nodes and 2 rings of four nodes. The four bidirectional connections could be realized using only 1 wavelength, while the other 2 rings need at most 4 wavelengths, which makes up a total of 5 compared to 16 in the case of a passive star coupler. It is likely that one could reduce the number of wavelengths further by applying a more clever algorithm than this greedy one.

9.6. Conclusion

WDM networks based on an AWG are either single–hop or multihop networks. In the single–hop network each node has (at least) one tunable transceiver. Equipping each node with one or more fixed–tuned transceivers results in multihop networks that consist of multiple virtual rings. We have compared both single–hop and multihop networks in terms of mean hop distance and aggregate capacity. Moreover, we have addressed the question which network type provides a higher capacity for a given number of nodes. The answer to this question largely depends on the cost ratio of fixed–tuned and tunable transceivers and on the tuning latency of the tunable transceivers. Fig. 9.5 can be interpreted such that it shows the relation of the cost of a tunable transceiver *unit* to the cost of a fixed transceiver *unit*. A *unit* comprises not only the respective transceiver but all additional items that are required for proper operation. Generally, these include initial, operation, management, and maintenance items. Fig. 9.5 illustrates that for interconnecting $N = 16$ nodes in a multihop architecture around 3.51 (that is, at least four) fixed transceiver units should replace one tunable transmitter (this can also be observed in Fig. 9.3). If, however, the desired capacity should be four times as large, requiring 4 tunable transceivers, the relation reduces to around 1.8 , meaning that 8 fixed transceivers would be sufficient per node. It can be seen from this that not just the price of the components should influence the choice of the network architecture, but also the number of nodes and the desired network capacity.
Clearly, many more factors add to the price of the single transmitter and receiver component. For instance, wavelength multiplexers and demultiplexers are needed before and after fixed transceivers, respectively. Optical splitters and combiners would be used in the same

place when applying a single–hop architecture. In addition, we note that the cost comparison of AWG–based single–hop vs. multihop networks has also to take the power budget into account. Since in the multihop networks packets have to pass the AWG multiple times optical amplifiers (EDFAs) might be mandatory. The resulting costs had to be added to the costs of the fixed–tuned transceiver.
It should be noted, however, that the figures provided here are independent of the actual cost of the transmitter *units*, but instead should serve as a general guideline for the choice of the proper network architecture.
After the comparison of the general architectures we assessed the benefit of a parallel transmission of packets belonging to one flow. From eq. 9.42 it is clear that this benefit will be larger for high R and low N. This result is straightforward as it means that for an increasing parallelism in the network it is better to use parallel transmission. The figure 9.4.1 shows the situation for $N = 11$ and $R = 4$. In this example for an offered load of about 6% per flow both strategies perform equally well. For a lower offered load λ_{others} a flow can take advantage of all R rings. Although the load for a single flow seems to be very low here keep in mind that 109 flows of this load sum up to a total network load of around 6.6! This is about 30% of the total network capacity ($R(R + 1) = 20$) that can be achieved using the shortest–path routing. Thus, there is a huge potential benefit in a load–adaptive choice of the routing strategies.

10. PrimeNet MAC protocol

10.1. Options for header transmission

To cope with the speed difference between optical transmission and electronic processing, the bit rate of the header should be much smaller than the rate of data. This approach has been widely used in other networks, for instance in IEEE 802.11 [Soc97], HIPERLAN-I[(RE96] and KEOPS [ea98]. All of these use low bit-rate headers that are transmitted (and arriving) just before the actual data packet.
Other approaches transmit the header a certain well-defined time before the start of the data transmission. The burst switching by Turner [Tur99] and Qiao [QY99] is an example of this approach. Please see section 5.7 for details on OBS. In principle this technique could work with any delay in the forwarding decision, but the number of hops in the network has to be known a priori. Also, dynamic routing of bursts is impossible.
When the delay for the forwarding decision is constant for every node (or if there exists an upper bound for it) then it is possible to delay each packet by the time needed to do the lookup of the destination port. This delay mainly consists of three elements: the time to receive the header, possibly compute a header checksum and extract the destination address ($t1$ in figure 10.1), the forwarding decision time ($t2$, in most cases this will be a table lookup) and the actual switching time ($t3$).

The time $T2$ determines the maximum packet length, while the interframe space should be at least $t3$. The total length of the DL should be $t1+t2+t3$. Smaller packets have to be padded up to the length of $t1+t2$. It is open which of the three parts will be the largest. There is optimization potential in all of them.
The actual encoding of the packet may be done in different ways. Three promising candidate technologies are introduced in the following.

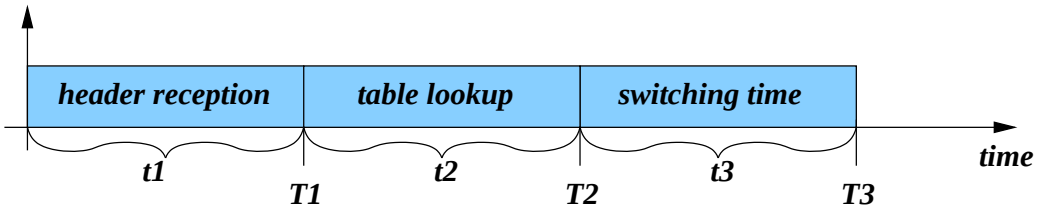

Figure 10.1.: Schematic of the components of the delay that makes up the delayline.

10.1.1. Direct sequence spreading

Gladisch et al. proposed in [GGH+98] a direct sequence modulation of OAM (operation and management) signals onto the transmitted data signal. This modulation was done with a 1024 chip Gold (pseudo-random) sequence resulting in an actual data rate of 9.6 kbit/s for the control signal. This data rate was limited by the simplicity of the experimental setup rather than technologically, according to the authors. Therefore it could be possible increase it by one or two orders of magnitude.
Assuming a data rate for the header of 1 Mbit/s, for instance, on a 2.4 Gbit/s data rate a 4 octet header (like the MPLS shim header in fig. 5.5) would be $t1 = 32\mu s$ long, resulting in a minimum packet size of around 10000 byte[1]. It should be possible to reduce the length of the chip sequence and thus, the header, but this would increase the amount of power that would have to be drawn out by the splitter.
Another problem is that of *label swapping*. It should be possible to change certain header information at each node. Assuming classical IP packet forwarding, at least the TTL (time to live) field must be decremented. With direct sequence modulation of the header information onto the payload it is hardly possible to erase the header from the payload. One possible solution here could be to use different (possibly orthogonal) codes at each neighboring node in the network. Still, adding more and more headers to the payload is likely to increase the interference between the headers at least, but would possibly even lead to a worsened optical payload signal.

10.1.2. Subcarrier modulation

The IP ring network HORNET that has been developed at UC Stanford employs a different but similar kind of header transmission. Here, as it is described in chapter 6.4.3 and [SSW+00], the header information is modulated on a subcarrier that has to be extracted at every station.

10.1.3. Exploitation of AWG periodicity

A totally different way of transmitting the header could be realized through the AWG. As shown in section 2.11, wavelengths in the next FSR (free spectral range) are routed to the same output as their corresponding wavelengths in the first FSR. This opens up the opportunity to transmit the header on a different wavelength that is going exactly the same way through the network as the data packet. It can be extracted in the node using fixed optical filters.
Besides the technological advantages of such an approach there is another attractive feature of going this way: The delay line is not that long anymore! It only has to be of length $t1+t2$. The time $t3$, denoting the optical switching duration, can be an offset that the header is sent before the payload. Figure 10.2 shows how the transmission of the header for the next packet overlaps the transmission of the previous data packet. An architecture like

[1]Note that this is about the size of a Jumbo packet!

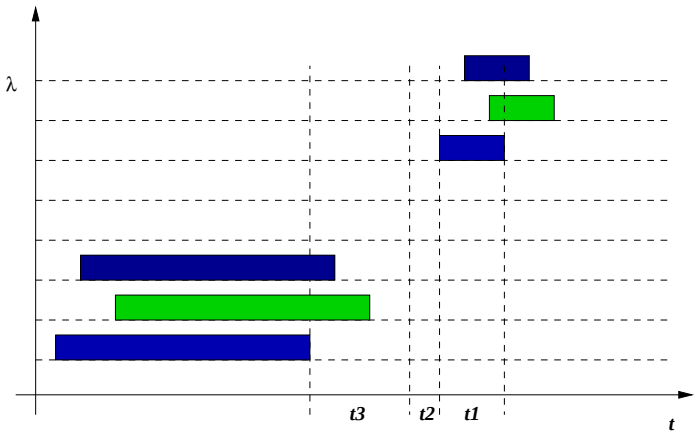

Figure 10.2.: Transmission of the header and payload in different FSRs.

this would lead to the termination (optoelectrical conversion) of the header transmission in each node, which should not be a problem at these bit rates. Thus, the payload could travel totally untouched through the network, while the header would be rewritten in each node. This, on the other hand, gives the opportunity to assign IP flows to certain FECs (Forwarding equivalence classes) and do a classical MPLS here.

10.2. Access Protocol

The given node design determines a certain behaviour of the node in a ring. A node may send a packet whenever it can be sure that this packet would not lead to a collision on the ring. This is the case when the FDL is empty or when the arriving packet is to be stripped (taken off the ring) by that node. The forwarding decision can be made only after having received the whole header (after $t1$). While this is true for every ring network, a lot of degrees of freedom remain in the choice of a MAC protocol. We will discuss some of them in the following.

Destination vs. source stripping Destination stripping is an option for ring networks that has good arguments in favor and against. The first ring networks like TokenRing and FDDI did source stripping, i.e. the sender of a packet removed it after one rotation. The destination had to make a copy of each packet and decide some time after whether to actually receive (send to upper layers) the packet. This allowed for loose time constraints for the evaluation of the packet's destination address. In addition, it was not necessary for the source to match the source address of each packet to its own, but instead it could just count the bytes and start removing the packet after exactly one round trip time. Another advantage of source stripping is the (implicit) acknowledgment that the source receives with its own packet returning to it. On the other hand, this kind of an immediate ACK for each

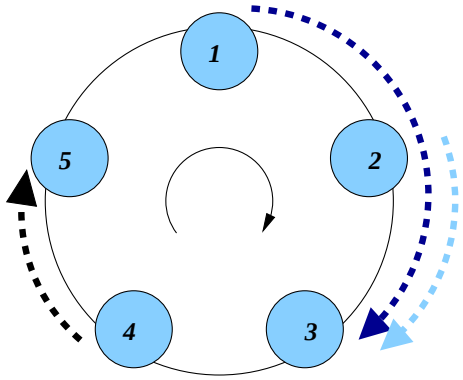

Figure 10.3.: Local vs. global fairness. A transmission between nodes 4 and 5 does not influence the other nodes and hence, should not be blocked.

packet is only needed for highly error prone media or collision based MAC protocols. For a collision-free fiber ring like the one proposed here the need of a MAC-level ACK is arguable at least.
The main reason to do destination stripping is the increase in capacity. In average half of the ring remains unused for the transmission of a certain packet and may therefore be reused for other transmissions. So, for a single ring, the mean hop distance drops to $N/2$ from N, resulting in roughly double the capacity compared to a source stripping ring. For a ring network like PRIMENET, it is absolutely necessary to use destination stripping, because the spatial reuse of wavelengths that is the main feature of an AWG has to be accompanied by a MAC protocol that allows for a spatial reuse of the rings. An analysis of the mean hop distance and the total network capacity follows in chapter 9.

Global vs. local fairness algorithms There are a number of problems that arise out of destination stripping. First there are the above-mentioned time constraints (since one has to be much faster in reading the packet's addresses and making the forwarding decision). Second, the problem of local fairness appears. It is illustrated in figure 10.3. Here two transmissions take place on the link between nodes 2 and 3. Both nodes 1 and 2 should therefore get 50% of the available bandwidth, given they are of the same service class. A third transmission from node 4 to 5 does not affect the other nodes and should therefore get access to the full bandwidth. In a network employing source stripping, however, all three transmissions would share the bandwidth and each should get one third, leading to *global fairness*. Because this is not necessary here, the notion of *local fairness* was introduced. It could also be called link based fairness and was considered in the development of SRP (see section 6.4.1). The basic idea is that a node may take more than its "fair share" of the bandwidth, as long as it does not prevent the other nodes from getting their "fair share".
The easiest way of guaranteeing a fair access to the medium would be a central controller

that has full knowledge of the load of each node in the ring. This full knowledge, however, is hard to get and outdated by definition when it arrives at the controller node. In addition, this strategy requires a reservation phase prior to each (larger) transmission. The classical fairness mechanisms for *source stripping* ring networks like Token Ring and FDDI were therefore token-based, which means that only the station(s) that are currently in possession of the token are allowed to transmit. By the use of timers that limited the token holding time or the number of packets that a node is allowed to transmit it was easy to guarantee a global fairness. Fairness mechanisms for advanced packet rings like CRMA-II and MetaRing use cyclic reservations of slots or a cyclic update of transmission credits. These mechanisms are all inherently global although they can be implemented in a distributed way.
The next problem arises out of the topology: MetaRing and SRP were designed for *bi-directional destination stripping* rings. It is shown in [Set98] that any node would only send halfway around each ring in maximum (assuming a shortest path routing) and thus, does not need to care about any transmission that is going on on the other half. Therefore it is possible to introduce a so-called *semi-global* fairness. This results in the cycle length being only half of the previous time and thus, in a reduced access delay and improved performance.
In the case of PrimeNet, the above statement is even more critical. The more rings there are in the network the less useful is a global fairness algorithm because of the decreasing mean number of hops for every packet in the network. It is even less useful to have a central controlling node that collects reservations and issues permits. Thus, we have to look for a local fairness algorithm that works in a distributed way. SRP, like CRMA-II and MetaRing, performs a cyclic update of its transmission credits. But, in contrast to the other protocols this update is done in the node itself using a timer function that periodically adds tokens to a bucket. This way, a node does not have to wait for a reservation period or a SAT packet to start transmitting. Of course, a backpressure mechanism has to be used to adapt the rate at which tokens are generated to the load in the part of the ring that is influenced by the transmission. The dimensioning of this algorithm concerning the achievable degree of fairness and speed of adaptation is not trivial and will be dealt with in chapter A.

Fixed vs. variable packet size This question has often been discussed in the literature and is of interest for network architectures in general. For Gbit/s LANs, the HOL-problem (head-of-line blocking) is not that critical anymore as it was for the classical Ethernet using 10 Mbit/s. In the discussion about the use of Jumbo frames that was mentioned in section 6.1 the relation of packet size to transmission speed is frequently used to show that the absolute length (in time) of a packet may be increased by several orders of magnitude without blocking time-sensitive applications. Following this line of discussion, it should no be a problem to set the maximum transmission unit to a value that corresponds to a Jumbo packet.
The next question is: should this packet size be fixed? Technological factors like the speed of optical switches require a minimum packet size that is considerably above the usual 40 or 64 byte. For example, at a data rate of 10 Gbit/s, the transmission time of a Jumbo packet (recall the discussion in section 6.1) of 9216 byte is 7.4 μs. To keep the guard time between

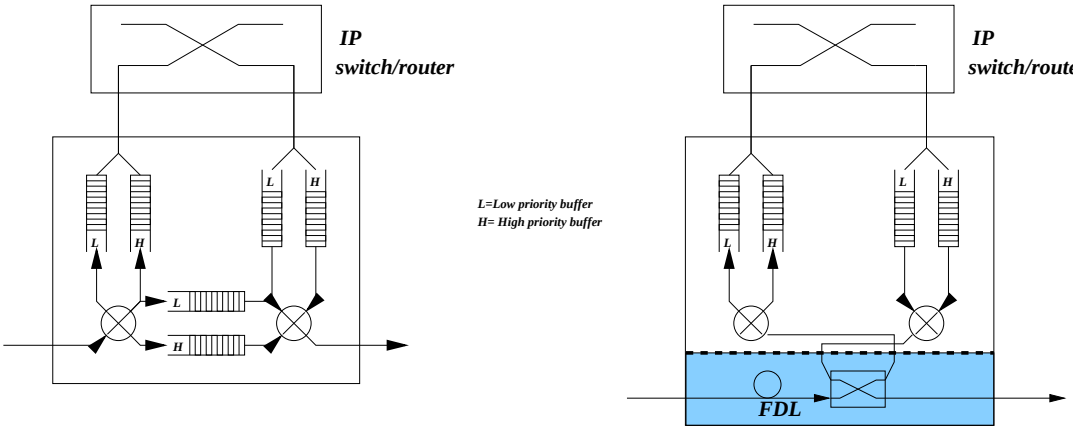

Figure 10.4.: The node architectures of DPT/SRP (left) and PrimeNet (right). The shaded area remains all-optical.

two packets under 10%, switching times in the order of 500 ns are required. For a 64-byte packet switches would have to be 100 times as fast! This together with the requirement for a fast header evaluation and control of the switches is very unlikely to become feasible in the near future. Thus, we conclude that a FDL architecture is best done with a fixed and not-too-small packet size. Out of the reasons described above and in the context of a better TCP performance (see section 4.2.2) we decided for a fixed packet size of 9216 byte in PrimeNet. Given the architecture introduced in the previous sections, it is absolutely necessary to have *packets* of a fixed length, not necessarily *slots*. If the packet length would be variable, it would be unavoidable to cut the transmission of packets whenever a packet arrives on the ring, as it is the case in CSMA/RN or HORNET.
On the other hand there is a waste of bandwidth attributed to fixed slot sizes. The ATM-"cell tax" is a famous example for that. In result the need for the aggregation of packets arises. This can be performed in different ways. Virtual output queueing (VOQ) as it is employed in RINGO [AVR+02, SH99] is a good pre-requisite for the aggregation of packets that are going to the the same destination. Whenever the queue is not empty and the packet is not filled a node may add packets from the queue to the aggregate. At the receiving side, packets are to be extracted from the aggregate before they are processed on.
Together with a Virtual Input Queueing it would even be possible to use the "pipe" of large aggregates in the same way as it is done in PoS (Packet over SONET, cf. sec. 5.1). This would result in a byte-oriented transmission line of variable bandwidth.

10.2.1. Modification of SRP

Since most of the protocol features that have been considered necessary are already included in SRP, the decision was to modify this protocol such that it could work on the simplified node architecture that is assumed here. As it can be seen in figure 10.4, there are four main

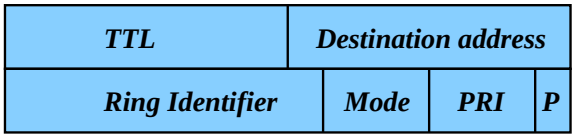

Figure 10.5.: The proposed frame header for FDL-SRP.

differences between the node architecture for PrimeNet and DPT/SRP:

- There is the possibility for an optical cut-through in PrimeNet. To make use of this, the ordering of packet treatment has to be changed. Low priority packets not destined for a node should be passed on optically before sourcing own traffic onto the ring. (This is different in SRP, where it is possible to buffer incoming packets.)

- There are possible ring numbers between 1 and $(N-1)*x$ with x being the number of free spectral ranges in use. These rings are not necessarily counterdirectional, resulting in the need to explicitly find a *mate* that has a shortest hop distance to the upstream node. This hop distance is not necessarily 1, which leads to explicit addressing of control packets. These control packets have to be relayed by intermediate nodes to reach the upstream node. The RI (Ring Identifier) field in the packet header has to be significantly longer than 1 bit. We chose a number of 9 bit to make up 512 rings in maximum, corresponding to e.g. 64x64 AWG using 8 FSRs. Figure 10.5 shows the 4 byte generic header of FDL-SRP.

- There is only one transit buffer of length 1 for both priority classes. Since SRP buffers low priority packets that come in from the ring in its Low Priority Transit Buffer (LPTB), the length of this buffer can be taken as a measure for the load in the ring. This is not possible in PrimeNet, so we had to look for alternatives. There are two possibilities that may be combined: count the bytes that transit a certain node in a certain time and observe the length of the own transmit queue.

- Following the different node architecture, the packet size is fixed to 9216 byte, the maximum packet size in SRP. This requires an aggregation of packets according to their destination MAC address and priority class. The way this is done in the first approach (and in the simulation model) is quite simple: There is only one low (LP) and one high priority (HP) queue per MAC. The LP queue is emptied as long as:

  – there is another packet is the queue *and*
  – this packet has the same destination address *and*
  – it fits into the aggregate.

  HP packets are not aggregated at all.

10.2.2. Protocol operation

10.2.2.1. Priority classes

There are two priorities, a *high* and a *low* class. High priority (HP) packets cannot be blocked by intermediate nodes once they entered the ring. Therefore it is possible to calculate a fixed transmission delay for such packets. This delay consists of the number of links and FDLs that a packet has to traverse on its way to the destination. Because these packets cannot be blocked, care has to be taken when assigning this priority class to a packet. The priority field in the header allows for eight priority classes. The reason for having potentially more priority classes is that there are 3 bits in the ToS (Type of Service) byte of the IPv4 header (see Fig. 4.2) that make up the so-called *precedence field* [Pos81e]. This kind of *coloring* of IP packets is already supported by Cisco [Pap00], but is still more or less proprietary. In any case, some mapping function has to be applied between the two classes that are supported inside the MAC and the fine grain QoS support that is possible using 8 classes. Usually this task would be fulfilled by a scheduling algorithm that controls the flow of packets from QoS-marked queues. For now, we consider the QoS mapping, signaling and scheduling a task of the upper layers and thus to be outside the scope of this work. Control packets are treated with high priority, too. The fairness algorithm only applies to low priority (LP) packets.

10.2.2.2. Basic access

As already mentioned, the FDL is used to store the incoming packet while evaluating its header. Therefore it is shown in figure 10.6 (a) that the packet is evaluated in the MAC layer and sent down to the PHY layer if it is not to be received by the node. For the sake of clarity it has to be reemphasized here that the actual payload is not touched and conceptually remains in the medium until it reaches its destination. A node is allowed to transmit packets as long as its FDL is empty. It may start transmitting, however, when the incoming packet is to be received.

When the node has high priority (HP) packets to send (regardless of any low priority data), it may do so whenever the FDL is empty. If not, and the packet in the FDL turns out to be of high priority (HP) as well and is a transit packet, then the node has to defer its transmission. If, however, the incoming packet is a LP, then the switch is set to *cross* and the packet is being received regardless of its destination address. This situation is illustrated in figure 10.6 (b). It is then sent up the stack for a re-routing in the network layer (this will most probably mean the IP router). There is no other way to separate the priority classes since there is only one FDL per interface. In result, sending high priority packets may in return increase the LP queue length in a node. There are good reasons to send such a LP packet up the stack, however. There may be interfaces (=rings) with a shorter hop distance to the receiver or with a lower LP queue length that could enable the transport of the received packet to its destination even faster than on the original ring.

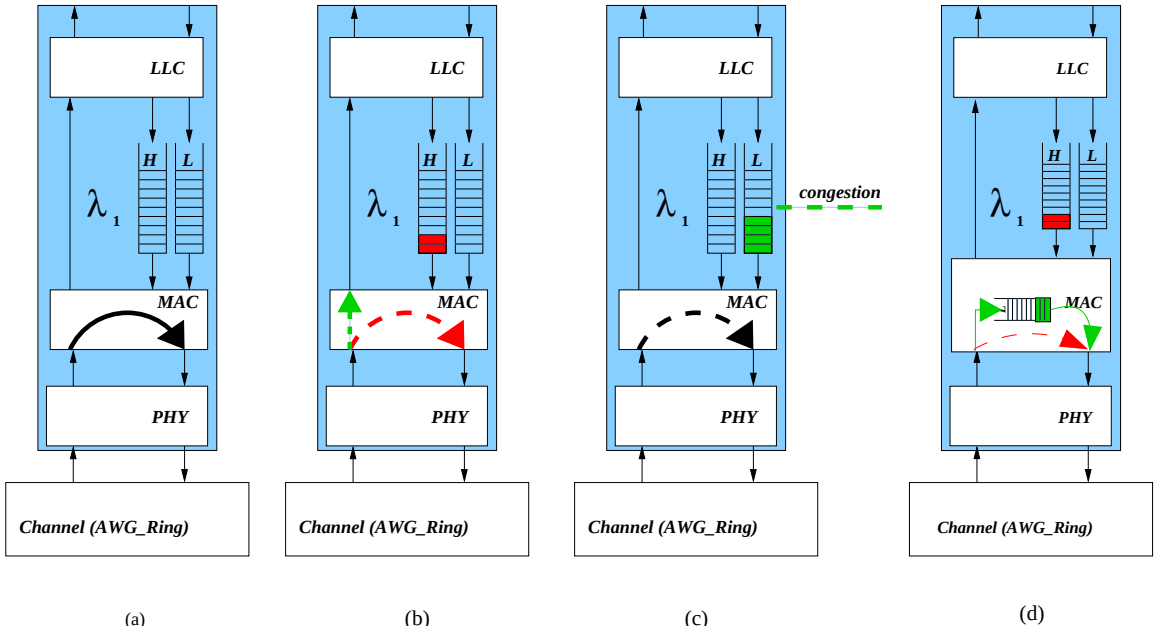

Figure 10.6.: (a): The node has nothing to send, (b): The node has high priority data, (c): The node has low priority data.

The proposed policy leads to a strict separation of priority classes. Because the transmission of HP packets increases the LP load for a certain node, it may be desirable to relax this separation a little. This can be done using a threshold in the HP queue or a HOL (head-of-line) timer that switches between a "nice" behaviour (let some LP packets transit and wait for "regular" access) and the rigorous described above.

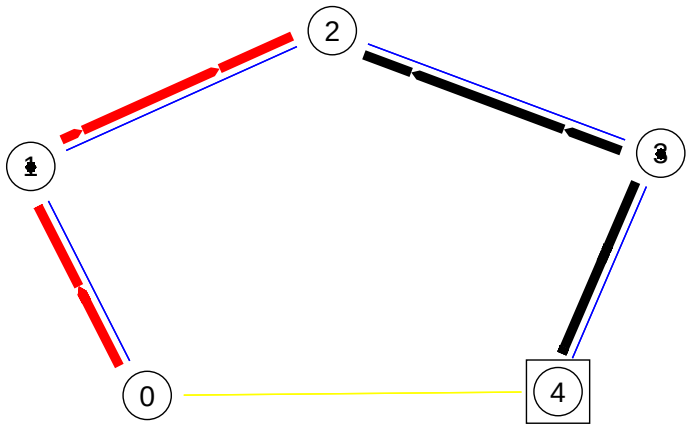

Figure 10.7.: Unfairness in a bi-directional ring configuration.

10.3. Introducing fairness

10.3.1. Unfairness in the basic access mechanism

Without any fairness algorithm applied to the rings, downstream stations suffer severe starvation problems. This is because of the fact that a node employs a carrier sense mechanism and is only allowed to send if the FDL is empty. Whenever an upstream node starts to transmit packets, the probability of finding the FDL occupied increases. Figure 10.8 illustrates this case for a 5 node bi-directional ring. All stations try to transmit packets to node no. 2. This means that under a high offered load stations 0 and 4 in figure 10.7 transmit all the time while the stations 1 and 3 are almost never allowed to send. This is shown in the figure by the overflowing nodes 1 and 3.

It is easy to understand that some algorithm is needed that prevents an upstream station from taking too much of the available bandwidth. Because of the low mean hop distance this algorithm should care for local fairness and introduce little to no overhead to the parts of the network that are not affected by the traffic that is controlled. Within the next section one of the possible candidate algorithms is chosen and adapted to the node architecture that was proposed earlier.

10.3.2. Fairness Algorithm

Algorithms to ensure each node a fair access to the medium have been treated vastly in the literature. Some of these were introduced in the previous chapter 6.
The fairness algorithm that is introduced next is very similar to SRP/RPR, with few exceptions that are explained together with the main parameters. The names of the parameters are similar to the values in SRP, but all of the counters are normalized to 1. Thus, a *lp_my_usage* value of 0.3 means that the node got around 30% of the bandwidth over the last few milliseconds.

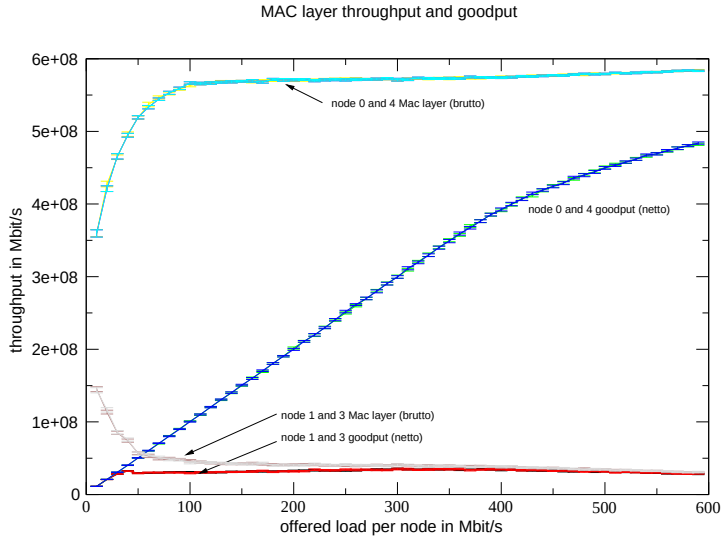

Figure 10.8.: Throughput of a bidirectional ring configuration without any fairness mechanism applied.

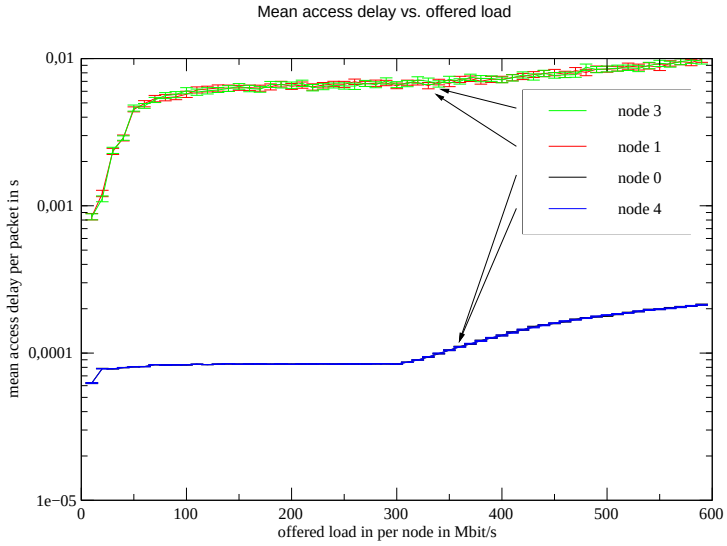

Figure 10.9.: Mean access delay (mean queuing time) w/o fairness. Configuration as in fig. 10.8

The basic mechanism to control the rate at which a node is sourcing packets onto the ring is a modified token bucket algorithm. It is modified in the sense that the rate at which tokens are produced can be decreased whenever a downstream node suffers congestion. It is increased again automatically according to a couple of parameters that are introduced next.
A node has a boolean variable *congested* that indicates that the LP transmit queue is filled above a certain threshold (e.g. half of its total length). If this is the case, the node asks its *mate*[2] to transmit the dynamic average of its own LP data rate (*lp_my_usage*) upstream, i.e. on another wavelength that has the shortest hop distance towards the upstream MAC. Because of the nature of PrimeNet, this MAC may be one or more hops away. The *usage packet* is therefore marked with destination address and ring identifier. Whenever this packet arrives at the upstream node, it is handed over to the MAC that is identified by the ring identifier. The MAC then reduces its own bucket size to the value that it received. The token production rate is implicitly reduced, too.

Configurable parameters The constants listed in table 10.1 determine the speed of the rate adaptation. They have to be configured such that a node does not assign too much transmission credit to itself too fast. Since the five of them are almost independent, simulations are performed in chapter 10.4.4.1 to find out good values.

DECAY_INTERVAL	time interval for the recomputation of the fairness values	number of packets
AGE	used to for the ageing of *my_usage* and *fd_rate*	positive integer, unit-less
LP_ALLOW	Low pass filter that determines the speed of the rate increase when no usage packet was received	positive integer, unit-less
LP_MY_USAGE, LP_FD_RATE	Low-pass filter value to compute long-term averages of *my_usage*, *fd_rate*	positive integer, unit-less

Table 10.1.: Configurable parameters

As stated above, the basic mechanism used here is a token bucket. This is generally characterized by two parameters, r and B, that stand for the *rate* at which the bucket is filled and the *bucket size*. The amount of data that a node is allowed to send in a certain period of time t is $r/t + b$ for a bucket initially filled with $b \leq B$ tokens. In the long run,

[2]The *mate* is the partner MAC that has the shortest hop distance to the node containing the upstream MAC. The mapping is assumed to be fixed an can be found out using a simple algorithm that is shown in chapter A. Although every MAC has a *mate*, this mapping is not necessarily 1:1. There may be MACs that serve as *mate* for more than one other MAC, resulting in unequally shared control traffic in the network.

the mean data rate that the node may source equals r.
A general expression for the data that may be sent using a token bucket filter in a certain time Δt is:

$$b + \int_{t_0}^{t_0+\Delta t} r \cdot dt. \tag{10.1}$$

The problem we face here is to dynamically adjust r and b to the traffic conditions in the ring. In an empty ring, r should be equal to the full line rate. Because a node is allowed to send up to *allow_usage* we can identify $B = 1$ at every instance in time. The rate at which the bucket is filled is not constant as in other token bucket schemes.[3] As stated above, all counters here are normalized to 1. This is done dividing them by MAX_LINE_RATE. Next we demonstrate the rate adaptation algorithm. Let us first assume an empty queue in the node. Every $DECAY\_INTERVAL(t = 0, 1, ...)$ the *allowed_usage* builds up until the bucket is filled.

$$\begin{aligned}
allowed_usage\prime &= allowed_usage + \frac{1 - allowed_usage}{LP_ALLOW} \\
allowed_usage(0) &= 0 \\
allowed_usage\prime(0) &= \frac{1}{LP_ALLOW} \\
allowed_usage\prime(1) &= \frac{1}{LP_ALLOW}\left(2 - \frac{1}{LP_ALLOW}\right) \\
allowed_usage\prime(t) &= 1 - \left(1 - \frac{1}{LP_ALLOW}\right)^{(t+1)}
\end{aligned}$$

To have the *allowed_usage* as a function of time enables us to calculate the slope of this function:

$$\frac{allowed\_usage(t)}{dt} = -\ln\left(1 - \frac{1}{LP\_ALLOW}\right) \cdot \left(1 - \frac{1}{LP\_ALLOW}\right)^{(t+1)} \tag{10.2}$$

We observe that the right term of the above equation approaches 0. This is similar to an overflowing bucket (tokens that are generated when the bucket is full are discarded). When the node starts to transmit, it may do so at full line rate, and for every packet that it transmits it increases its *my_usage* counter by $\frac{1}{MAX\_LINE\_RATE}$. Because it may send if $my\_usage \leq allow\_usage$ this is equivalent to taking that many tokens from the bucket. After each $DECAY\_INTERVAL$ *my_usage* is decreased, which means that tokens are generated again. For the node operating under maximum load ($my\_usage = allow\_usage$) the total rate at which tokens are generated therefore results in:

[3] We could see in eqns. (6.2) – (6.5) the amount of octets that a node is allowed to transmit in the next $DECAY\_INTERVAL$ (this is b).

$$
\begin{aligned}
my\_usage\prime &= my\_usage \cdot \left(1 - \frac{1}{AGE}\right) \\
my\_usage\prime(0) &= 0 \\
my\_usage\prime(1) &= \left(\frac{1}{LP\_ALLOW}\right) \cdot \left(1 - \frac{1}{AGE}\right) \\
my\_usage\prime(t) &= \left(1 - \left(1 - \frac{1}{LP\_ALLOW}\right)^{t}\right) \cdot \left(1 - \frac{1}{AGE}\right) \\
r &= allow\_usage\prime(t) - allow\_usage(t) - my\_usage\prime(t) + my\_usage(t) \\
&= allow\_usage\prime(t) - my\_usage\prime(t) \\
&= 1 - \left(1 - \frac{1}{LP\_ALLOW}\right)^{(t+1)} - \left(1 - \left(1 - \frac{1}{LP\_ALLOW}\right)^{t}\right) \cdot \left(1 - \frac{1}{AGE}\right) \\
&= \frac{1}{AGE} + \left(1 - \frac{1}{AGE}\right) \cdot \left(1 - \frac{1}{LP\_ALLOW}\right)^{t} - \left(1 - \frac{1}{LP\_ALLOW}\right)^{(t+1)}
\end{aligned}
$$

As we can see, the result is constant of $1/AGE$, when $AGE = LP\_ALLOW$. A permit to send a packet is $1/MAX\_LINE\_RATE$ tokens worth. A rate of $1/AGE$ therefore means that a station is allowed $DECAY\_INTERVAL$ packets in one $DECAY\_INTERVAL$, leading to a 100% load.

When the node receives a *usage* packet it sets its *allowed_usage* to the received value (*lp_my_usage* of the downstream node). Therefore, for $my\_usage \leq allowed\_usage$:

$$
r \leq \frac{allowed\_usage}{AGECOEFF \cdot DECAY\_INTERVAL} = \frac{allowed\_usage}{MAX\_LINE\_RATE}
$$

Since the incoming *lp_my_usage* is a fraction of MAX_LINE_RATE, at most the rate of the downstream node is generated in the next cycle.
While $my\_usage > allowed\_usage$ sending is prohibited.

What we showed here is that a certain value of the *allowed_usage* determines the token production rate to be of exactly rate in the following $DECAY\_INTERVAL$.
Using this result it becomes obvious why it is enough for a congested station to send its long-term average data rate upstream. The receiving node accepts the received rate as its own *allowed_usage* and effectively reduces its data rate to the received value.

10.4. Simulation results

To evaluate the behavior of the fairness algorithm, a simulation of a 5-node network like the one shown in figure 10.7 was implemented in ns-2. The details of this implementation are explained in the appendix A. Initially, two bi-directional rings are set up using wavelengths λ_1 and λ_4. We refer to this as the [1,4]-configuration in the following.

10.4.1. Exponential On/Off traffic over UDP

The first approach takes a rather simplistic traffic model that is motivated by the packet length distributions observed in todays Internet (see chapter 4). Three sources generating packets of lengths 40, 576 and 1500 bytes, respectively, were placed in each node, thus generating a mix of packet lengths similar to the one observed in real traffic measurements. Just like in the first simulation in figure 10.8, all other nodes transmit unidirectionally to node 2. A UDP-like transport protocol (basically, since the error detection capabilities are not used, none at all) was chosen in ns-2. The offered load per node varied between 10 Mbit/s and 600 Mbit/s. The total line rate was set to 622 Mbit/s.
Assuming a fair access to the medium, it could be expected that both throughput and mean access delay would be equal for the nodes that share one link. These links are between nodes 1 and 2 (for nodes 0 and 1) and between 3 and 2 (for nodes 3 and 4). Due to the symmetry of the network the lines for nodes 0 and 4 and the lines for nodes 1 and 3 are very close to each other.
All simulations that follow were performed using the AKAROA-2/ns-2 combination that is described in appendix B. This assures a 95% confidence level for the mean to be in the 5% half-width.
Figure 10.10 shows the *brutto* and *netto* throughput in the ring. We refer to the *brutto* throughput as the number of *Jumbo* packets (9216 bytes) that are transmitted by a node in a certain time (multiplied with the packet length in bits). The *netto* throughput is sometimes called *goodput* and refers to the number of bits that were actually delivered to the destination application. It can be seen from the figure that the *netto* throughput of all stations is almost equal for all offered loads in the network. There is, however, a slight advantage of about 2% for the inner nodes 1 and 3. This is subject to fine tuning of the algorithm, which will be performed later.
Figure 10.11 shows the mean access delay of the fragments in the *Jumbo* packets. This delay is the queuing delay in network interface until the time the Jumbo packet is actually being sent onto the fiber. No transmission delay is included here since this would be just a fixed overhead of a few microseconds.
The main problem of this fairness algorithm is already visible here: Its notion of fairness is based on throughput rather than access delay. For a low offered load the packets in the transmit queues of the inner nodes have to wait longer than the packets in the outer nodes' queues. This corresponds to the many almost empty Jumbo packets that are sent by the outer nodes. However, since the queuing times are still below 1 ms, this should not pose a problem under realistic circumstances.

10.4.2. Exponential On/Off traffic over TCP

In the first step a unidirectional traffic model was chosen to evaluate the performance of the system. Since the vast majority of the Internet traffic is indeed controlled by TCP rather than UDP, the next step aims at the inclusion of TCP between the existing packet generating process and the IP and MAC layer. It is clear that the large number (around 50%) of very short packets that are observed in traffic measurements on the IP level are

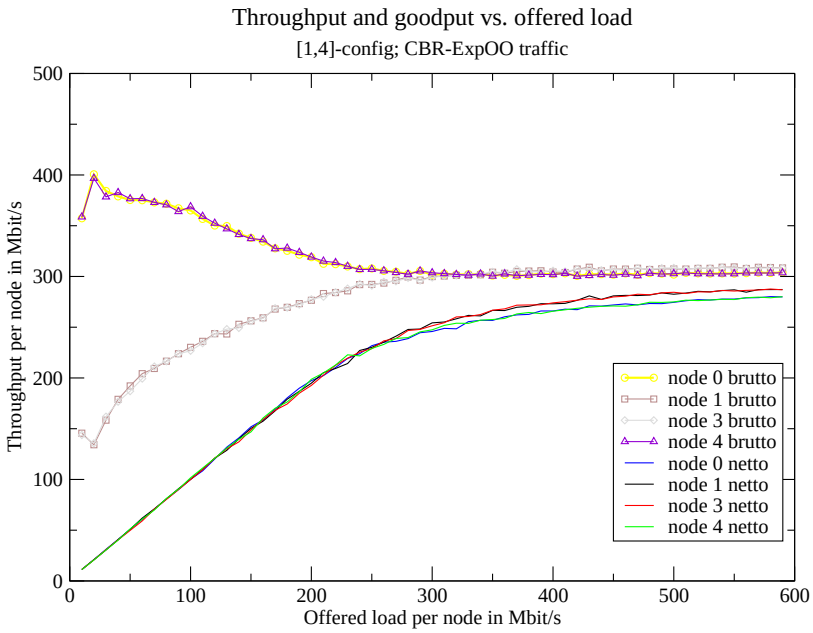

Figure 10.10.: Throughput of nodes 0,1,3 and 4 transmitting to node 2. Exponential On/Off traffic of 3 different packet lengths.

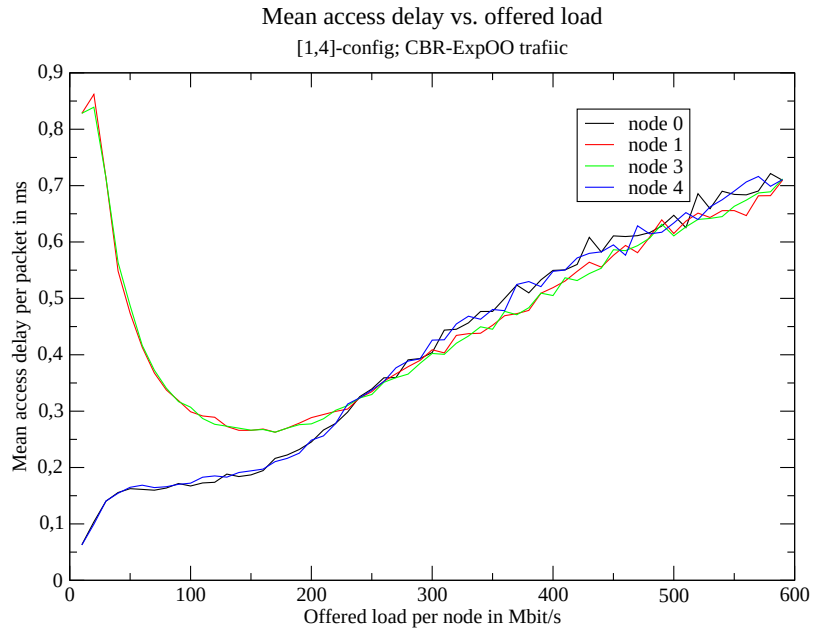

Figure 10.11.: Mean access delay of packets from nodes 0,1,3 and 4 transmitting to node 2. Exponential On/Off traffic of 3 different packet lengths.

actually TCP's SYNs and ACKs. This number of ACKs differs with the numerous TCP implementations that are being used today. Therefore, and to be comparable with the previous simulations, trace files of the first simulations (without TCP) were generated and taken as input to the second one. The event that was recorded was the *enqueue* operation of the transmit queue in one node.[4]
Surprisingly, the behavior of the network became unstable for a medium load and under high load, the inner nodes either got around 40% of the total link bandwidth (they should get 50% to be fair) or nothing at all. After investigating this problem it was found that the dynamic behavior of TCP's congestion window mechanism was exactly the reason for this. As explained in section 4.1.2, the *slow start mechanism* is mandatory for all TCP implementations. Because of this, the TCP sender transmits only a few fragments in the beginning or after fragment loss. This may lead to the situation where a node waits forever for the medium to become free, because its queue length did not grow over the *congested threshold*.

To illustrate the problem, simulations were performed where the inner TCP connections (from nodes 1 and 3 to node 2) started 100 ms later than the outer connections. This way, the `cwnd` of the outer connections (nodes 0 and 4 in figures 10.12 and 10.13 could be already open to the extent that the channel was totally filled. When doing so, the inner nodes constantly starve under high load. The above mentioned situation of a node getting 40% or so of the bandwidth did not occur anymore. The instability was thus only generated in the simulated case where the inner node could open its congestion window early enough to cross the *congested threshold* before the outer node could fill the medium totally. This is also an explanation for the not-so-smooth curves in the corresponding figures.[5]
Given the fact that the usual traffic observed in a high speed metro or backbone network is a mix of many TCP connections, the situation considered here may be artificial or even pathological. However, it would always occur when a previously unloaded node wanted to start a TCP connection over an already full link.

10.4.3. Introducing a Head-of-line timer

To avoid this, another timer was introduced that measures the medium access delay of the first packet in the queue. Whenever this timer elapses a flag called *HOL_flag* is set and a *usage packet* is sent upstream immediately. In the following $DECAY\_INTERVAL$ the node declares itself *congested* and follows the rules described above. The value of this HOL timer is motivated by a basic result from queuing theory – that the time a customer has to wait for service is the inverse of the service rate. The timer is computed in multiples of the packet transmission time as follows:

[4] This is marked by a "+" in the first column of ns-2's trace file output.

[5] The simulations results were often bi-stable, i.e. either of the connections got full bandwidth over a certain time and zero in the other. This resulted in extremely long simulation durations to reach a given confidence interval. It is even questionable if the mean value that is shown in figures 10.12 and 10.13 is really meaningful. It can however be seen as a long-term average rather than the instantaneous throughput and delay of a single connection.

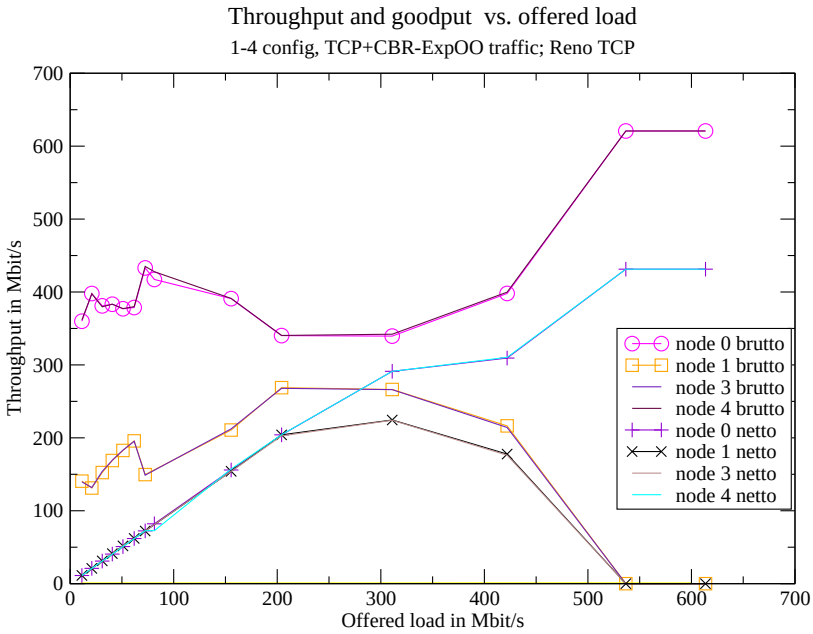

Figure 10.12.: Throughput of nodes 0,1,3 and 4 transmitting to node 2. Exponential On/Off traffic of 3 different packet lengths - over TCP!

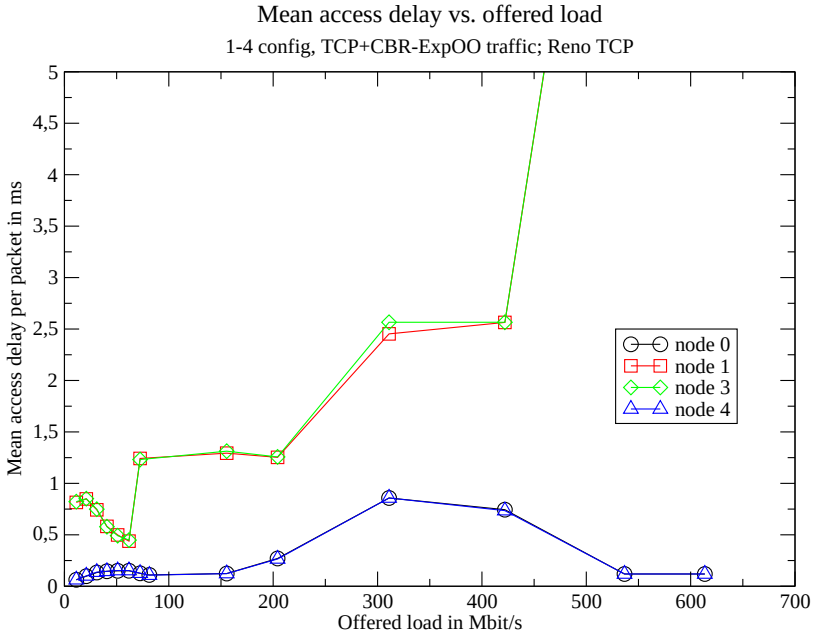

Figure 10.13.: Mean access delay of packets from nodes 0,1,3 and 4 transmitting to node 2. Exponential On/Off traffic of 3 different packet lengths - over TCP!

$$HOL = \begin{cases} \frac{1}{lp\_my\_usage} & : \quad 0 < lp\_my\_usage \leq \frac{1}{DECAY\_INTERVAL} \\ DECAY\_INTERVAL & : \quad else \end{cases} \tag{10.3}$$

The lower term (the setting of $HOL = DECAY\_INTERVAL$ for a lp_my_usage of zero) is done to avoid infinite waiting time. This way, when a node did not transmit for a long time ($lp\_my\_usage = 0$) it has to wait only one packet time until it declares itself *congested* and is allowed to send a *usage packet*. The higher the average load in the node is, the shorter is the waiting time. For a load of, say $lp\_my\_usage = 0.5$, a node has to let pass on the ring only two packets until it starts complaining with the upstream node. Since the HOL timer is started immediately when a segment arrives at the head of the line this means that the first of these two packets may be transmitted by the node itself (it was the predecessor in the queue). Even if a node would by any chance not get the bandwidth it demands (for instance due to some high priority transit traffic) the time instances at which the *usage_packets* would be sent would be spaced further and further, thus off-loading the system.
The change of the MAC fairness algorithm leads to an impressive result: the resources are shared in a fair manner now. All four connections show the same goodput. The (brutto) throughput of the aggregates in figure 10.14 is under light load a little higher for the outer nodes. This has no influence on the goodput, because the channel is not filled by then. A little more surprising is the mean access delay, that is slightly higher for the inner nodes. This may be explained by the backpressure mechanism that makes nodes 1 and 3 wait for some time until the upper node 0 and 4 back off. This waiting time is reflected in figure 10.15.

10.4.4. Using a different topology

So far, the topology under consideration was a bi-directional ring, just as it is used in the original SRP. PrimeNet however allows for the use of different wavelengths, leading to topologies that have unidirectional paths between nodes that were not neighbored in the bi-directional ring. An example for this can be seen in figure 10.16. The next question that had to be answered was: Is the proposed fairness algorithm able to work on non-bidirectional rings? The main problem here is that the upstream node that is to receive the *usage packets* is not necessarily the downstream node in the other ring. Therefore, there has to be an algorithm that decides about the shortest path to the upstream node. In the simulation this is done by subtracting the number of the wavelength from the number of the node (i.e. the number of the input port at the AWG) modulo N. The wavelength with the least hop distance to the upstream node is then chosen to carry the *usage packets*.
To keep comparable to the previous simulations, the traffic scenario is the same, meaning that the remaining nodes have TCP connections to node 2. As can be seen from the figure, the shortest path between node 3 and 2 is now on wavelength λ_3 via node 1 and node 4, meaning that it has length 3. The path for the TCP ACKs from node 2 to 3, however, is the direct one on the outer wavelength λ_1. We refer to this topology as the [1-3]-configuration.

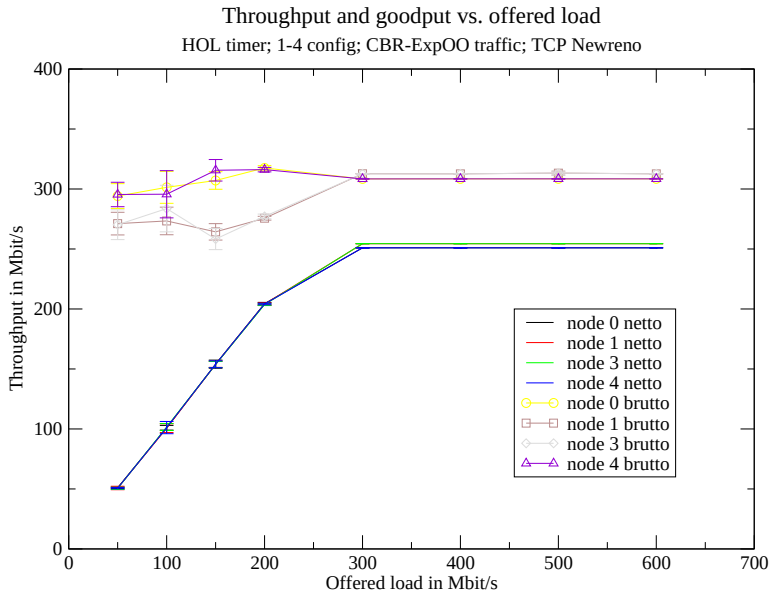

Figure 10.14.: Throughput of nodes 0,1,3 and 4 transmitting to node 2. Exponential On/Off traffic of 3 different packet lengths - over TCP. HOL timer based fairness algorithm.

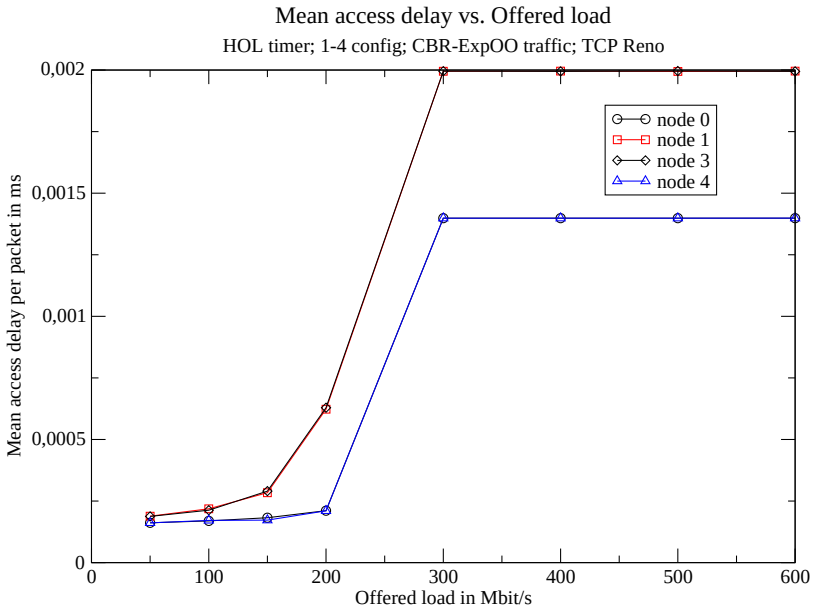

Figure 10.15.: Mean access delay of packets from nodes 0,1,3 and 4 transmitting to node 2. Exponential On/Off traffic of 3 different packet lengths - over TCP. HOL timer based fairness algorithm.

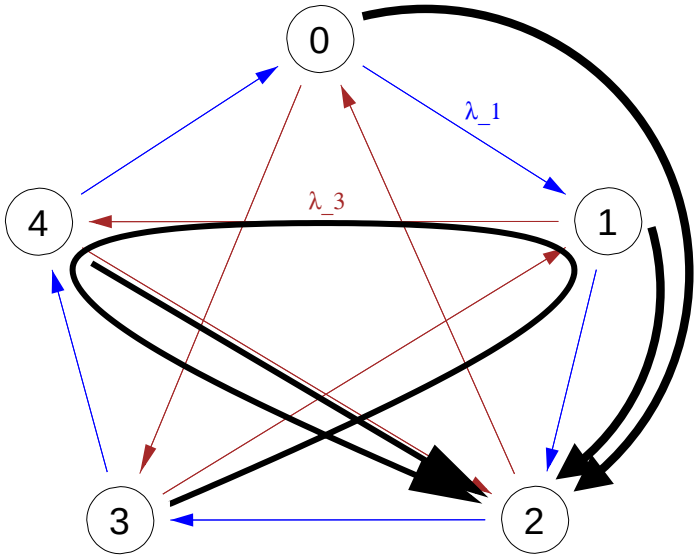

Figure 10.16.: Another possible topology – using wavelengths 1 and 3. The fat arrows show the four unidirectional connections.

To send a *usage_packet* from node 1 to node 0 (the upstream node) it takes now 3 hops, which delays the usage information.

10.4.4.1. Increasing the DECAY_INTERVAL

The first simulations of the given traffic (figure 10.17 again shows results of the unidirectional UDP-like traffic for the beginning) showed a slight decrease in throughput for all connections compared to the [1-4] configuration. This can be explained by the fact that now the data and control traffic are really multiplexed on one link. For instance, the *usage packets* that control the link between nodes 1 and 2 have to be sent be node 1 via the three links $1- > 4$, $4- > 2$ and $2- > 0$ an the other wavelength λ_3. Of these, at least link $4- > 2$ will be filled by two data connections (we refer to the term connections here for the directed traffic between source and destination). As stated before, the *usage_packets* have to be full packets, since the packet length is fixed. This leads to a large overhead and may even cause a total breakdown of data traffic when the $DECAY\_INTERVAL$ is not properly chosen. Because of the fixed packet length it was decided to let the $DECAY\_INTERVAL$ be integer multiples of the packet length. The impact of this huge overhead for the transmission of fairness messages could be seen in the first simulation results. Consequently, *usage packets* should be as rare as possible in the network. This has several consequences for the MAC and fairness algorithm:

- The event that causes the transmission of a *usage packet* is the HOL timeout, not the invocation of the $DECAY\_INTERVAL$ handler. In contrast to the SRP-fa, the *usage packet* is sent *at most once* per $DECAY\_INTERVAL$. If *usage packets* from

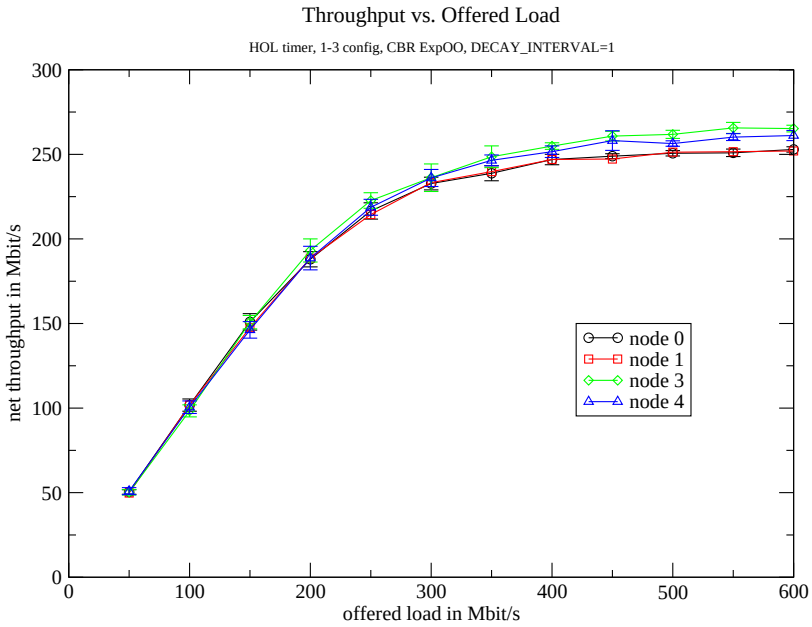

Figure 10.17.: Goodput vs. offered load for the [1-3]-configuration. Exponential On/Off-traffic with HOL-timer based access. The brutto throughput is not shown here. No surprises there.

other nodes have to be forwarded, subsequent HOL timeouts are also suppressed. This, however, does not rule out the possibility of having more than one *usage packet* on a link per $DECAY\_INTERVAL$. (If the own *usage packet* has been transmitted before.)

- The $DECAY\_INTERVAL$ value is increased to a value that keeps the fairness overhead in a reasonable area, e.g. ≤ 4.
- Higher values of the $DECAY\_INTERVAL$ timer mean that the counters in table 10.2 are computed less often. To achieve the same (impulse?) answer or timely behavior the low pass filter values (see table 10.1) have to adjusted.

By substituting the variable t with $\frac{t}{DECAY\_INTERVAL}$ in eqns. 10.2 and 10.3 the new values for the variables compute to:

$$NEW\_VALUE = \frac{1}{1 - (1 - (\frac{1}{OLD\_VALUE})^{DECAY\_INTERVAL)}} \tag{10.4}$$

Surprisingly, even though the original equations for the long term averages looked different, the modification of the filter parameters is the same for LP_MY_USAGE, LP_FD_RATE, LP_ALLOW and $AGECOEFF$. However, the decrease of the LP_ALLOW value leads to a faster increase of the *allowed_usage* counter. At long $DECAY\_INTERVALs$ the above formula obviously fails to keep the fairness in the network. Therefore, a simulation was performed that was aimed at finding a proper range for the LP_ALLOW constant. Figure 10.18 shows the throughput for all four nodes over a range from 2 to 4096. As it can be seen in the figure, a good fairness is achieved for a relatively wide range, indicating that the algorithm is quite robust against a change of this parameter. Values around 512 seem to give good results for the throughput of all nodes. The mean throughput is around 285 Mbit/s. Compared to the simulation with $DECAY\_INTERVAL = 1$ in figure 10.17 the overall throughput is about 10% higher. There is, unfortunately, bad news also: This fairness under high throughput is only achieved in the long term. The counter variables in the MAC of nodes 0 and 1 take about half a second to converge. This behavior is shown in figure 10.19 in contrast to the behavior with the original settings (as they were used to produce figure 10.17).

10.4.5. Problems with TCP, again...

The next step was again the simulation of the same traffic characteristics over a TCP connection from each of the four nodes to node 2. The result in figure 10.20 was an unpleasant surprise. The connection from node 3 to node 2 took only about 60% of the bandwidth of the other connections. The overall (brutto) throughput however did not indicate an unfair behavior of the MAC to the same extent. Obviously, the TCP connection in node 3 did not really fill the aggregates. It took some time to find the reason for this, but since the

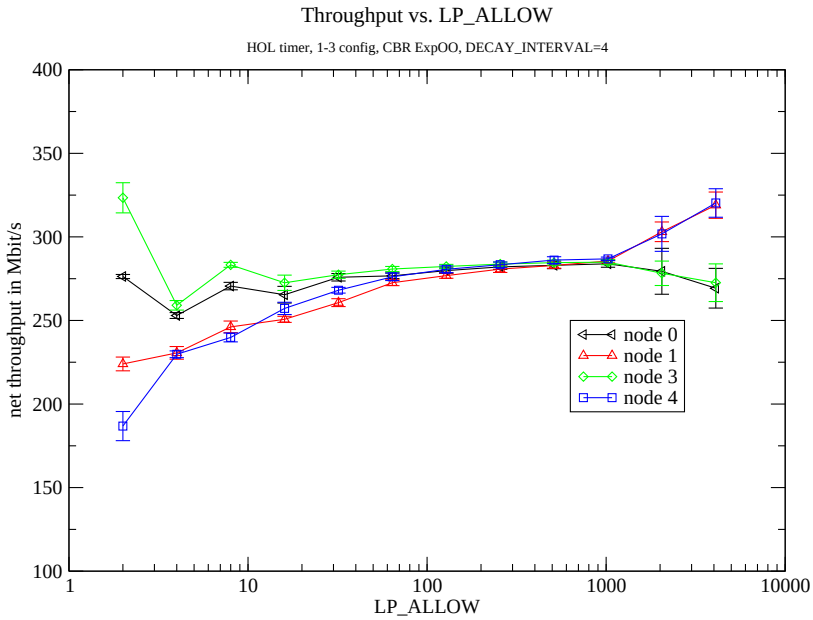

Figure 10.18.: Throughput vs. LP_ALLOW for an offered load of 600 Mbit/s per node. A rather wide range (between 64 and 1024) seems to give acceptable values.

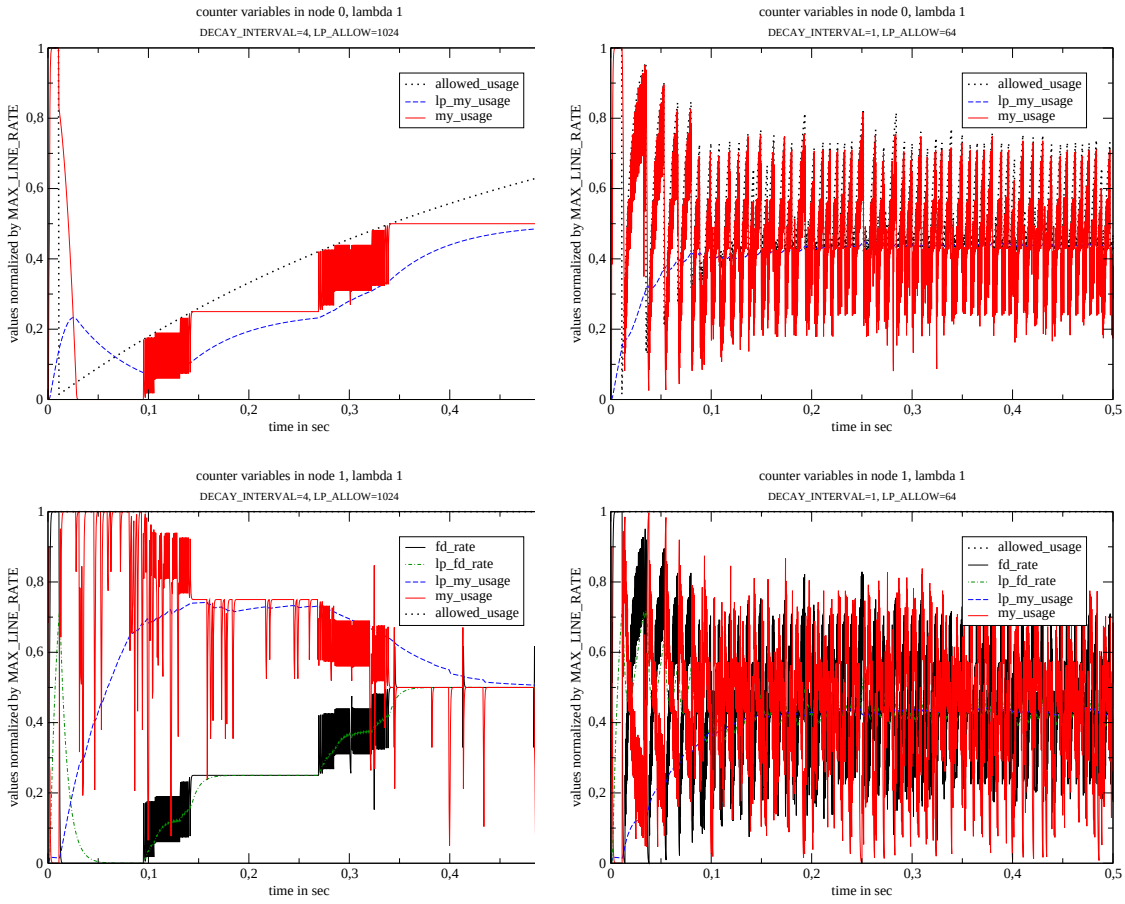

Figure 10.19.: Illustration of the timely behavior of the counter variables in node 0 (upper) and node 1 (lower pictures). Left column: DECAY_INTERVAL=4, LP_ALLOW=1024. Right column: DECAY_INTERVAL=1, LP_ALLOW=64.

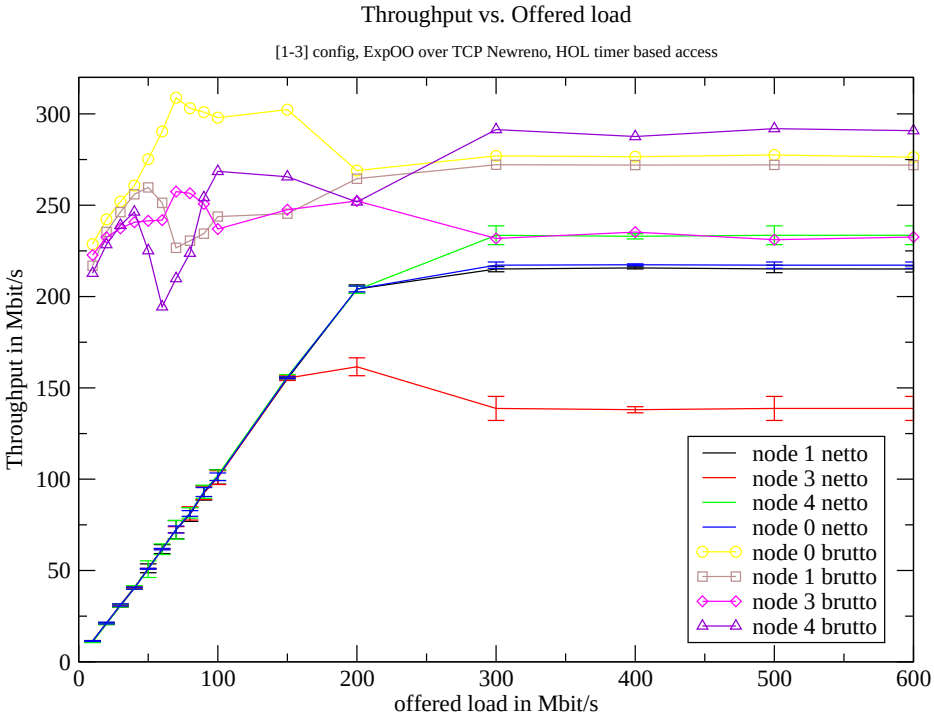

Figure 10.20.: Throughput vs. offered load in the [1-3] configuration with TCP! Note that 3 fills its Jumbo packets less that the other nodes do. Only 60% in average.

experience with the TCP slow start mechanism that led to the redesign of the MAC fairness algorithm we had become cautious. The `tcptrace` program that was developed at the MIT [She91] allows to analyze output traces of ns-2 simulations. Together with the `xplot` program it is an powerful tool to visualize the dynamics of TCP.

10.4.6. ...and the reason: packet reordering

A detailed look into the time sequence graph of this connection showed retransmissions of TCP segments. There were no losses of segments in the simulation, however. The reason for these retransmissions was the reordering of packets. In the center of figure 10.21 such a reordering can be observed. Obviously, one aggregate arrives out of sequence. All segments that were packed into the aggregate are marked by `tcptrace` with an "O" for "Out of sequence". There is a simple reason for this: To insert a *usage packet*, node 1 has to take a LP (low priority) data packet off the ring (remember, *usage packets* are HP!). This packet is sent onto the ring again as soon as there is enough free space, i.e. the delay line is empty. Much in the same way as there are always some bold car drivers following a fire engine

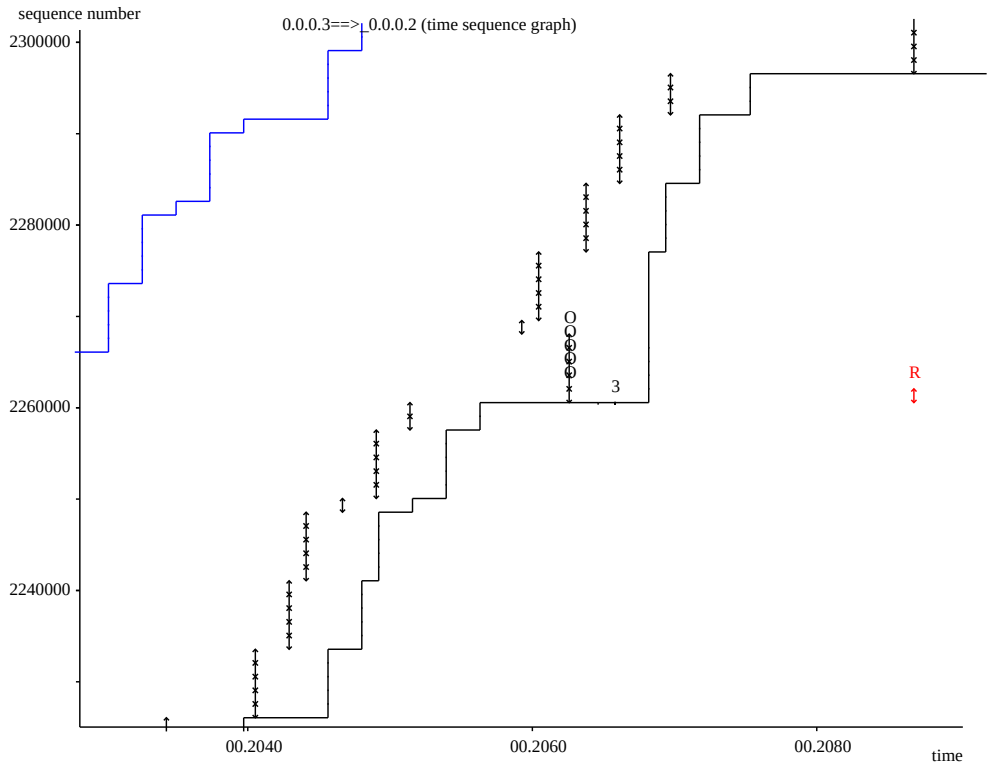

Figure 10.21.: A cutout from the time sequence graph of the connection from node 3 to node 2. Note the retransmitted segment on the right edge of the figure.

through the traffic jam there are some aggregates that rush through before the LP data packet can be inserted again. This leads to the reordering of LP packets. But how and why is TCP affected by this? Shouldn't it be robust against this, especially since TCP is designed to use IP packets that have no notion of a sequence anyway? The answer is yes and no. The original TCP [Pos81d] introduced the congestion window mechanism to do flow and error control in a combined way. Every incoming segment is acknowledged with the number of bytes that have been received "in-sequence" up to this segment. The order in which the packets arrive at the receiver thus may have an influence on the speed of the opening of the `cwnd` but not directly on the size of it.

The reason for the problems here is the *fast retransmit mechanism* that assumes that duplicate ACKs are a sign of a lost segment. If the duplicate acks are caused, however, by reordering, then the fast retransmission is unnecessary and wastes bandwidth by sending the same data twice. To make things worse, the sender reduces its `cwnd` and the slow start threshold in the belief that the dupacks were caused by packet loss. This is what caused the reduction of data rate over the connection from node 3. The retransmission of the missing segment can be observed on the right edge of figure 10.21. It is marked with an "R" and is

obviously spurious, since the missing segment had indeed arrived long before. While TCP is robust against a simple reordering of two segments (because of the `dupack_` threshold of 3) it fails here because of the aggregation of several (5 in this case) segments into a *jumbo packet*.
Packet reordering is a problem that also exists in the Internet of today. Bennett et al. published measurements in 1999 that showed that 90% of the observed 50-packet-ICMP-ping bursts to 140 Internet hosts that were topologically close to the main exchange point MAE-East suffered reorderings [BPS99]. A discussion of the influence of reordering on TCP can be found there, too.

10.4.6.1. Making TCP robust to packet reordering

The finding that packet reorder is quite common in the Internet fostered numerous investigations on how to deal with this problem. Two recent publications came to quite different conclusions: While Blanton and Allman [BA02] emphasize on making TCP able to distinguish between segment reorder and loss, Laor and Gendel [LG02] conclude that is preferable to avoid the reordering of packets in the router hardware. Both these main directions will be sketched next together with their implications of the work done here.

TCP related ways to deal with reorder The scenario that was simulated to achieve the results in the first paper was a single bulk TCP data transfer. Being similar to the traffic simulated here, this kind of one long TCP connection is the worst case when considering the effects of reordering. Short TCP connections suffer less in the presence of reordering because in most cases the `cwnd` will not open far enough to be harmed severely by the fast retransmit.
Several candidate algorithms to detect a spurious retransmission on the sender side were proposed:

- The TCP Eifel algorithm developed by Ludwig [LK00] offers two methods to decide whether an incoming ACK is for the original or the retransmitted segment. One way is to use 2 of the reserved bits in the TCP header to explicitly mark retransmitted segments (and the ACKs for the retransmissions). This option has to be negotiated at the start. The alternative method makes use of the TCP timestamp option standardized in [JBB92]. Here, the retransmit is marked with a 12-byte TCP time stamp (`ts_first_rexmit`). Whenever the sender receives an ACK with a timestamp value less than this, it concludes that the original segment had arrived meanwhile. The actual algorithm is very simple: Whenever the sender retransmits, it stores the original values of `cwnd` and `ssthresh_`. After finding that the retransmission was spurious, the original values are restored.

- When using the DSACK option (see [FMMP00]) the receiver informs the sender about a segment that was received twice by sending a DSACK (Duplicate SACK). This way

the information about a spurious retransmission is explicit and fast, given that the ACK returns to the sender and is not dropped in the network.

- a timing of the ACK to decide if it came too early to be meant for the retransmitted segment.

The reaction on the detection of a spurious retransmission is two-fold: First, just as described together with the Eifel algorithm, the original size of `cwnd` and `ssthresh_` is restored. This may immediately cause a burst of several segments sourced into the network. To avoid this, some smoothing function (called Limited Transmit in [ABF01]) is needed. Second, the `dupack_` threshold may be increased. Several ways of doing so are compared in [BA02]. A simple increase by 1 or $K > 1$ per spurious retransmission seems to be very successful already and reduces the number of spurious retransmissions by one order of magnitude. An upper limit on `dupthresh_` is (`cwnd` $- 1$) to make sure that enough data can be generated to cause a fast retransmit.
For the case considered in this work things are a little more complicated. Because of the aggregation, a number of TCP segments are usually transmitted within one *jumbo packet*, each of these clocking an ACK at the receiver. Since reordering takes place at the level of the *jumbo packets* here, the `dupack_` threshold should be increased by more the number of segments per aggregate with each reordering. A high `dupack_` value however slows the detection of and increases the possibility of not finding lost segments (and thus falling into the retransmission timeout RTO).
Because the ns-2 version 2.1b7 implementation of TCP SACK is explictly stated "buggy" and the long list of necessary changes to ns-2's TCP in [BA02] the question of adapting TCP to the reordering of packets is not touched here but left for further work, possibly after the transition of the whole simulation model to a newer version of ns-2.

Hardware related ways to avoid reorder The second possibility to deal with reordering is the construction of hardware (IP routers in the usual case) that a-priori avoids reordering. The basic feature that causes reorder in a router is load balancing over output queues. If at all, this should be done in a ow sensitive way. It is much harder to avoid reordering in the architecture that is under consideration here. Remember, the basic aims of the node architecture were:

- to keep things as simple as possible
- to leave the packets in the optical domain
- to separate two priority classes

Reordering comes from the separation of the priority classes, here. In order to keep the sequence of the incoming traffic, it is possible to take not only the one LP data packet off the ring that is replaced by the usage packet, but all following LP data packets until there is a hole in the incoming data stream. It is very unlikely that this can be done optically. Cascaded FDLs might be an option, but since the number of consecutive HP packets from

a node does not have an upper bound the length of this optical packet queue would have to be infinite. The alternative is to take the LP packets into the electronic domain and buffer them in a "recycle" queue (see Fig. 10.6). This of course drastically increases the O/E/O conversion rate in the node and is thus not desirable. A possibility is to use exactly one additional FDL per node, i.e. a recycle queue of length 1. The idea is to use this FDL as a real insertion buffer that is used whenever a usage packet has to be inserted. From that point in time on whole traffic in the ring could be switched through the FDL. Whenever the additional FDL becomes empty because of a hole in the data stream it might be taken out of the ring again. This would be enough for the one usage packet to be inserted, but not for HP data traffic. In any case, fighting reorder in a node architecture like this would result in a increased hardware complexity. Because of this, it seems desirable to go the first way, namely to make TCP robust against packet reorder.

my_usage	consumed tokens from the bucket	incremented by $\frac{1}{MAX\_LINE\_RATE}$ for every packet that is being sourced onto the ring, decremented by $\min(\frac{allow\_usage}{AGE}, \frac{my\_usage}{AGE})$ every $DECAY\_INTERVAL$
allowed_usage	number of tokens in the bucket	incremented by $\frac{1-allowed\_usage}{LP\_ALLOW}$ every $DECAY\_INTERVAL$
fd_rate	number of packets that have been forwarded on the ring	incremented by $\frac{1}{MAX\_LINE\_RATE}$ for every packet that is forwarded, decremented by $\frac{fd\_rate}{AGE}$ every $DECAY\_INTERVAL$
lp_my_usage	long-term average of the transmitted number of packets, in case of congestion this value is being sent upstream	recalculated using $\frac{(LP\_MY\_USAGE-1)\cdot lp\_my\_usage}{LP\_MY\_USAGE}$ every $DECAY\_INTERVAL$
lp_fd_rate	long-term average of the number of forwarded packets, needed to compare it against lp_my_usage to decide if a node is congested because of its own high load or because of the load the upstream nodes are generating.	recalculated using $\frac{(LP\_FD\_RATE-1)\cdot LP\_FD\_RATE}{LP\_FD\_RATE}$ every $DECAY\_INTERVAL$

Table 10.2.: Counters that observe traffic conditions.

11. Interconnection of Primenets

11.1. The AWG as a Cayley Graph

To interconnect several PrimeNets in a way that allows for a maximum fault tolerance, regular multihop networks may be constructed. One possible solution is the construction of Cayley graphs. This is motivated by the fact that the AWG itself makes up a permutation graph of the input and output ports for every wavelength. As described in section 7.5, a graph C =(V, G) is a (directed) Cayley graph with vertex set V if (V,*) is a finite group with $G \subset V \setminus \{I\}$ and the following condition holds for every two vertices (cf. [Big74]):

$$\text{Vertex } v_1 \in V \text{ is connected to vertex } v_2 \in V. \Leftrightarrow v_1 = v_2 * g \text{ for some } g \in G.$$

A permutation rule g can be represented by a string of digits $1, 2, 3, \ldots, n$ with n being the base of the permutation group S_n. Thus a generator $g_2 = 21543$ means to swap the first two digits and the third and fifth digit, while the fourth digit is invariant. This corresponds to the second column in the output matrix of the 5x5 AWG in Eq.2.9 shown in section 2.11. It represents the wavelength routing that signals on wavelength λ_2 experience in this AWG. Similar generators exist for the other wavelengths: $\lambda_1 : 15432, \lambda_3 : 32154, \lambda_4 : 43215, \lambda_5 : 54321$. They all show the symmetric property of the AWG in the fact that the re-application of e.g. g_1 onto itself leads to the identity element 12345 (and then back to g_2). This way, only two elements are generated by g_2 (out of the $n! = 120$ possible). The way the outputs of the AWG are exchanged to achieve a cyclic permutation over all wavelengths is described in section 8.2. Since this is also a permutation of outputs, we can write it in the form $g_1 = 15432$. After applying this to the original permutation ($g_2\prime = g_1 * g_2 = 21543 * 15432 = 23451$) appears. This operation is similar to the matrix multiplication in section 8.2, although again only the second column is considered here. The new generators that describe the behavior of the AWG are therefore:

$$g_1\prime = 12345, g_2\prime = 23451, g_3\prime = 34512, g_4\prime = 45123, g_5\prime = 51234$$

g_1, being the identity element e cannot be considered useful. Thus, the four generators $g_2, \ldots, g_5$ remain that describe the wavelength routing function of the AWG.

11.2. Building larger graphs

With the knowledge of PrimeNet being a Cayley graph it is possible to construct larger graphs that show all the desirable properties that Cayley graphs in general are known to

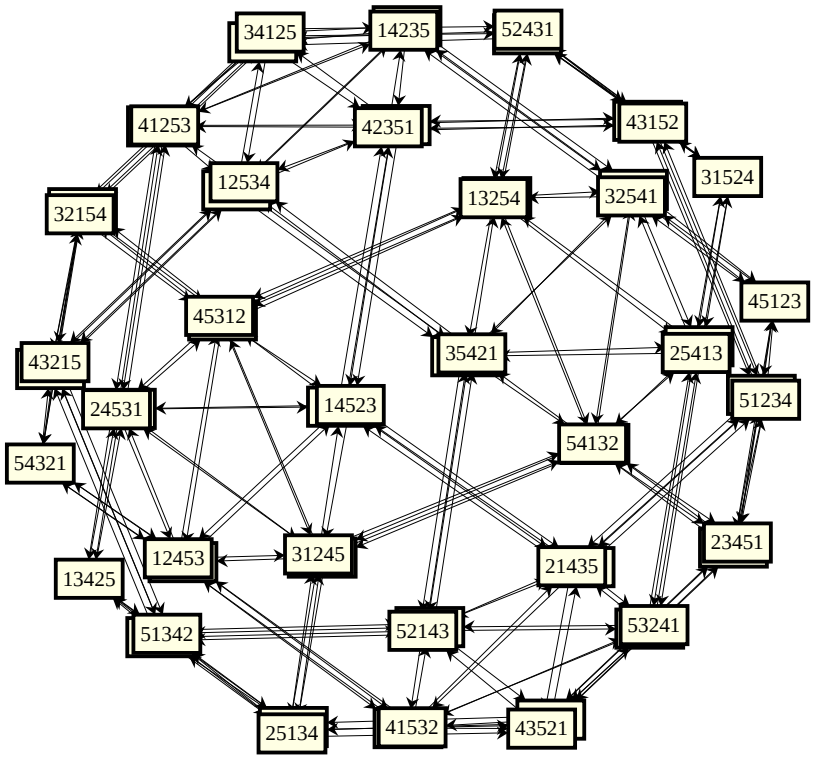

Figure 11.1.: $diameter = 6, \overline{h} \approx 4.356, N = 60, g1 = 23451, g2 = 25413$

provide. In particular, we are interested in the maximum fault tolerance that has been shown for strongly hierarchical graphs and the ease of routing. The maximum fault tolerance is shown only for strongly hierarchical Cayley graphs. Following the definition given in section 7.5.5 we construct two graphs next. The first generator is always taken as $g_1 = 23451$ for the unidirectional ring based on an AWG, and the second generator g_2 is chosen such that the resulting graph has a minimum diameter and mean hop distance for a given number N of nodes in the graph. These generators have been found by exhaustive search through the 118 remaining generators. It remains an open question if there is a way to discriminate "good" from "bad" generators, since the exhaustive search is not feasible for any prime number larger than 7.

11.3. Properties of certain graphs

The graphs in Fig. 11.1 and Fig. 11.2 are compared to ShuffleNets of the same degree 2, next. The buckyball-like graph in Fig. 11.1 may be compared to a $S_{(2,4)}$ graph with $N = 64$ nodes. The next larger ShuffleNet is the $S_{(2,5)}$ with $N = 160$, so the comparison is not really fair.

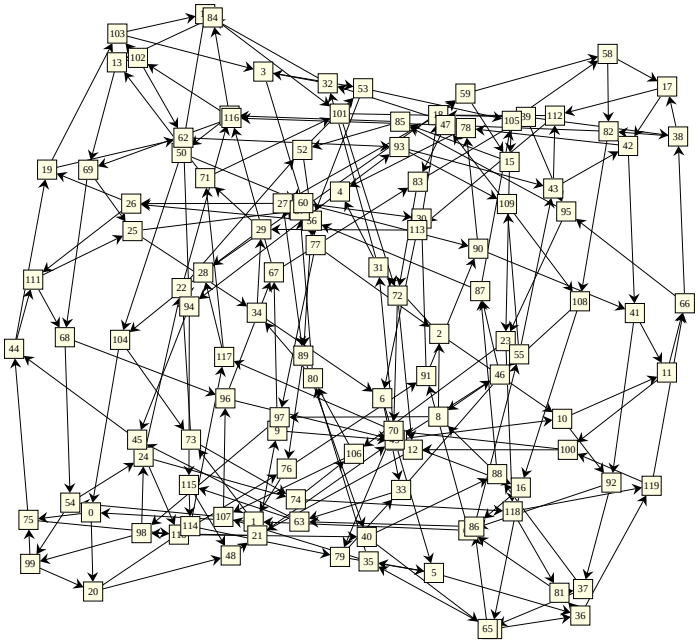

Figure 11.2.: $diameter = 8, \overline{h} \approx 5.25, N = 120, g1 = 23451, g2 = 21453,$

Name	N	$\overline{h}$	diameter	second generator
buckyball	60	4.35	6	25413
diabolo	120	5.25	8	21453
$S_{(2,4)}$	64	4.63	7	–
$S_{(2,5)}$	160	6.07	9	–

Table 11.1.: comparison of 2 Cayley graphs with 2 ShuffleNets

11.4. Conclusion

There are numerous publications on routing and fault tolerance in Cayley graphs[AK89, JM92, hSDH95]. The Cayley Graph Connected Cycles (CGCC) seem to be close to the problem that is given here with the AWG. They are, however, constructed differently. The authors propose a composition technique that takes Cayley graphs that are known to have the desired properties concerning fault tolerance and mean hop distance and replaces the nodes in the previous graph with cycles. Some of the resulting graphs are no Cayley graphs anymore, as the star connected cycle (SCC). The graphs that can be generated using the presented construction technique obviously show a low diameter and mean hop distance. The search for better "construction rules" is left for future work.

12. Conclusions

Within this dissertation, a new architecture for an optical packet network was proposed and analyzed. This architecture — PRIMENET — is based on a single AWG that connects the attached nodes in a physical star topology. A logical ring may be set up on each wavelength, iff the number of input ports (and consequently, nodes) is prime. This architecture has several advantages compared to other WDM networks. It offers a FT^r/FR^r architecture where r is the number of rings in use. This r may be scaled according to the capacity needed in the network and thus offers an easy upgrade path. The mean hop distance in the network is shown to be $\frac{N}{r+1}$ leading to a growth in network capacity that is proportional to $(r^2 + r)$.

A potential drawback of the architecture is its poor scalability. This is alleviated somehow by the concept of Cayley graphs that interconnect the PRIMENETs. Because the AWG itself can be depicted as a Cayley graph G with each wavelength being a generator g, additional generators may be introduced that connect several PRIMENETs, thereby increasing the fault tolerance in the network.

A comparison to a single–hop network based on an AWG shows that the number of fixed transmitter/receiver pairs that is needed per node to achieve the same capacity as a node equipped with tunable transceivers is rather low (3.5 in the worst case). Of course, this figure depends on certain parameters like the tuning time and range of the lasers and filters in a single–hop node, but a framework has been developed that allows to assess precisely which of the two network architectures is preferable with a given financial budget and desired network capacity.

The strategy of transmitting a segmented message over all available paths (=wavelengths) in parallel has been compared to the sequential transmission over the shortest path with the result that it may be advantageous to go for parallel transmission if the background load is low (below 6% in the case of $r = 4$ and $N = 11$). Since it is hard to estimate the background load in advance and because of the potential for packet reordering it however does not seem desirable to transmit segments in parallel. For sure, there is a lot of further work to be done in the area of load–balanced routing.

Compared to other WDM ring networks, the number of wavelengths needed to achieve the same capacity is much less, thus allowing for a higher number of nodes that could be supported with band-limited amplifiers like the EDFA.

The next step after the analysis of the network concept was to define an access protocol and a fairness algorithm. Because of the small mean hop distance in PRIMENET (given that the number of wavelengths would usually be larger than 1) the backpressure–oriented SRP-fa mechanism was chosen as a prototype. Other mechanisms that rely on reservation cycles or rotating quota assignment packets were designed to achieve global fairness and

become less useful when the traffic is more and more local. SRP-fa was adapted to a purely optical node structure based on a fiber delay line and a CSMA-based MAC protocol. A simulation model of the node and network architecture was developed in ns-2 to evaluate the performance of the fairness algorithm. In addition, to assure the statistical correctness of simulation results and to be able to do parallel simulations an interface to the AKAROA-2 tool was developed. This is documented in the Appendix.
The simulations that were performed with different wavelength topologies brought up some major conclusions:

- When designing a fairness algorithm, keep in mind that TCP will control the largest part of the traffic that will be using this node!
- Do not rely on source flows to determine the future amount of bandwidth that a node should get!
- Do not trust simulation results which were produced using unidirectional traffic models!

To go into detail, it was shown that the idea of a threshold in a transmit queue that determines if a node is congested works fine with a unidirectional traffic model. It however works against TCP´s slow start mechanism and thus may prohibit TCP connections from starting up. This problem was solved due to a shift of the decision about a node "being congested". When this decision is done using a head–of–line timer the first TCP segments will eventually be delivered and the `cwnd` of the TCP connection is allowed to open.
The second problem was that of packet reorder. Again, a simulation with a unidirectional traffic model may not even show a problem when each node just counts the bytes it received. On the other hand, TCP may fall into fast retransmit and consequently reduce its `cwnd`. This problem is obviously harder to deal with, because it is the direct result of several priority classes while only having a single packet buffer for both. Possible ways to go would be called "Making TCP robust against reorder" and "increase hardware complexity to avoid reorder". Both of these ways are discussed in brief and open up the fields of further work.
The interconnection of PrimeNets is another direction of future work. It was shown that Cayley graphs that are based on PrimeNets show values for diameter and mean hop distance that are not worse than comparable ShuffleNets. The construction of larger networks based on Cayley graphs may be a way to overcome the size limitations that are set to the PrimeNet by the number of AWG ports.

A. Performance analysis by simulation

Whenever an analytical solution is not tractable or the results that are available through analysis are too limited in their application space simulations can be performed to study the behaviour of a system. This chapter describes the simulation model that has been used to evaluate the performance of the MAC and fairness algorithm of PRIMENET. At first, the simulation tool ns-2 is introduced shortly followed by the description of work that has been done to improve the statistical security of the results of the simulation.

A.1. The network simulator ns-2 as a simulation tool

ns, the *Network Simulator*, was developed at the Lawrence Berkeley laboratory of the UC Berkeley. It is a discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing, and multi-cast protocols over wired and wireless (local and satellite) networks. Recently, quite a number of publications on network simulation have been using ns-2, the second version of ns. Due to its free availability and in particular to its huge library of simulation models for different protocols of the network and transport layers ns has become a quasi-standard in the world of network simulations. Our experiences with other simulation tools like PTOLEMY[BHLM94] or CSIM[SM92] led us to ns. The basic reason for that was that the MAC protocol would have to be C++-coded for all the tools anyway, but the higher layer protocols are available only in ns.

Ns-2 simulations are plugged together using an object-oriented version of Tcl [Wel99] called OTcl but if you need some specific behavior of the OTcl Classes you can write your own in C++. For more information about Network Simulator check the URL of the ns and nam project [BBE+99].

A.1.1. What is to be done in ns

ns's notion of a network is purely IP. Any network in ns consists of nodes and links. While this is the case for every network, ns nodes are implemented as routers, and the links as queues, which makes it difficult (but not impossible) to model circuit switched networks in ns. The whole paradigm of ns is "packets". Higher protocol layers such as TCP/UDP or traffic sources are modelled by so-called agents that are attached to the nodes.

A.1.2. LAN simulation in ns-2

Extensions have been made to ns to make simulations of local area networks possible. A great share of these extensions was contributed within the CMU Monarch roject [hMCW+98].

To be correct, the understanding of a LAN in the existing ns-2 modelling is that of a shared medium broadcast network like it is underlaying within the IEEE 802.3 Ethernet or IEEE 802.11 wireless LAN simulations. Because we are interested in exactly the portion of the network that would be covered by the `link` otherwise, it was necessary to replace the classical `simplex-link` or `duplex-link` command with another command, `new-Lan`. This OTcl-command replaces the link between two nodes with a whole stack of layer 1 and 2 protocols. These protocols are usually implemented in C++, and the new OTcl class `NetworkInterface` is used to plug the protocol classes together and to attach them to a node's `entry`.

A.2. New OTcl and C++ classes

A.2.1. OTcl classes

A.2.1.1. WDMInterface and WDMLink

These classes are defined in the `ns/tcl/lan/wdm.tcl` file.
The otcl class `WDMInterface` is an implemtentation of a *colored* interface similar to the existing class `NetworkInterface`. Basically, the stack of the Link and MAC layer protocols is instantiated and internally connected correctly here. Note that two `InterfaceQueue` (a high and low priority queue) instances are generated when the LL/SRP Link layer class is used.
The class `WDMLink` is a similar implementation to the class `VLink` of ns-2, but with the difference that the `LanRouter` class is not used here. This class usually implements a true *shared medium* between all `NetworkInterface`s of a LAN. Since the use of an AWG requires a "routing" of packets according to their wavelength, it would have been necessary to extend/color the `LanRouter` class as well, which was not considered useful.
Instead, the interconnection of the colored `WDMInterface`s is done in the `instproc makeAWG` of the `Simulator` class. This procedure has a `txlist` argument (besides the usual like bandwidth and delay) that determines the number and wavelength of the colored interfaces per node. As such, a `txlist` of [1 3] means that every node is equipped with two `WDMInterface`s, one on wavelength λ_1 and the other on wavelength λ_3. The procedure interconnects all nodes that have been created so far, since it uses the `Node_` array of the `Simulator` class. Should other nodes be needed, for instance to generate traffic into the AWG LAN, be sure to create them *after* the `makeAWG` call in the Tcl script.

A.2.2. New C++ classes

The whole stack of lower layer protocols had to be redesigned mainly to incorporate the wavelength property of packets. All classes are derived from their ns-2 base classes, i.e. PHY/SRP is derived from PHY, Mac/SRP from Mac and so on.

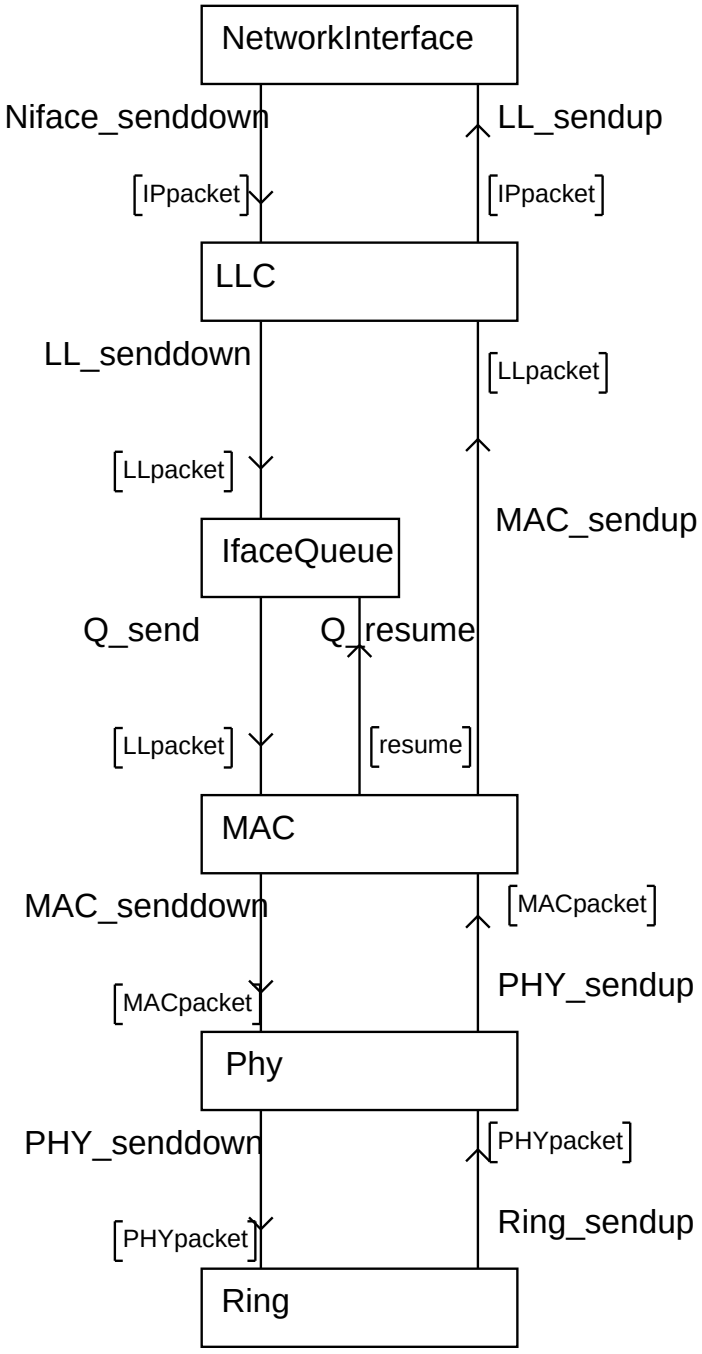

Figure A.1.: Lower layers (DLC and PHY) of the simulation model.

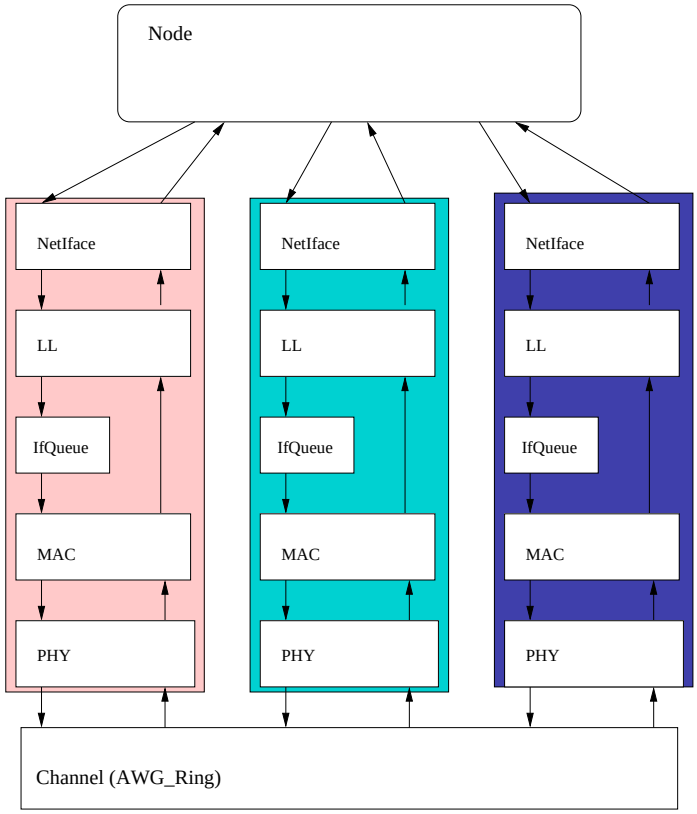

Figure A.2.: A whole protocol stack will be attached to a node for every wavelength.

A.2.2.1. The class AWG_ring

file: awg_ring.cc XXX CHANGE THE NAME to awg-ring.cc!!! This class is a very simple model of the AWG wavelength routing. It decides according to the wavelength of an incoming packet from a certain node to which node this packet has to be sent. This decision is based on channel numbers rather that real physical properties of an AWG or other wavelength router. All PHYs of a node get a copy of the packet. They decide afterwards about the correct reception. If a closer modelling of physical properties like attenuation, delay variations or co-channel crosstalk should be desired, this is the place to put it.
Up to now, all packets get delayed by the same fixed `delay_`. This is configurable from the Tcl script using the `opt(delay)` variable there.

A.2.2.2. PHY/SRP

file:phy-srp.cc This simple PHY layer model is derived from `Biconnector` and therefore has two branches: `sendUp` and `sendDown`. In the `sendDown` method, each packet is marked with its wavelength using the `init` method of the ns `class PacketStamp`. Afterwards it is given to the channel (here: `AWG_Ring`).
In the `sendUp` branch the PHY decides according to the wavelength of the packet if it should be received. If a closer modelling of interchannel crosstalk should be desired, this would have to be done here. Up to now, if the packet does not match the wavelength of the PHY, is it dropped. Else it is given to the upper layer, the MAC.

A.2.2.3. Mac/SRP

This class had the main emphasis. Although the fiber delay line (FDL) functionally is rather PHY (my excuses for the bad English), it is implemented here because of the many logical interactions between the FDL (or rather the address recognition ahead of it) and the MAC state machine. Being derived from the `BiConnector` class, the only incoming method is `recv()`. Here according to the direction of the packet it is decided where it comes from (not yet, where it goes!). If it comes from the PHY layer, it is first sent into the FDL using the `sendUp` method. The handling of the FDL is done using a special `Handler` class that is described next.
If the packet has been received from the Link Layer (LL/SRP), it is to be sent down. The first discrimination is done according to the priority of the packet. High priority packets are sent whenever the state is IDLE. If the state is not IDLE they are stored in the `retr_high` variable to be sent at the next possible instance in time (for instance when a low priority transit packet arrives, see the section A.2.2.4 for this). The `callback_high` handler that is needed to get the next packet from the HP queue is stored. Up to now it is not planned to aggregate HP packets, too.
Low priority packets, however, should be aggregated to increase bandwidth utilization. This is done as it is described in section 10.2.1. When the aggregate `Jumbo` is full or the LP queue is empty, the `Jumbo` is sent to the PHY layer in the `sendDown` method. Here, in addition, the access delay is measured using the timestamp of the packet that is set in the

`LL/SRP::sendDown` method. This access delay therefore does not include any transmission delay but is pure queueing time. Note that this time may be different for every packet in the `Jumbo`.

A.2.2.4. DelayLineSRP

The FDL is modelled as a timer handler in ns-2. This means that it basically consists of two methods, one that is called when a packet enters the delay line (`DelayLineSRP::in(Packet *p, double t)`) and the other that handles the event of the packet leaving the FDL (`DelayLineSRP::handle (Event *)`). Within the latter, the receiver side decisions are made. The first decision is if the packet is destined to the node. If it is a *usage packet*, the `recv_usage` method of the MAC for which this node is the correspondent MAC is called. If the packet is a data packet and to be received, the aggregate broken up into the individual packets that are given to the link layer.
If the packet is a transit packet, the following decisions are taken:

- if (priority == HP)
  pass the packet on

- else if (node has a HP or Usage packet buffered)
  take the packet off the ring, store it in the Recycle queue.

  - if ($Recycle.length > queue\_limit$)
    receive the packet, send it up to LL

- else
  pass the packet on, increase fd_rate

A status variable `busy_` that is set in the first method tells how many packets there are in the FDL currently. it may have the values 0 (which means that the FDL is free and the state of the MAC is `IDLE`), 1 (which means that a packet is being received, i.e. running into the FDL or that a packet is leaving the FDL with no packet following it) or 2 (which means that one packet is leaving the FDL, but there is a second packet immediately following it and just running into the FDL.) After handling an outgoing packet the next timer, `EndOfPacketSRP` is called.

A.2.2.5. Other handlers

Here it is decided according to the `busy_` variable if the other timers are called. Only if `busy_==1` the next timer `IFSHandlerSRP` is scheduled. This means that there is some additional time (the interframe space of 100 bit) before the resume() method of the MAC is called to fetch a new packet from the transmit queue, if there is one.

A.2.2.6. LL/SRP

The link layer is not functionally modeled. The class `LL/SRP` finds a correct value for the TTL field in the SRP header by a lookup in the virtual ARP table (`VARP-Hagen`) to get the MAC address of the packet's destination. After that it counts the number of hops towards the destination and sets the TTL value to one more than the counted hop number. According to the Flow ID of the packet IP header the packet is then classified into being high ($flowID > 10001$) or low priority ($flowID \leq 10001$) and sent to the corresponding queues.

A.2.3. The SRP packet

The header of the SRP packet has been shown in Fig. 10.5. For ns-2 reasons, the destination address is not defined in this `struct`, but rather taken from the MAC header of ns-2. One additional field is defined in this class, the `usage` information. This field shall only be evaluated if the packet is a *usage packet*. The last field in the `struct` is a pointer to the first packet of the aggregate. The aggregation of many SRP packets into one *Jumbo frame* is done by linking a list of the packets. Because queues and the *freelist* (!!!) are organized in the same way, all packets have to be physically copied into the aggregate rather than being re-linked.

A.3. Setup of the simulations

To get a feeling for the results that can be expected and to verify the solution for the total network capacity we derived in9.23, we performed some simulations of a small setup consisting of five nodes[1]. The number of rings (=wavelengths) may vary from 1 to (n-1) (which is 4 in this case), So we go from a unidirectional ring towards the fully meshed network. Concerning the traffic flows, there are a number scenarios of interest. We performed simulations with the [1-3] and the [1-4] scenario. The relevant parameters for the simulation were the following (if not stated otherwise in the text):

```
set AKAROA 1
Mac/SRP set debug_ 0

set opt(num)    5 ;# the number of nodes
set opt(chan)   Channel/AWG_Ring
set opt(tcp)    TCP/FullTcp
set opt(sink)   TCP/FullTcp
set opt(packet) 9216 ;# the packet length
set opt(header) 20 ;# the header length in byte
set opt(window) 64 ;# the TCP advertised window in segments
```

[1] after all, what we need is a prime number, the next would be 7, but we do not expect substantially different numbers there.

```
set opt(app)    Application/Traffic/Trace
set opt(bw)     622000000.0 ;# 622 Mbit/s raw line rate
set opt(delay)  0.0002; # ~40 km distance between two nodes
set opt(mac)    Mac/SRP
set opt(ll)     LL/SRP
set opt(phy)    Phy/SRP
set opt(qsize)  100 ;# the Interface queue length
set opt(tr)     /tmp/out
set opt(ifq)    Queue/DropTail
set opt(stop)   1
set opt(IFS)    100 ;# the Interframe Space in bits
$opt(ifq) set limit_    $opt(qsize)

WDMIface set llType_    $opt(ll)
WDMIface set ifqType_   $opt(ifq)
WDMIface set macType_   $opt(mac)
WDMIface set phyType_   $opt(phy)

set FACTOR_0 [lindex $argv 0]
set FACTOR_1 [lindex $argv 1]

set tfile [new Tracefile]
$tfile filename /home/horst/ns/ns-2.1b7a/results/traces/CBR$FACTOR_0

Mac/SRP set DECAY_INTERVAL $FACTOR_1
Mac/SRP set packetsize $opt(packet)
Mac/SRP set bandwidth_ $opt(bw)
Mac/SRP set congestedL 50 ;# has no effect in HOL-timer based MAC!
Mac/SRP set MAX_USAGE $opt(bw)
Mac/SRP set LP_MY_USAGE [expr 1/(1-pow((511.0/512),[Mac/SRP set DECAY_INTERVAL]))]
Mac/SRP set LP_FD_RATE [expr 1/(1-pow((63.0/64),[Mac/SRP set DECAY_INTERVAL]))]
Mac/SRP set LP_ALLOW [expr 1/(1-pow((63.0/64),[Mac/SRP set DECAY_INTERVAL]))]
Mac/SRP set AGECOEFF [expr 1/(1-pow((3.0/4),[Mac/SRP set DECAY_INTERVAL]))]
Mac/SRP set MAX_LINE_RATE  [expr [Mac/SRP set DECAY_INTERVAL]*[Mac/SRP set AGECOEFF]]
Mac/SRP set ifs_       $opt(IFS)
set delayline_ [expr [expr [expr 8 * $opt(packet)]+$opt(IFS)-10]/$opt(bw)]
# reduce the length of the delay line to avoid synchronous events
# of packets leaving and entering the FDL at the same time
Mac/SRP set delayline_ $delayline_
set ns [new Simulator]
$ns rtproto Manual
lappend observe_list 0 2 7 9
set numconnections [llength $observe_list]
```

The following line starts e.g. a [1-3] simulation script (`trace_srp.tcl`) with a load of 600 Mbit/s per node and a $DECAY\_INTERVAL = 1$:

```
ns trace_srp.tcl 600M 1 1 3
```

As it can be guessed from the preceding script, the first argument is the name of the trace file, the second in the $DECAY\_INTERVAL$ and the following arguments are treated as wavelengths. The output would be similar to this:

```
warning: no class variable Tracefile::debug_

        see tcl-object.tcl in tclcl for info about this warning.

max_line_rate = 4.0
agecoeff = 4.0
lp_my_usage = 512.0
lp_allow = 64.0
delayline is 0.00011867845659163987
link delay is 0.0002
Wellenlängen: 1 3
AWG_Ring: ring setup hier!
numparams= 8
MacSRP: 0 akaroa 1 activated!
MacSRP: 2 akaroa 3 activated!
MacSRP: 7 akaroa 5 activated!
MacSRP: 9 akaroa 7 activated!
Param    Estimate       Delta  Conf         Var   Count   Trans
    1  0.00124674 9.31057e-05  0.90 2.84261e-09   49200     656
    2 2.82653e+08 4.11725e+06  0.90 5.55879e+12   18513    2057
    3 0.000985264 7.77178e-05  0.90 1.98064e-09   48024    2001
    4 2.78626e+08 4.56925e+06  0.90 6.84629e+12   18585    2065
    5 0.000294936 0.000129059  0.90 5.46184e-09   27720    1155
    6 2.24494e+08 9.95577e+06  0.90 3.25024e+13   22380    1492
    7  0.00181222 0.000130881  0.90 5.61714e-09   51084    4257
    8 3.00766e+08  1.3318e+07  0.90 5.81626e+13   22320    1488
    9 2.21815e+08 3.71772e+06  0.90 4.53229e+12    2256     376
   10 2.21197e+08 3.96406e+06  0.90 5.15283e+12    2178     363
   11 1.32555e+08 1.13916e+07  0.90 4.25538e+13    2256     376
   12 2.44023e+08 9.74234e+06  0.90 3.11237e+13    2874     479
#Results obtained using the Akaroa2(c) automatic parallel simulation manager.
horst@jochn:~/ns/ns-2.1b7a>
```

In the first cutout from the script it could be seen in the last rows that MACs no. 0,2,7, and 9 should be observed. Here one can see the output of AKAROA-2. The 90% confidence level has been reached for 12 variables, which represent the mean access delay for each

segment and the (brutto) throughput in Mbit/s for each of the four MACs in the first 8 lines. The next 4 lines give the net throughput. These are printed from the sink of the TCP connection in tcl rather than the C++ MAC layer. This is a good example for the use of Akaroa-2, which will be explained in the next chapter.

A.4. Load models

A.4.1. CBR traffic

The finding in [CMT98] was that although only about 10% of the packets are 1500 byte long, but produce 50% of the data bytes. So three `ExpOO` sources of the ns-2 distribution generating packet lengths of 1500, 576, and 40 bytes make up one flow. The `ExpOO` traffic agent generates bursts of 0.001s followed by idle periods of the same length. Therefore, the load parameter `$FACTOR_0` that is a settable parameter has to be multiplied with the appropriate factors to produce a mix of bursts of load `$FACTOR_0`.

```
set source0 [attach-expoo-traffic $Node_(0) $sink_(0) \\
             1500 0.001s 0.001s [expr 1.0* $FACTOR_0]]
$source0 set fid_ 1
set source1 [attach-expoo-traffic $Node_(0) $sink_(0) \\
             576 0.001s 0.001s [expr 0.6 * $FACTOR_0]]
$source1 set fid_ 1
set source2 [attach-expoo-traffic $Node_(0) $sink_(0) \\
             40 0.001s 0.001s [expr 0.4* $FACTOR_0]]
$source2 set fid_ 1
```

A.4.2. Packet length traces

The load generated for the unidirectional traffic was read from the "enqueue" event at the interface queue of node 0 (marked with "+" in the first column of the trace file). Between 90000 and 120000 packet arrival events were recorded in a file. The ns-2 class `traffictrace` reads the packet arrival events from the file with different (random) start points per node.

B. Parallel and distributed simulations with ns-2 and Akaroa-2

Within the last decade, discrete simulation has become a standard tool of scientists and engineers wherever it was necessary to estimate the behavior of large stochastic systems like computer networks or national economies. As can be seen in the number of contributions to scientific journals or conferences, almost every result that has been derived analytically is nowadays justified or verified by simulation curves that support the thesis of the author. Although quantitative stochastic simulation is a useful tool for studying performance of stochastic dynamic systems, it can consume much time and computing resources. To overcome these limits, parallel or distributed computation is needed.
Universities and research institutes often have lots of computers connected in a LAN. Using these heterogenous (in terms of speed) computers for simulations is a straightforward idea. One of the main obstacles is the easy distribution of simulations over this cluster. There are two approaches to solve this problem. One is the explicit parallelization of the simulation and the other one is the MRIP-approach taken by Akaroa-2. It runs *Multiple Replications In Parallel*. This results in an almost linear speedup with the number of hosts. NS-2 is a nice tool for network simulation, but does not provide support for statistical analysis of the obtained results. So by combining NS-2 and Akaroa-2 we add run-length control for simulations based on statistic measures to NS-2 as well as a speed-up if the simulation can be run on many hosts in parallel. The existing package for writing parallel simulations with ns is – at least in our eyes – a bit more complicated and does not provide statistical run length control. The changes were made only in the NS-2 package. The original code seems to be quite stable, so these changes should not depend too much on the NS-2 version used. Ns-2's main advantage is the multitude of network protocols implemented. On the other hand it lacks support for statistical evaluations. Usually one writes the interesting variables into a trace file, and measures such as mean and variance are evaluated with an awk-script. But how many values should be written into the trace file? Sometimes simulations are run much longer than necessary – or more often much shorter, which devaluates the conclusions drawn out of the simulations. To get an expression of the quality of any simulation or measurement result. Another problem concerns the length of the random number generator. Some simulations need less than a day to exhaust the random number stream.

B.1. Statistical Security

In order to ensure that the made predicates have a statistical security, it is necessary to control the temporal duration of the simulation. For this purpose AKAROA-2 records

results and calculates the half–length H of the confidence interval. The half–length is the momentary deviation from the current average value.

Given a number of values x_n then the mean value is:

$$\bar{x}(n) = \frac{\sum^n x_i}{n} \tag{B.1}$$

and the variance of the mean values $\bar{x}(n)$ is:

$$\sigma_n^2 = \frac{\sum(x_i - \bar{x}(n))^2}{n-1} \tag{B.2}$$

Thus the half–length H of the confidence interval, which indicates the precision of the mean value, which is to be situated within statistical security z,

$$H = z \cdot \sqrt{\frac{\sigma_n^2}{n}} \tag{B.3}$$

whereby H is situated in both negative and positive direction of the mean value $\bar{x}(n)$.

$$[\bar{x} - H, \bar{x} + H] \tag{B.4}$$

In the above notation σ_n^2 is the variance and n the number of simulation values. The value z corresponds to a certain safety limit of the confidence interval that can be stated. The corresponding confidence level values of z can be taken from a table [Jai91].

confidence level	z
0,90 %	1,645
0,95 %	1,966
0,99 %	2,576

B.1.1. Akaroa-2

Akaroa-2 is designed for running quantitative stochastic discrete-event simulations on Unix multiprocessor systems or networks of heterogeneous Unix workstations. Instead of dividing up the program for parallelization – a tremendous effort – multiple instances of an ordinary simulation program are run simultaneously on different processors.

For more information about Akaroa check the URL of the AKAROA-II project. [Paw01].

Akaroa-2 is a process oriented simulation engine. Written completely in C++ it is easy to write fast simulations with it. The random number generator used in the current release has a really long period. It would take decades to exhaust it. The main disadvantage is the lack of network protocols. It uses statistical run length control, and the observed variables are summarized in two statistics, mean and variance, which is enough for most purposes. If you need some more you can write your own class.

Akaroa-2 does the parallelization of the simulation in a client-server manner. A unique process called `akmaster` runs on one machine in the cluster and controls the execution of

the simulations using a number of `akslave` processes. These have to be started beforehand on every machine that shall be used to run simulations on it. The command to run the simulations is e.g.:

```
akrun -n 5 ns mm1.tcl
```

The value of n=5 means to start the simulation on five machines, whereas the simulation to be executed is given after. Eventually, a result in the form of

Param	Estimate	Delta	Conf	Var	Count	Trans
1	97.6467	0.400812	0.95	0.0405465	264078	1259
2	2.77603	0.134357	0.95	0.0045561	263346	1673

should appear.

B.2. Interface internals

During the project we experimented with several implementations. The first one was the most conservative and was guarded against a lot of errors that simply don't occur. Although this was useful to learn about the structure of NS-2 and Akaroa-2 the overhead was unnecessary and following versions were much simpler.

We ended with a new file `akaroa.cc` and some changes in the existing files of NS-2 `rng.h` and `rng.cc`. You will have to install all these files in order to use the Akaroa-NS-interface.

B.2.1. Call mapping

In the file `akaroa.cc` the complete Akaroa-NS-interface is defined. The interface consists of the new class `Akaroa`, which is derived from `TclObject`. The class `AkaroaClass` provides the interface for Tcl to create C++-objects. When an `Akaroa`-method is called from OTcl, the complete OTcl string is passed to the `Akaroa`-method `command`. In this method the string is evaluated by simple string comparisons and the appropriate library function is called.

B.2.2. Random Number Generator

When running multiple replications of simulation model in parallel, it is important that each simulation engine uses a unique stream of random numbers. So we had to change the Random Number Generator.

The new class `AkRNGImplementation` was derived. It is also contained in the file `akaroa.cc`. It maps calls for a new uniformly distributed random number to the Akaroa Random Number Generator. Additionally some initializations are performed.

The original `RNGImplementation` of NS-2 was not intended to be inherited, so changes `rng.h` and `rng.cc` became necessary.

The class `RNGImplementation` has now a `virtual long next()` and a `virtual long next_double()` methods. This class and `AkRNGImplementation` don't belong to the public interface, but are used only internal. The "official" interface to the random number generator is the class `RNG`. We added the `static void setRNGImplementation(RNGImplementation`

`*imp)` method, to set a new RNGImplementation when needed. The `Akaroa`-class uses it to install its own implementation. Due to its dynamic nature the random number stream of `RNG` is a pointer now. We added reference counting for the `stream` to ensure that there is only one `RNGImplementation`-object.

It is impossible to use more than one random number stream, even multiple `RNG`-objects would use only one. This is a restriction if you want to use NS-2 without `Akaroa`.

The seeds which can be set and obtained are useless with `Akaroa`. Seeds are managed centrally by the `akmaster`-process.

B.3. Acronyms

ATM Asynchronous Transfer Mode

AWG Arrayed Waveguide Grating A passive wavelength router.

BER bit error rate

CBR Contant Bit Rate

CSMA Carrier Sense Multiple Access

DXC Digital crossconnects A DXC multiplexes and switches SONET/SDH connections.

EDFA Erbium-doped Fiber Amplifier

FDL Fiber Delay Line

GMPLS Generalized Multi-Protocol Label Switching

HOL Head-Of-Line

IP Internet Protocol

ITU International Telecommunication Union The former CCITT.

MAC Medium Access Control

MAPOS Multiple Access Protocol Over SONET

MPLS Multi-Protocol Label Switching

OPS Optical Packet Switching

OBS Optical Burst Switching

ONA Optical Network Adapter

OTDM Optical Time Division Multiplexing

OXC Optical crossconnects

PoS Packet over SONET

PSC Passive Star Coupler

PXC Photonic Crossconnect

QoS Quality of Service

RAM Random Access Memory

RPR Resilient Packet Ring The IEEE 802.17 working group is working on this standard for PHY and MAC of a dual optical packet ring.

RWA Routing and Wavelength Assignment

SDH Synchronous Digital Hierachy

SOA Semiconductor Optical Amplifier

SONET Synchronous Optical Network

SRP Spatial Reuse Protocol

TCP Transmission Control Protocol

UDP User Datagram Protocol

WAN Wide Area Network

WDM Wave Division Multiplexing

WRN Wavelength Routed Networks

WWW World Wide Web

Bibliography

[ABF01] M. Allman, H. Balakrishnan, and S. Floyd. RFC 3042: Enhancing TCP's Loss Recovery Using Limited Transmit, January 2001.

[Aca87] A. S. Acampora. A multichannel multihop local lightwave network. In *Proc. IEEE Globecom '87*, pages 1459–1467, Nov. 1987.

[AK89] S.B. Akers and B. Krishnamurthy. A group theoretic model for symmetric interconnection networks. *IEEE Transaction on Computers*, C-38(4):555–566, April 1989.

[AMA+99] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. Requirements for traffic engineering over mpls, 1999.

[APS99] M. Allman, V. Paxson, and W. Stevens. RFC 2581: TCP Congestion Control, April 1999.

[AR01] Daniel Awduche and Yakov Rechter. Multiprotocol lambda switching: Combining mpls traffic engineering control with optical crossconnects. *IEEE Communcations Magazine*, 39(3):111–116, March 2001.

[AS91] A. S. Acampora and S. I. A. Shah. Multihop lighwave networks: A comparison of store–and–forward and hot–potato routing. In *Proc., IEEE INFOCOM*, pages pp. 10–19, 1991.

[Atk94] R. Atkinson. RFC 1626: Default IP MTU for use over ATM AAL5, May 1994. Obsoleted by RFC2225 [LH98]. Status: PROPOSED STANDARD.

[AVR+02] A.Carena, V.Ferrero, R.Gaudino, V.De Feo, F.Neri, and P.Poggiolini. Ringo: a demonstrator of wdm optical packet network on a ring topology. In *ONDM 2002 Conference Proceedings*, February 2002.

[BA02] Ethen Blanton and Mark Allman. On Making TCP More Robust to Packet Reordering. *ACM Computer Communications Review*, 32(1):20–29, January 2002.

[BBE+99] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Shurma, Kannan Varadhan,

Ya Xu, haobo Yu, and Daniel Zappala. Improving Simulation for Network Research, March 1999.

[BDL+01] Ayan Banerjee, John Drake, Jonathan Lang, Brad Turner, Daniel Awduche, Lou Berger, Kireeti Kompella, and Yakov Rekhter. Generalized multiprotocol label switching: An overview of signaling enhancements and recovery techniques. *IEEE Communications Magazine*, 39(7):144–151, July 2001.

[BGRS98] R.-P. Braun, G. Grosskopf, D. Rohde, and F. Schmidt. Low-phase-noise millimeter-wave generation at 64 ghz and data transmission using optical sideband injection locking. *IEEE PHOTONICS TECHNOLOGY LETTERS*, VOL. 10(NO. 5):pp. 728–730, MAY 1998.

[BHLM94] Joseph Buck, Soonhoi Ha, Edward A. Lee, and David G. Messerschmitt. Ptolemy: A framework for simulating and prototyping heterogenous systems. *Int. Journal in Computer Simulation*, 4(2):0–, 1994.

[Big74] N. Biggs. *Algebraic Graph Theory*. Cambridge Univ. Press, Cambridge, 1974.

[BJB+97] M. S. Borella, J. P. Jue, D. Banerjee, B. Ramamurthy, and B. Mukherjee. Optical components for WDM lightwave networks. *Proceedings of the IEEE*, vol. 85:pp. 1274–1307, August 1997.

[BJM99] M. S. Borella, J. P. Jue, and B. Mukherjee. Simple scheduling algorithms for use with a waveguide grating multiplexer based local optical network. *Photonic Network Commun.*, 1(1), 1999.

[BJS99] S. Banerjee, V. Jain, and S. Shah. Regular multihop logical topologies for lightwave networks. *IEEE Communication Surveys*, 1(1):2–18, First Quarter 1999.

[BM96] D. Banerjee and B. Mukherjee. A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. *IEEE Journal on Selected Areas in Communications*, 14(5):902–908, May 1996.

[BP02] Patrick L. Barry and Dr. Tony Phillips. Sit. speak. good photon! http://science.nasa.gov/headlines/y2002/27mar_stoplight.htm, March 2002.

[BPS99] Jon C. R. Bennett, Craig Partridge, and Nicholas Shectman. Packet reordering is not pathological network behavior. *IEEE/ACM Transactions on Networking*, 7(6):789–798, Dec. 1999.

[Bra96] C. Brackett. "foreword, is there an emerging consensus on wdm networking?". *IEEE J. Lightwave Technologies*, vol. 14:pp. 936–941, June 1996.

[BRPS02] Ilia Baldine, George N. Rouskas, Harry G. Perros, and Dan Stevenson. Jumpstart: A just-in-time signaling architecture for wdm burst-switched networks. *IEEE Communications*, February 2002. (to appear).

[BT94] T. Brown and K. Tesink. RFC 1595: Definitions of managed objects for the SONET/SDH interface type, March 1994. Status: PROPOSED STANDARD.

[CAI] CAIDA. packet size distribution. http://www.caida.org/analysis/AIX/plen_hist/.

[CB97] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, December 1997.

[CBP93] K. Claffy, H.-W. Braun, and G. Polyzos. Long-term traffic aspects of the nsfnet. In *Proceedings of INET'93*, 1993.

[CCO93] Jeane S.-C. Chen, Israel Cidon, and Yoram Ofek. A local fairness algorithm for gigabit LANs/MANs with spatial reuse. *IEEE Journal on Selected Areas in Communications*, 11(8):1183–1191, August 1993.

[CL99] D.G. Cunningham and W.G. Lane. *Gigabit Ethernet Networking*. Macmillan Technical Publishing, Indianapolis IN, 1999.

[CLHVM00] J. Carlson, P. Langner, E. Hernandez-Valencia, and J. Manchester. PPP over simple data link (SDL) using SONET/SDH with ATM-like framing. IETF RFC 2823, May 2000.

[CMT98] K. Claffy, Greg Miller, and Kevin Thompson. the nature of the beast: recent traffic measurements from an internet backbone. In *Proceedings of INET'98*, 1998.

[CO93] I. Cidon and Y. Ofek. MetaRing – a full duplex ring with fairness and spatial reuse. *IEEE Transactions on Communications*, 41(1):110–119, January 1993.

[Con96] AON Consortium. Slides on WDM. http://www.ll.mit.edu/aon/WDMSlide38.html, 1996.

[DGSB00] Klaus Dolzer, Christoph Gauger, Jan Späth, and Stefan Bodamer. Evaluation of reservation mechanisms for optical burst switching. Technical Report No.35, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (IND), 2000.

[DH98] S. Deering and R. Hinden. RFC2460: Internet Protocol, Version 6 (IPv6) Specification, December 1998.

[DR00] Rudra Dutta and George N. Rouskas. A survey of virtual topology design algortihms for wavelength routed optical networks. *Optical Networks Magazine*, 1(1):73–89, January 2000.

[Dra91] C. Dragone. An nxn optical multiplexor using a planar arrangement of two star couplers. *IEEE Photonic Technology Letters*, pages 812–15, 1991.

[DY01] Robert Doverspike and Jennifer Yates. Challenges for mpls in optical network restoration. *IEEE Communcations Magazine*, 39(2):89–95, February 2001.

[ea98] P. Gambini et al. Transparent Optical Packet switching: Network architecture and demonstrators in the KEOPS project. *IEEE J. Select. Areas Communication*, 16(7):1245–1249, Sept. 1998.

[FMMP00] Sally Floyd, Jamshid Mahdavi, Matt Mathis, and Matt Podolsky. RFC 2883: An extension to the selective achmowledgement (sack) option for TCP, July 2000.

[GA96] Duanyang Guo and A. Acampora. Scalable multihop WDM passive Ring with optimal wavelength Assignment and adaptive wavelength routing. *Journal of Lightwave Technology*, vol. 14(no. 6):pp. 1264–1277, June 1996.

[GGA95] D. Guo, Wei Guo, and A. Acampora. Shufflenet = hypercube x ring and embedding shufflenet on mesh network. In *Proc. IEEE Globecom '95*, pages 1762–1766, Nov. 1995.

[GGH+98] Lutz Giehmann, Andreas Gladisch, Norbert Hanik, Olaf Ziemann, and Joachim Rudolph. "the application of code-division multiple access for transport overhead information in transparent optical networks". In *Proceedings of OFC 1998, San Jose*, 1998.

[Gre92] P.E. Green. *Fiber Optics Communication Networks*. Prentice Hall, 1992.

[hMCW+98] h Monarch, P. CMU, M. Wireless, M. to, N. Available, f www, and m cmu. The cmu monarch project's wireless and mobility extensions to ns, 1998.

[hSDH95] Sabine R. Öhring, Falguni Sarkar, Sajal K. Das, and Dirk H. Hohndel. Cayley Graph Connected Cycles - A new class of Fixed Degree Interconnection Networks. In *Proc. of the 28th Annual Hawaii International Conference on System Sciences – 1995*, pages pp. 479–488. IEEE, 1995.

[Inc00] Corning Inc. Metrocor product information (pi1302), October 2000.

[Jai91] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, Inc., New York, NY, 1991.

[JB88] V. Jacobson and R. T. Braden. RFC 1072: TCP extensions for long-delay paths, October 1988. Obsoleted by RFC1323 [JBB92]. Status: UNKNOWN.

[JBB92] V. Jacobson, R. Braden, and D. Borman. RFC 1323: TCP extensions for high performance, May 1992. Obsoletes RFC1072, RFC1185 [JB88, JBZ90]. Status: PROPOSED STANDARD.

[JBZ90] V. Jacobson, R. T. Braden, and L. Zhang. RFC 1185: TCP extension for high-speed paths, October 1990. Obsoleted by RFC1323 [JBB92]. Status: EXPERIMENTAL.

[JM92] Zoran Jovanovic and Jelena Misic. Fault tolerance of the star graph interconnection network. 1992.

[JM98] Jason B. Jue and B. Mukherjee. Multiconfiguration multihop protocols (MMPs): A new class of protocols for packet-switched WDM optical networks. In *Proceedings of INFOCOM '98*, April 1998.

[Jue01] J.P. Jue. *Advances in Optical Networks*, chapter An Overview of Lightpath Establishment in Wavelength-Routed WDM Optical Networks. Kluwer Academic Publishers, 2001.

[KA98] Ezhan Karasan and Ender Ayanoglu. Effects of wavelength routing and selection algorithms on wavelength conversion gain in wdm networks. *IEEE/ACM Transactions on Networking*, 6(2):186–196, April 1998.

[Lau94] M. Laubach. RFC 1577: Classical IP and ARP over ATM, January 1994. Obsoleted by RFC2225 [LH98]. Status: PROPOSED STANDARD.

[LG02] Michael Laor and Lior Gendel. The Effect of Packet Reordering in a Backbone Link on Application Throughput. *IEEE Network*, pages 28–36, September/October 2002.

[LH98] M. Laubach and J. Halpern. RFC 2225: Classical IP and ARP over ATM, April 1998. Obsoletes RFC1626, RFC1577 [Atk94, Lau94]. Status: PROPOSED STANDARD.

[LK00] Reiner Ludwig and Randy Katz. The eifel algorithm: Making tcp robust against spurious retransmissions. *ACM Computer Communications Review*, 30(1), January 2000.

[Lo98] Selina Lo. Jumbo frames? Yes!
http://www.nwfusion.com/forum/0223jumboyes.html, Feb. 1998.

[LS99] L. Li and A. K. Somani. Dynamic wavelength routing using congestion and neighborhood information. *IEEE/ACM Transactions on Networking*, 7(5):779–786, May 1999.

[MADD98] J. Manchester, J. Anderson, B. Doshi, and S. Dravida. IP over SONET. *IEEE Communications Magazine*, pages 136–142, May 1998.

[Max85] N. F. Maxemchuck. Regular Mesh Topologies in Local and Metropolitan Area Networks. *AT&T Techn. Journal*, 64:1659–1686, Sept.1985.

[MCN97] M. Ajmone Marsan, C. Casetti, and F. Neri. The fairness issue in the crma-ii mac protocol. *Computer Networks and ISDN Systems*, 29(6):pp.653–673, May 1997.

[MD90] J. C. Mogul and S. E. Deering. RFC 1191: Path MTU discovery, November 1990. Obsoletes RFC1063 [MKPM88]. Status: DRAFT STANDARD.

[MKPM88] J. C. Mogul, C. A. Kent, C. Partridge, and K. McCloghrie. RFC 1063: IP MTU discovery options, July 1988. Obsoleted by RFC1191 [MD90]. Status: UNKNOWN.

[MM97] K. Murakami and M. Maruyama. RFC 2173: A MAPOS version 1 extension — node switch protocol, June 1997. Status: INFORMATIONAL.

[MP85] J. C. Mogul and J. Postel. RFC 950: Internet Standard Subnetting Procedure, August 1985. Updates RFC0792 [Pos81c]. See also STD0005 . Status: STANDARD.

[MRW00] M. Maier, M. Reisslein, and A. Wolisz. High performance switchless wdm network using multiple free spectral ranges of an arrayedwageguide grating. In *Terabit Optical Networking: Architecture, Control, and Management Issues*, volume vol. 4213, page 101. SPIE, November 2000.

[MRW02] M. Maier, M. Reisslein, and A. Wolisz. "towards efficient packet switching metro wdm networks". *Optical Networks Magazine (Special Issue on Optical Packet Switching Networks)*, vol. 3(no. 6):pp. 44–62, November 2002.

[MSMO97] Matthew Mathis, Jeff Semke, Jamshid Mahdavi, and Teunis Ott. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *Computer Communication Review*, 27(3), July 1997.

[Muk92a] B. Mukherjee. WDM-based local lightwave networks - part I single hop systems. *IEEE Network*, 6:12–27, May 1992.

[Muk92b] B. Mukherjee. WDM-based local lightwave networks - part II multihop systems. *IEEE Network*, 6:20–32, July 1992.

[Muk97] Biswanath Mukherjee. *Optical Communication Networks*. McGraw Hill, 1997.

[NEH+96] P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. Ching Liaw, T. Lyon, and G. Minshall. RFC 1953: Ipsilon Flow Management Protocol Specification for IPv4 version 1.0, May 1996. Status: INFORMATIONAL.

[NML98] Peter Newman, Greg Minshall, and Thomas L. Lyon. Ip switching – atm under ip. *IEEE/ACM Transactions on Networking*, 6(2):117–129, April 1998.

[Ogu96] Kimio Oguchi. New notations based on the wavelength transfer matrix for functional analysis of wavelength circuits and new networks using awg-based star coupler with asymmetric characteristics. *Journal of Lightwave Technology*, 14(6):1255–1263, June 1996.

[Pap00] Cisco White Paper. Cisco ip/tv and qos: How to enable ip precedence on an ip/tv server for use with qos policy. http://www.cisco.com/warp/public/cc/pd/mxsv/iptv3400/tech/ipqos_wp.htm, October 2000.

[Paw01] K. Pawlikowski. Project akaroa. http://www.cosc.canterbury.ac.nz/research/RG/net_sim/simulation_group/akaroa/about.chtml, Feb. 2001.

[Pet02] Prof. K. Petermann. Einführung in die optische Nachrichtentechnik. Skript zur Vorlesung, TU Berlin, Institut für Hochfrequenztechnik, 2002.

[PONJ99] J.J.O. Pires, M. O'Mahony, N.Parnis, and E. Jones. Size limitations of a WDM ring network based on Arrayed-Waveguide Grating OADMs. In Maurice Gagnaire and H. van As, editors, *Proceedings of the Third IFIP ONDM Conference*, pages 71–78, February 1999.

[Pos80a] J. Postel. RFC 760: DoD standard Internet Protocol, January 1980. Obsoleted by RFC0791, RFC0777 [Pos81b, Pos81a]. Obsoletes IEN123. Status: UNKNOWN. Not online.

[Pos80b] J. Postel. RFC 768: User datagram protocol, August 1980. Status: STANDARD. See also STD0006.

[Pos81a] J. Postel. RFC 777: Internet Control Message Protocol, April 1981. Obsoleted by RFC0792 [Pos81c]. Obsoletes RFC0760 [Pos80a]. Status: UNKNOWN. Not online.

[Pos81b] J. Postel. RFC 791: Internet Protocol, September 1981. Obsoletes RFC0760 [Pos80a]. See also STD0005. Status: STANDARD.

[Pos81c] J. Postel. RFC 792: Internet Control Message Protocol, September 1981. Obsoletes RFC0777 [Pos81a]. Updated by RFC0950 [MP85]. See also STD0005 . Status: STANDARD.

[Pos81d] J. Postel. RFC 793: Transmission control protocol, September 1981. See also STD0007 . Status: STANDARD.

[Pos81e] J. Postel. RFC 795: Service mappings, September 1981. Status: UNKNOWN. Not online.

[PT94] Jon M Peha and Fouad A. Tobagi. Analyzing the Fault Tolerance of Double-Loop Networks. *IEEE/ACM Transactions on Networkin1g*, vol. 2(No. 4):pp.363–373, 1994.

[QY99] C. Qiao and M. Yoo. Optical burst switching (OBS) - a new paradigm for an optical internet. *J. High Speed Networks (JHSN)*, vol. 8(no. 1):pp. 69–84, 1999.

[(RE96] ETSI Radio Equipment and Systems (RES). Radio equipment and systems (res) high performance radio local networks (HIPERLANs) type 1 functional specification, 1996.

[RS98] R. Ramaswami and K.N. Sivarajan. *Optical Networks A Practical Perspective*. Morgan Kaufmann Publishers, San Francisco, 1998.

[Sab68] G. Sabidussi. Vertex transitive graphs. *Monatshefte Mathematik*, 1968.

[SAS96] S. Subramaniam, M. Azizoglu, and A. K. Somani. All-optical networks with sparse wavelength conversion. *IEEE/ACM Transactions on Networking*, 4:544–557, August 1996.

[SCT01] John Strand, Angela L. Chiu, and Robert Tkach. Issues for routing in the optical layer. *IEEE Communcations Magazine*, 39(2):98–104, February 2001.

[Set98] Pisai Settawong. A fair control mechanism with qos guarantee support for dual ring lans/mans. Master's thesis, University of Tokio, Dept. of Frontier Informatics, 1998.

[SH99] R. Schoenen and R. Hying. Distributed cell scheduling algorithms for virtual-output-queued switches, 1999.

[She91] Tim Shepard. TCP Packet Trace Analysis. Technical Report TR-494, Massachusetts Institute of Technology, February 1991.

[Sim94] W. Simpson. RFC 1619: PPP over SONET/SDH, May 1994. Status: PROPOSED STANDARD.

[Sim99] W. Simpson. PPP over SONET/SDH. IETF RFC 2615, June 1999.

[SM92] H. Schwetman and R. Manual. Microelectronics and computer technology corp, 1992.

[Smi88] M. K. Smit. New focussing and dispersive planar component based on optical phased array. *Electronic Letters*, 24(7), March 1988.

[Soc97] IEEE Computer Society. Wireless lan medium access control, 1997.

[SR94] Kumar N. Sivarajan and Rajiv Ramaswami. Lightwave networks based on de Bruijn graphs. *IEEE/ACM Transactions on Networking*, 2(1):70–79, 1994.

[SSW+00] Kapil Shrikhande, A. Srivatsa, I. M. White, M. S. Rogge, D. Wonglumsom, S. M. Gemelos, and L.G. Kazovsky. CSMA/CA MAC protocols for IP-HORNET: An IP over WDM metropolitan area ring network. In *Proc. of Globecom 2000*, 2000.

[Ste94] W. R. Stevens. *TCP/IP Illustrated, Volume 1.* Addison-Wesley, 1994.

[Ste97] W. Stevens. RFC 2001: TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms, January 1997. Status: PROPOSED STANDARD.

[Tan94] K. Wendy Tang. Cayleynet: A multihop wdm-based lightwave network. In *Proc. of INFOCOM 1994: Toronto, Ontario, Canada - Volume 3*, pages pp.1260–1267, 1994.

[TIIN96] Y. Tachikawa, Y. Inoue, M. Ishii, and T. Nozawa. Arrayed-waveguide grating multiplexer with loop-back optical paths and its applications. *Journal of Lightwave Technology*, 14(6):977–984, June 1996.

[TMW97] Kevin Thompson, Gregory J. Miller, and Rick Wilder. Wide-area internet traffic patterns and characteristics. *IEEE Network*, November/December 1997.

[TS00] D. Tsiang and G. Suwala. The cisco SRP MAC layer protocol. IETF RFC 2892, August 2000.

[Tur99] J. S. Turner. Terabit burst switching. *Journal of High Speed Networks*, 1(8):3–16, 1999.

[vALSZ91] H.R. van As, W.W. Lemppenau, H.R. Schindler, and E.A. Zürfluh. CRMA-II: A Gbit/s MAC protocol for ring and bus networks with immediate access capability. In *EFOC/LAN 91, Lndon, England*, pages 56–71, June 1991.

[VvdVTB01] Mark Volanthen, Marcel van der Vliet, Vivek Tandon, and Jim Bonar. *Characterization of Arrayed Waveguide Gratings.* Alcatel, November 2001.

[Wel99] Brent B. Welch. *Practical Programming in Tcl and Tk.* Prentice Hall, 3 edition, 1999.

[Woe97] Hagen Woesner. Primenet - network design based on arrayed waveguide grating multiplexers. In L.S. Lome R.T. Chen, editor, *Design and Manufacturing of WDM Devices*, volume 3234 of *Proceedings of SPIE*, pages pp.22–28, Bellingham, Washington, USA, Nov 1997. SPIE.

[WV96] Jean Walrand and Pravin Varaiya. *High-Performance Communication Networks*, chapter 5: Asynchronous Transfer Mode. Morgan Kaufmann Publishers, Inc., 1996.

[YQD01] Myunsik Yoo, Chunming Qiao, and Sudhir Dixit. Optical burst switching for sevice differentiation in the next-generation optical internet. *IEEE Communcations Magazine*, 39(2):98–104, February 2001.

[ZJM00] Hui Zang, Jason P. Jue, and Biswanath Mukherjee. A review of routing and wavelength assignment approaches for wavelength-routed optical wdm networks. *Optical Networks Magazine*, 1(1):47–60, January 2000.