Fibre-Optic AWG Networks Supporting Real-Time Communication in High-Performance Embedded Systems

KRISTINA KUNERT

Department of Computer Science and Engineering
Chalmers University of Technology

School of Information Science, Computer and Electrical Engineering
Halmstad University

Göteborg, Sweden, November 2008
Fibre-Optic AWG Networks Supporting Real-Time Communication in High-Performance Embedded Systems

KRISTINA KUNERT

Copyright © Kristina Kunert, 2008.
All rights reserved.

Technical report no. 58L
ISSN 1652-876X

Department of Computer Science and Engineering
Chalmers University of Technology
Hörsalsvägen 11
SE-412 96 Göteborg
Sweden
Phone: + 46 (0) 31-772 1000

Contact Information:
Kristina Kunert
School of Information Science, Computer and Electrical Engineering (IDE)
Halmstad University
Box 823
SE-301 18 Halmstad
Sweden
Phone: +46 (0) 35-167100
Email: kristina.kunert@hh.se

Cover:
Schematic view of the proposed network architecture.

Printed in Sweden
Chalmers Reproserivce
Göteborg, Sweden 2008
Fibre-optic AWG networks supporting real-time communication in high-performance embedded systems

KRISTINA KUNERT

School of Information Science, Computer and Electrical Engineering, Halmstad University and Department of Computer Science and Engineering, Chalmers University of Technology

Abstract

High-performance embedded systems communicating heterogeneous traffic with high bandwidth and strict timing requirements are in need of more efficient communication solutions. This thesis proposes two multi-wavelength passive optical networks able to meet these demands. The networks are based upon a single-hop star topology with an Arrayed Waveguide Grating (AWG) placed in the centre. The intended application areas for the two networks are short range embedded communication systems like System Area Networks (SANs) and router architectures with electronic queuing. The AWG’s attractive property of spatial wavelength reuse, as well as the combination of fixed-tuned and tuneable transceivers in the end nodes, enables simultaneous control and data traffic transmission. This, in turn, makes it possible to support heterogeneous traffic with both hard and soft real-time constraints.

Additionally, two Medium Access Control (MAC) protocols, one for each network solution, are developed. Traffic scheduling is centrally controlled by a node, the protocol processor, residing together with the AWG in a hub. All nodes use Earliest Deadline First (EDF) scheduling and communicate with the protocol processor through physical control channels. A case study, including simulations, in the field of Radar Signal Processing (RSP) and simulations using periodic real-time traffic are conducted for the two application areas respectively, showing very good results. Further, a deterministic real-time analysis is conducted to provide throughput and delay guarantees for hard real-time traffic and an increase in guaranteed traffic is achieved through an analysis of existing traffic dependencies in a multichannel network. Simulation results incorporating the traffic dependency analysis indicate a considerable increase in the possible guaranteed throughput of hard real-time traffic.

Keywords: Arrayed Waveguide Grating (AWG), real-time communication, optical communication, network architecture, optical network, medium access control (MAC) protocol, real-time analysis, traffic dependency analysis
Acknowledgements

Some people express already as a toddler a career ambition that is realistic enough to be pursued successfully throughout their lives. I wanted to be an astronomer and at the same time work with animals, so I do not really count myself into this category. Other people’s professional fate is to a large degree decided by the persons they encounter and those persons’ ability to convince and spread their infectious enthusiasm and joy. I want to thank my supervisor Professor Magnus Jonsson for being one of those key persons. I still remember calling him to gather information about the bachelor program in ICT that he was responsible for back then. Well, the rest is history… I also want to thank Magnus for his patience and understanding. Without him I would not do what I am doing and I would not be where I am today.

A particularly warm thank you goes to my friend and colleague Annette Böhm. Her influence and support stretch far beyond the scope of this thesis. We are different enough to complement each other perfectly and alike enough to strive for common goals, have a fruitful cooperation and great fun on the way.

This research was done at the Centre for Research on Embedded Systems (CERES), funded by the Knowledge Foundation. CERES not only provided me with a creative work environment but also with supportive colleagues. In particular, I want to thank Matthias Weckstén for an enjoyable research cooperation, and Dr Xing Fan and Dr Hoai Hoang for interesting and helpful discussions when I was in the beginning of my Ph.D. studies. Thank you also to Dr Elisabeth Uhlemann, Dr Nicholas Wickström and Professor Bertil Svensson for constructive feedback on my thesis.
List of publications

Papers on which this thesis is based upon:


Shorter version of Paper 3, presented at a national conference.


Presentation of an optoelectronic router architecture with an AWG-based switch fabric and electronic queuing. A MAC protocol with Quality of Service support is described and evaluated in a simulation study. Additionally, a real-time analysis for hard real-time traffic is provided.


Description of a network architecture and real-time MAC protocol for a System Area Network together with a case study of the suitability of the network for radar signal processing traffic. An analysis for guaranteeing hard real-time traffic is provided.


Short summarization of Paper 2 and Paper 3, presented at a national conference.

Shorter version of Paper 2, presented at a national conference.


Presentation of an analysis of traffic dependencies, incorporated with a real-time scheduling analysis, which makes it possible to guarantee a higher amount of hard real-time traffic in a network.


Journal version of Paper 6, including a summary of results from Paper 2 and Paper 3.

Other papers


## Contents

1 INTRODUCTION.................................................................................................................. 13

2 FIBRE-OPTIC NETWORK DESIGN ELEMENTS AND PRINCIPLES................................................. 17
   2.1 OPTICAL WDM SYSTEMS ........................................................................................................ 17
   2.2 THE AWG COMPONENT .......................................................................................................... 18
      2.2.1 Comparison with the PSC ............................................................................................ 21
   2.3 TRANSCEIVERS ..................................................................................................................... 22
   2.4 SINGLE-HOP NETWORKS VS. MULTIHOP NETWORKS ............................................................ 23
   2.5 CLASSIFICATION OF MAC PROTOCOLS ............................................................................. 24

3 RELATED WORKS .................................................................................................................. 27
   3.1 MAC PROTOCOLS FOR AWG-BASED SINGLE-HOP WDM STAR NETWORKS .................... 27
   3.2 REAL-TIME MAC PROTOCOLS FOR SINGLE-HOP WDM STAR NETWORKS .................... 29
   3.3 AWG-BASED ARCHITECTURES ............................................................................................ 35

4 THE REAL-TIME AWG NETWORK ......................................................................................... 37
   4.1 NETWORK REQUIREMENTS ................................................................................................ 37
      4.1.1 Real-time traffic classes ................................................................................................ 38
   4.2 NETWORK ARCHITECTURE .................................................................................................. 38
   4.3 PROTOCOL BASICS .............................................................................................................. 41

5 USING THE REAL-TIME AWG NETWORK IN SYSTEM AREA NETWORKS .................................. 45
   5.1 NETWORK ARCHITECTURE .................................................................................................. 45
   5.2 PROTOCOL DETAILS ............................................................................................................ 45
      5.2.1 The control traffic ......................................................................................................... 46
      5.2.2 The scheduling algorithm ............................................................................................ 48
   5.3 CASE STUDY WITH SIMULATION ANALYSIS .................................................................. 49
      5.3.1 Case definition .............................................................................................................. 49
      5.3.2 Simulation results ........................................................................................................ 50

6 USING THE REAL-TIME AWG NETWORK IN A DISTRIBUTED OPTOELECTRONIC ROUTER ....... 57
   6.1 ROUTER ARCHITECTURE ...................................................................................................... 58
   6.2 PROTOCOL DETAILS .......................................................................................................... 58
      6.2.1 The control traffic ......................................................................................................... 60
      6.2.2 The scheduling algorithm ............................................................................................ 62
   6.3 SIMULATION ANALYSIS ..................................................................................................... 62

7 IMPROVED HARD REAL-TIME TRAFFIC SUPPORT .................................................................. 69
   7.1 REAL-TIME ANALYSIS ........................................................................................................ 69
   7.2 TRAFFIC ANALYSIS ALGORITHM ...................................................................................... 73
   7.3 IMPROVED, LESS PESSIMISTIC FEASIBILITY ANALYSIS ................................................... 75
   7.4 SIMULATION ANALYSIS ..................................................................................................... 77

8 CONCLUSIONS ..................................................................................................................... 83

REFERENCES ............................................................................................................................ 85
1 Introduction

As new applications based on embedded systems are developed, the requirements on communication networks for such systems continue to increase. New architectures are proposed and new components are employed to meet demands like high performance, speed and reliability to a reasonable price. Real-time services, i.e., being able to guarantee that the data over a communication network reach their destination within a predefined deadline, play a vital role in high-performance networks used in a wide range of different applications. Examples of such applications include cluster computing, radar signal processing and streaming video.

While network speeds are increasing, the performance of data- and telecommunication equipment has to ensure to not lack behind in development. However, at the same time, the internal interconnection complexity in the communication equipment, which might be growing at a rate of $N^2$ ($N$ being the number of ports), must not be neglected. This requires new interconnection technologies to be used internally in the equipment. Optical interconnection technology is a promising alternative that enables both high throughputs and presents solutions to the problem of interconnection network complexity, as light beams can cross each other without interference. On the other hand, electronic components are still needed to cope with the increasing demands on Quality of Service (QoS) support as electronic solutions, e.g., make flexible traffic scheduling possible. An overview of optical interconnections in communication equipment can be found in [Jonsson 2003].

Due to their advantageous properties, optical interconnects and devices are commonly used in modern communication systems. High bandwidth, low loss, and immunity to electro-magnetic interference (EMI) and radio-frequency interference (RFI) are some of the reasons that give optical architectures an advantage over those merely built upon electronics. While using optics is rather common in long distance communication systems, the research on optical solutions inside embedded systems is limited. In the past, broadcast-and-select networks incorporating wavelength-independent components, as, e.g., Passive Star Couplers (PSC), have received a great amount of attention in the research community. However, thanks to recent advances in the research concerning optical components, wavelength-sensitive components, as, e.g., Arrayed Waveguide Gratings (AWGs), present a viable alternative. Wavelength-routing avoids unnecessary broadcasting, and instead makes it possible, when looking at the example of a single-hop network, to address each node by an individual
wavelength. AWG-based networks, especially in combination with real-time support, are an area that is quite unexplored by researchers so far.

Embedded systems have a growing need for high-performance real-time communication, leading to a constant demand to search for possibilities to guarantee real-time performance for a high fraction of the network traffic in the emerging embedded networks. Network solutions incorporating the AWG component were chosen to investigate due to its reported good qualities, as, e.g., high concurrency, high reliability, low loss, and nonblocking connectivity [Takahashi et al. 1995][Smit et al. 1996], which make it well suited for high-performance embedded networks. The first approach was to investigate the possibility of combining the AWG with electronic packet scheduling and previously acknowledged real-time analysis methods, aiming at the design of a network architecture and a medium access control protocol. Furthermore, two scenarios were studied in order to investigate the suitability of the proposals in two embedded system applications.

This thesis presents two single-hop Wavelength Division Multiplexing (WDM) star networks, with an AWG at its centre, for implementation in high-performance embedded systems with high demands on bandwidth and timing. Two medium access control (MAC) protocols are described that make use of the special properties of the architecture and the AWG component, offering support for three different traffic classes, guaranteed hard real-time (HRT), soft real-time (SRT) and nonreal-time traffic (NRT), in order to handle the heterogeneous traffic present in the targeted systems. The proposed MAC protocols rely on traffic scheduling centralized in one node, called the protocol processor (PP), which communicates with the end nodes through physical control channels. The AWG’s attractive property of spatial wavelength reuse and a combination of fixed-tuned and tuneable transceivers in the end nodes enable simultaneous control and data transmissions. One network solution is intended for short range communication systems such as, e.g., System Area Networks (SANs) and similar high-performance embedded systems with possible application areas as cluster computing, distributed large routers and distributed video and imaging applications. The feasibility of the proposed SAN is demonstrated by a case study in the area of Radar Signal Processing (RSP), including a simulation study. The analysis of the throughput, delay and deadline miss ratio for defined requirements showed the network’s suitability for applications with this kind of heterogeneous real-time communication requirements. Furthermore, the suitability of an AWG-based network in a router architecture with electronic queuing and QoS handling is studied, where the traffic in the router is transported from input ports to output ports through the passive optical network. A similar network architecture as for SANs is proposed, and the MAC protocol is enhanced, though still handling three different traffic classes in order to ensure the QoS handling of the traffic. In this case,
HRT can, e.g., be used for remote surgery over the network, while SRT is the main class of real-time traffic. The optoelectronic router’s suitability for supporting heterogeneous real-time traffic is illustrated by the means of simulation. Prioritization between different packets within one real-time class is done using Earliest Deadline First (EDF) scheduling, both in the end nodes and the protocol processor, in both networks.

Aiming at the support of hard real-time traffic, one has to be able to provide guarantees for the throughput of packets with timing constraints. The deterministic analysis available for real-time communication has its origin in the area of task scheduling in real-time systems. This analysis has been mapped onto the communication context by Hoang et al. [Hoang et al. 2003]. However, included in the analysis method is considerable pessimism which originates from the fact that scheduling on a uniprocessor cannot be mapped directly on a multichannel network that has the possibility of concurrent transmissions. This thesis presents an approach of successfully increasing the amount of possible guarantees by analyzing traffic dependencies.

The results indicate clearly that AWG-based multichannel optical networks in cooperation with real-time medium access control protocols are very well suited for meeting the demands that real-time traffic generated by typical high-performance embedded systems applications might have. The proposed network architectures and MAC protocols were able to easily support the characteristic real-time traffic patterns typical for the studied embedded systems applications, i.e., System Area Networks and distributed routers. However, the original real-time analysis has limitations as the guaranteed throughput never can reach over one packet per time slot and wavelength channel. By improving the analysis, taking into account traffic interdependencies, and, by that, integrating the possibilities of concurrent transmissions into the analysis, the amount of throughput guarantees could be increased considerably.

The rest of the thesis is organised as follows. Aspects of the network design space are described in Chapter 2, while Chapter 3 gives a short overview over related work. Chapter 4 introduces the basics of the proposed network architectures and MAC protocols, while Chapters 5 and 6 deal with the two chosen application areas. Chapter 5 describes the network architecture, MAC protocol and scheduling algorithm intended for System Area Networks, and presents the results of the case study, including a simulation study. Correspondingly, a description of the network architecture, MAC protocol and the scheduling algorithm for the distributed router is provided in Chapter 6, and an evaluation by simulation is presented. Chapter 7 discusses the feasibility analysis and how to improve its basic version through traffic
analysis. The improvement is illustrated by simulation results. Chapter 8 concludes this thesis.
2 Fibre-optic Network Design Elements and Principles

In order to provide a background for the suggested approach for the network architecture and the medium access control protocol, this chapter shortly treats different network design elements and principles. After a brief introduction to WDM systems, the AWG component is described and compared with the PSC, and also a short introduction to optical transceivers is given. Furthermore, the advantages and disadvantages of single- and multihop networks are discussed, while the chapter is concluded by a short classification of medium access control protocols.

2.1 Optical WDM Systems

Optical networks offer the potential of communication with a low bit error probability and high bandwidth, properties highly useful and much coveted by embedded systems. The optical medium offers an extremely wide bandwidth, not fully exploitable by components working at a speed limited by electronic data rates. In order to make a more satisfying usage of the optical bandwidth possible, a technique for implementing concurrency in communication is needed. The most recognized approach is Wavelength Division Multiplexing, which increases the transmission capacity by making it possible to send separate signals at multiple wavelengths through one single optical fibre.

![Figure 1. A simple WDM system.](image-url)
The basic elements of a WDM system are illustrated in Figure 1. Transmitters emit different individual wavelengths which are combined by the help of a multiplexer so they can travel on the same fibre to a demultiplexer. The demultiplexer in its turn splits the different wavelengths into individual wavelengths again in order to route them to wavelength-specific receivers. In wavelength-routed networks, different optical channels are routed to different destinations based on their wavelength. One way of implementing this is by the help of the AWG component described in the next Chapter. For sources and destinations to be able to send and receive on different optical channels, the transmitters and receivers need to be tuneable. These components are described in Chapter 2.3.

### 2.2 The AWG component

The AWG, also known as Phased Array or PHASAR [Verbeek et al. 1995], is an optical component which consists of an arrangement of optical waveguides on a substrate plate (see Figure 2). Its architecture gives it special spatial and spectral properties suitable for highly effective future photonic communication networks.

![Figure 2. Illustration of a 3 × 3 AWG.](image)
The AWG’s area of usage comprises implementations as wavelength multiplexers and
demultiplexers, but also as an integrated building block in WDM equipment as
Optical Cross-Connects (OXC) or Optical Add-Drop Multiplexers (OADM) [Verbeek et al. 1995].
However, in this thesis, the proposed network makes use of
another application of the AWG, namely as a Waveguide Grating Router (WGR) or
wavelength router [Dragone 1991a][Dragone et al. 1991b], a passive routing device
for optical communication networks. The AWG promises high potential as it is a
strictly nonblocking device with the possibility of a high degree of concurrency
[Dragone 1991a][Dragone et al. 1991b][Smit et al. 1996].

**Basic principles**

An $N \times N$ AWG has $N$ input waveguides and $N$ output waveguides, in the following
also called input and output ports respectively. $N \times N$ AWGs can simultaneously
accept $N$ wavelengths at each of their input ports and route each wavelength to a
specific output port without collision. This ability makes AWGs strictly nonblocking
devices [Smit et al. 1996]. As shown in Figure 3, the AWG offers full connectivity,
which means that each node attached to any of the input ports of the component can
send on all of the AWG’s wavelengths and can directly reach each node connected to

![Figure 3. Wavelength routing pattern of an $N \times N$ AWG.](image)
any output port [Woesner et al. 2003]. This property makes the AWG very well suited for implementing single-hop networks. In order to be able to send and receive on all wavelengths supported by the AWG, the nodes attached to the AWG need to be equipped with tuneable transmitters and tuneable receivers at their input and output ports respectively (or, alternatively, with an array of fixed-tuned transmitters and an array of fixed-tuned receivers). For more information on tuneable transceivers, see Chapter 2.3.

AWGs have two interesting properties described and illustrated in the following. Firstly, they apply periodic wavelength routing [Dragone et al. 1991b]. Suppose a $3 \times 3$ AWG receives six different wavelengths at one of its three input ports (see Figure 4). In a $3 \times 3$ AWG component, each third wavelength will be routed to the same output port. The period between two routed wavelengths at the same output port is called Free Spectral Range (FSR). The number of wavelengths per FSR on each output port is equal to the degree of the AWG, which, in this example, would be equal to three. This also means that each wavelength can be spatially reused exactly three times [Maier 2003b]. Spatial wavelength reuse, the second interesting property of the AWG, signifies that each wavelength can be used on all output ports at the same time without collision [Verbeek et al. 1995]. This is accomplished by wavelength routing and can be utilized for the implementation of multiple, concurrent communication channels. Assuming an $N \times N$ AWG using one FSR, it can simultaneously accept $N$ wavelengths at each input port, and without collisions route each of these wavelengths to a particular output port (dictated by the design of the AWG) [Dragone 1991a][Dragone et al. 1991b]. Each output port will in its turn receive $N$ wavelengths, one from each input port. The routing mechanism is easiest illustrated by plotting the

\[\text{Figure 4. Periodic wavelength routing.}\]
wavelength distribution in the aforementioned $3 \times 3$ AWG. The wavelengths on input 1 are routed to the three output ports as described before, while wavelengths on input 2 and 3 are routed in a corresponding way, filling all the existing gaps in a cyclic manner (see Figure 5). This routing scheme makes it possible to build efficient network architectures based on AWGs as it improves the throughput and latency behaviour of optical networks [Maier et al. 2000b][Dolzer et al. 2002]. One drawback of the AWG though is its incapability of supporting multi- or broadcast communication over AWG-based networks. This shortcoming can, however, be compensated for by using spectral splicing of a broadband light source as presented in detail in [Maier et al. 2003b], or by simultaneously activating several emitters in an array component.

2.2.1 Comparison with the PSC

Up to now, the most commonly used routing device in optical single-hop networks is the PSC. Unlike the AWG, numerous research projects have dealt with the PSC and the broadcast-and-select networks implemented by the help of them. Those networks are characterized by the fact that all input wavelengths are equally distributed (broadcasted) to all the output ports of the PSC, and the destination nodes have to select the corresponding wavelength by the help of a tuneable receiver [Acampora et al. 1989] [Gerstel 1996]. The splitting of the signal when broadcasting results in splitting losses and waste of optical power used for sending data to all nodes, not just to the intended destination node. In the AWG, data are exclusively sent to the node that is intended as destination, which leads to the possibility of better bandwidth

![Figure 5. Spatial wavelength reuse.](image)
usage and an improved throughput-delay performance [Maier et al. 2001]. Compared with the PSC, the signal is transported through the AWG with almost no losses. The main drawback of the PSC is the fact that, due to its broadcast nature, each wavelength can only be used once at a time. In order to accommodate many channels simultaneously, a large number of transceiver pairs is required by the connected nodes. Due to its capability of spatial wavelength reuse and its unicast nature, the AWG offers the possibility of a much higher concurrency and throughput [Maier et al. 2000b]. A detailed comparison between the AWG and the PSC is presented in [Maier et al. 2000b], whereas the advantages of both approaches are combined in an optical single-hop network based on an AWG in parallel with a PSC in [Fan et al. 2003].

### 2.3 Transceivers

In order to make optimal use of the AWG, each, directly to it connected, node in the network has to be equipped with transmitters and receivers that support the range of all wavelengths used for communication in the network.

**Transmitters**

There are two types of light sources that can be used in this context, Light Emitting Diodes (LEDs) and diode lasers. LEDs are very commonly used in optical communication due to their rather low price. For short distances and/or low data rates, a spectrally sliced LED can be a cost effective substitute for an array of expensive lasers [Maier 2003a]. The advantage of lasers is their high bandwidth-distance product and the high output power that can be achieved. Lasers come in two different working modes, fixed-tuned to one single wavelength or tuneable over a range of different wavelengths. (One tuneable transmitter can always be replaced by an array of several corresponding fixed-tuned ones.) In [Maier 2003a], Maier summarizes the properties, as, e.g., tuning range and tuning time, of three different types of tuneable lasers: mechanical tuneable lasers with a tuning range of 500 nm and a tuning time of 1-10 ms, acousto-optical lasers (tuning range 100 nm, tuning time 0.01 ms) and electro-optical lasers (tuning range 10-15 nm, tuning time 1-10 ns). As the time factor is crucial in real-time communication systems, it is most advantageous for the network proposed in this thesis to use fast tuneable electro-optical lasers in order to keep the delay caused by tuning as short as possible. An alternative is to use a laser diode array in each node, where the different diodes emit different wavelengths. The diode with the desired wavelength is activated when demanded.
Receivers

Exactly as a transmitter, a receiver can be either fixed-tuned or tuneable over several wavelengths, and also in this case, an array of fixed-tuned receivers can be used instead of a tuneable one. The tuning ranges and tuning times of the mechanically tuneable, the acousto-optical, and the electro-optical receivers are comparable to their transmitting counterparts as can be seen in the more detailed comparison in [Maier 2003a]. The electro-optical fast tuneable alternative seems to be the most reasonable choice due to the same reasons stated regarding the transmitters.

The following basic combinations of transceivers in the source and destination nodes are possible [Mukherjee 1992a]:

- FT-FR (fixed-tuned transmitter - fixed-tuned receiver)
- FT-TR (fixed-tuned transmitter - tuneable receiver)
- TT-FR (tuneable transmitter - fixed-tuned receiver)
- TT-TR (tuneable transmitter - tuneable receiver)

All of these combinations can be used with or without the usage of a logical control channel (CC) in the protocols designed for the network, and also more than one of each transceiver can be integrated in the network architecture. This results in the general abbreviation (CC)-FT\textsubscript{i}TT\textsubscript{j}-FR\textsubscript{m}TR\textsubscript{n}, where the number of transmitters in each node is specified by the integers \(i\) and \(j\), and the number of receivers in each node by \(m\) and \(n\). In the case of the proposed network, it is essential that tuning is made possible at both ends, i.e., at the source and at the destination, which makes TT-TR, e.g., implemented by arrays of fixed-tuned transceivers, the favourable alternative.

2.4 Single-hop networks vs. multihop networks

WDM networks can be divided into single-hop and multihop networks. In a multihop network, there might not exist a direct wavelength channel between each pair of nodes, which means that the traffic sometimes has to be routed via intermediate nodes. Generally, nodes in a multihop network only need to have a low number of fixed-tuned transceivers, but the nodes are functionally restricted by an electro-optical bottleneck when the packets have to be converted into the electronic domain, processed and converted back into the optical domain at each intermediate node on the routing path [Mukherjee 1992b].
Single-hop networks provide a distance of one single hop between all source-destination combinations, which gives them certain desirable features. Those types of networks are easily managed, bandwidth is used in an efficient manner and no capacity is used on data forwarding, which improves the throughput-delay performance. Additionally, no electro-optical bottleneck is experienced and the stations only need to process packets addressed to themselves, which reduces the protocol processing requirements at each node [Mukherjee 1992a]. However, the need for tuneable transceivers to tune over a larger number of wavelengths and the longer tuning latency introduced by that are conceived as drawbacks.

In [Woesner et al. 2003], the performance of a network with a certain number of nodes and the financial budget are compared for multihop and single-hop networks comprising one AWG component. One multihop network using an AWG is proposed in [Banerjee 1998], while related works in the area of AWG single-hop networks are presented in more detail in Chapter 3.

Due to its efficiency, the single-hop approach is more suitable in the studied case of distributed, networked embedded systems. When having decided on a single-hop approach, and the AWG as a central component, the choice of network topology becomes unproblematic and the star topology is given. Additionally, compared to bus and ring networks, star networks (see Figure 6) are less vulnerable as the failure of a single node (except in case of the central node) or a single interconnection does not jeopardize the connectivity of the system as a whole. Also, the latency in a star is low because the communication path between any two nodes never is longer than one hop. A large amount of research has been carried out in the area of optical star networks and many protocols for this architecture have been proposed. Jonsson [Jonsson 1999] provides an extensive list of WDM star related research.

2.5 Classification of MAC protocols

In [Maier 2002], a general classification of single-hop MAC protocols is provided. Generally, the three main protocol types are preallocation protocols using fixed assignment, random access protocols and protocols using pretransmission control, which in their turn can be divided into two further subgroups, namely, tell-and-go and attempt-and-defer protocols (see Figure 7).

For both medium and high uniform traffic intensity, preallocation protocols with fixed channel assignment are a suitable solution. Resources are allocated in a fixed manner to certain nodes and traffic classes and therefore it is possible to guarantee
QoS without requiring any signalling. For low or bursty traffic loads, this type of protocol leads, however, to a lower utilization of the network resources.

In random access protocols, idle channels can be accessed in a random manner without signalling. This type of contention-based protocol works best at low traffic loads due to the risk of a high number of collisions at medium and high traffic intensities. QoS cannot be guaranteed, thus such protocols are merely suited for nonreal-time traffic.

Pretransmission control protocols work well in highly flexible networks with unpredictable traffic loads as the bandwidth is assigned on demand by reservation signalling. In tell-and-go protocols, a reservation message is sent by the source node, and, without waiting for any result generated by the request, the data are transmitted to the destination. This obviously leads to the risk of possible collisions. In the attempt-and-defer approach, a response to the reservation request is waited for before transmitting any data. This introduces delay into the system, but on the other hand this makes it possible to handle traffic requiring QoS treatment. The MAC protocols proposed in this thesis fall into the category of attempt-and-defer pre-transmission control protocols, as the possibility for support of real-time traffic is crucial.

*Figure 6. Star architecture.*
Figure 7. MAC protocol types.
3 Related Works

As the AWG is a fairly new optical component and the research on how to use it in WDM networks is quite young, only a few related MAC protocols for this architecture have been presented by the research community. However, extensive research has been done in the field of single-hop WDM star networks where broadcast-and-select networks based on the PSC were the centre of attention [Mukherjee 1992a]. Opposed to the PSC, the AWG is a wavelength-sensitive device, a functionality that makes it possible to reach selected parts of the network and increase its concurrency and throughput. Borella et al. [Borella et al. 1999] showed already in 1999 that systems based upon AWGs as compared to PSCs can reach a much higher throughput while still having the same number of ports. Nevertheless, research on AWG-based network architectures, especially with real-time support, is limited.

3.1 MAC protocols for AWG-based single-hop WDM star networks

A MAC protocol based upon time division multiple access (TDMA), is proposed in [Borella et al. 1999] for a network architecture where every node is equipped with one tuneable transmitter and one tuneable receiver. The guarantee of an equal amount of bandwidth for each sender-receiver pair provides a high grade of fairness for all nodes. However, due to the fix allocation of capacity to each node, bandwidth utilization might be low while traffic still can suffer from unnecessary delay. The concept of meeting deadlines is disregarded. Bengi [Bengi 1999] studies the behaviour of the MAC protocol proposed in [Borella et al. 1999] for messages with timing constraints and mixed traffic (both real-time and nonreal-time traffic). He introduces the concept of priority classes into the concept, storing traffic in separate first-in-first-out (FIFO) queues according to priority, and calculates the individual mean queuing delays of each packet. In contrast to the work presented here, no worst-case delay analysis is done and therefore no information about the possible amount of real-time traffic guarantees can be given.

Spencer and Summerfield [Spencer et al. 2000] present in their paper a general MAC protocol (WRAP, WDM Request/Allocation Protocol) for wavelength-routed passive optical networks, e.g., AWG star networks. In the targeted network architecture, each end node is fitted with one fixed-tuned optical receiver and one tuneable optical transmitter. The protocol implements distributed scheduling by the means of in-band signalling, so no control channel is defined. Even if, due to wavelength allocation, this
is a collision-free medium access protocol, and a minimum bandwidth between any
source-destination pair can be guaranteed, no deterministic delay bound analysis is
provided and so no real-time guarantees can be given. The authors’ evaluations point
towards WRAP being suitable for general-purpose data communication applications,
as, e.g., local area networking.

In 2000, Maier et al. [Maier et al. 2000b] introduce a packet-switched, single-hop
WDM network based on an AWG. In simulations, their network with a preallocation
MAC protocol is shown to outperform PSC-based networks. However, as this
protocol is a simple fixed TDMA channel assignment scheme, it has the same
disadvantages as each preallocation MAC protocol, namely, low channel utilization
and unnecessarily long delays, i.e., traffic has to wait although the channel is idle. In
addition, no treatment of real-time traffic is introduced. In a later paper, Maier et al.
[Maier et al. 2000a] improve the network architecture and MAC-protocol. The
medium access protocol is changed to be reservation-based, scheduling packets in a
first-come-first-serve (FCFS) manner, and, amongst other things, the usage of several
FSRs increases network efficiency and concurrency. Although the protocol is
designed to be collision-free and throughput thereby can be increased, no delay
analysis is provided and real-time properties are excluded from treatment. In [Maier
et al. 2002], an alternative for circuit switching for QoS handling was added to the
MAC protocol by reserving a certain amount of bandwidth for the duration of one
node’s connection. An analysis provided in [Maier et al. 2000a] and [Maier et al.
2003b] calculates mean packet delay, but due to the lack of any worst-case discussion,
no delay bound guarantees can be given. For more details on this body of work see
[Maier 2003a] and [Maier 2004a]. By the same research group a comparison of the
PSC and the AWG is presented in [Maier et al. 2001], while the network proposed in
[Fan et al. 2003] combines the two components in a hybrid solution. In [Yang et al.
2004] a control channel based AWG star network with a distributed attempt-and-
defer MAC protocol was compared to different optical ring networks and found to be
superior in performance regarding throughput, packet loss and delay when assuming
unicast traffic. However, no discussion about real-time issues is included in this work.
[Fan et al. 2004] allows for a higher degree of spatial wavelength reuse by using array
components in the end nodes, in comparison to the TT-TR architecture of the earlier
mentioned architectures of this research group. Again only mean packet delay is
analysed probabilistically and, as one of the suggested medium access methods is
contention-based, no deterministic real-time service can be offered. Several of the
aforementioned network architectures and MAC protocols are summarized in [Maier
et al. 2004b], where the suitability of AWG-based networks for, e.g., metropolitan
area networks is argued for.
3.2 Real-time MAC protocols for single-hop WDM star networks

In 1995, Kim et al. [Kim et al. 1995] presented the first MAC protocol for a single-hop WDM star especially developed for multipriority traffic. The protocol is reservation-based, based upon the reservation scheme introduced by Jeon et al. [Jeon et al. 1990], and the access to the slotted control channel is organized by random access. The scheduling is distributed and it is the packet with the highest priority among the successful requests, i.e., those that did not collide, that is scheduled to be transmitted first. According to the performance analysis, the delay of the high-priority packets is considerably lower than that of lower priority ones. However, as no worst-case analysis is provided, nothing can be said about the amount of real-time traffic that can be guaranteed.

Selvakennedy et al. [Selvakennedy et al. 1996] introduce a piggybacked token-passing MAC protocol, including support for traffic of two different priorities. The token can be sent on a separate channel or share the same channel as the data packet, where the data packet is piggybacked behind the token. Although always selecting the higher priority traffic when available, the passing of the token in a round robin fashion introduces considerable delay into the network.

A distributed token-based MAC protocol for supporting real-time traffic is presented in [Yan et al. 1996]. The protocol is aimed for CC-TT-FR networks, where the control channel is dedicated for the token. The token serves both for propagating system information and assigning time for transmissions. Each node has a status table, containing the global status information about the network, as, e.g., which node has the highest priority traffic queued. This status information is used to decide the next hop for the token. However, the protocol is designed having soft real-time traffic in mind, as the QoS parameter is equal to the deadline miss probability of a packet, i.e., real-time classes where 20% of the packets are allowed to be missed have a higher priority as those where the corresponding amount is 25%. No discussion or evaluation for hard real-time traffic, i.e., where 0% deadline misses are allowed, is provided.

Tyan et al. [Tyan et al. 1996][Tyan et al. 2001] develop a preallocation-based access scheme for real-time message streams. A deterministic analysis is given in order to provide timing guarantees, but being a preallocation-based protocol it has inherent disadvantages, as, e.g., potentially low bandwidth utilization.

In [Jonsson et al. 1997], Jonsson et al. presented a control channel-based, time-slotted MAC protocol with distributed control for networks carrying both real-time and best-effort traffic. Each node is guaranteed a certain amount of bandwidth, i.e., a
certain number of time slots. Unused time slots can be claimed by other nodes; however, the slot release algorithm is very basic which leads to a nonnegligible probability of unused slots. This decrease in network utilization might result in a lower fraction of real-time guarantees and an increased deadline miss probability for soft real-time traffic. [Jonsson et al. 1997] provides a deterministic performance analysis and shows that the latency for real-time messages is bounded.

Dong et al. [Dong et al. 1998] introduced a preallocation-based time slot allocation scheme targeted towards real-time traffic with hard rate requirements, timing constraints and negotiable demands on jitter. The scheduler assigns the highest priority to the message stream with the shortest deadline; however, the deadline of each instance of this stream is dependent upon the allocation of the previous instance and its jitter requirement. When a set of message streams is judged not schedulable by the scheduler, the jitter demands are renegotiated. In a simulation, this allocation scheme is compared to [Tyan et al. 1996] and it is shown that a higher success rate for schedulability testing can be reached when relaxing the jitter demands. This way of allocating time slots is clearly targeted towards jitter-sensitive traffic as, e.g., streaming multimedia, which explains the prioritization of constant jitter over guaranteed deadline. Additionally, the preallocation of time slots tends to result in lower network utilization.

A hierarchical scheduling framework for on-demand traffic flow scheduling is introduced, analyzed and evaluated in [Li et al. 1998]. The framework separates flow scheduling within one node, deciding for which flow to request transmission, and transmission scheduling between all nodes, actually allocating wavelength channels to flows avoiding transmission and receiver collisions. Medium access is control channel-based with one wavelength channel dedicated to control traffic. The control channel is time-slotted, providing each node access to one mini-slot per control slot. The flow scheduling is simply divided into real-time and nonreal-time traffic, not taking into account possible differences in deadline within the real-time class where scheduling is done in a round-robin fashion between different flows instead. Transmissions are scheduled according to the earliest arrival time of real-time packets (with a random selection in case of equal arrival times), while nonreal-time traffic has the lowest priority. Also in this case, shorter deadlines are not taken into consideration. The deadline independence of both flow and transmission scheduling might lead to a higher probability of deadline misses. Being an on-demand algorithm, including random decisions and an unknown number of possible retransmissions, a deterministic analysis is not possible.

Kam et al. present in [Kam et al. 1998a] and [Kam et al. 1998b] a time-slotted star network with support for virtual circuits. Control channel-based scheduling
algorithms both providing and not providing bandwidth guarantee are described. Scheduling is done in a centralized fashion by a scheduler, which transmits the schedules to each node in the network in a pipelined way. This introduces considerable delay which grows with an increasing number of nodes. One of the assumptions is that only a certain amount of the network’s resources can be allocated to guaranteed traffic, while the rest is shared in a fair manner between all virtual circuits. This arrangement opens up for the possibility of prioritizing best-effort traffic over real-time traffic, increasing the risk of missing deadlines.

In [Bengi 1999], Bengi adapts two control channel-based MAC protocols for real-time traffic. The Distributed Queue (DQ) protocol and the Earliest Available Time Scheduling (EATS) protocol were originally presented in [Lu et al. 1992] and [Jia et al. 1995] respectively. Bengi introduces two improvements for both these protocols. The DQ-p protocol reserves bandwidth for real-time control traffic, and in the DQ*-protocol, real-time messages are always send in their entirety, holding on to once reserved time slots. The EATS protocol was adapted by adding priority scheduling in the transmission queues of sending nodes (EATS-p), and further by assigning the earliest available data channel to the traffic class holding highest priority while lower priority traffic is assigned later available channels (EATS*). The original EATS protocol assigned the earliest possible channel to all traffic instead. Although the mean delay for real-time packets is decreased by these improvements in both protocols, as well as the packet drop probability due to deadline misses, the results are only reached by simulation and no deeper analysis is given. Bengi continued by developing a new reservation-based MAC protocol called CONRAD (Convergence of Real-Time and Data Services) [Bengi 2001] [Bengi et al. 2001]. In order to balance network utilization by real-time and nonreal-time traffic, control channel access is divided into contention-based access for real-time traffic and TDMA-based, and thereby collision-free, access for best-effort traffic. Network control is implemented in a distributed way, leading to the need of status tables in each node, keeping track of resource allocation in the network. Evaluations indicate a high network throughput and an improved performance for the best-effort traffic compared to when using EATS-p, where real-time traffic is always prioritized over best-effort traffic. Clear disadvantages in CONRAD are firstly the FCFS scheduling in the nodes, as traffic with short deadlines can be blocked by traffic with longer deadlines, and therefore the probability of deadline misses increases, and secondly the need of several status tables in each node. Additionally, the lack of possibility to send control messages concurrently from all nodes, and instead having to wait for all nodes to send serially, increases the delay from transmission request to actual transmission considerably.
In [Ma et al. 1998], Ma et al. present several distributed, reservation-based scheduling algorithms for a CC-FTTT-FRTR network. Priority assignment to the messages is done depending upon their laxity (Minimum Laxity First, MLF) and the message length, i.e., short laxity or short message length result in higher priority. An analytical model to calculate the probability of a deadline miss is provided for the case of FCFS scheduling in the nodes and MLF transmission arbitration. Simulations indicate that sorting queues in laxity order and additionally scheduling transmissions according to MLF results in the lowest deadline miss probability for hard real-time traffic amongst the tested schemes. Unfortunately, only the message at the head of each queue is guaranteed a timely delivery, and no analysis is provided for taking into account the worst-case queuing delay. In [Ma et al. 2001], scheduling real-time messages with MLF is combined with an insertion scheduling technique where nonreal-time messages can be sent over wavelength channels not being used by real-time traffic at the moment. This technique decreases the mean delay of best-effort traffic and improves network utilization, but as shown in simulation it increases the amount of deadline misses for hard real-time traffic. A further alternative scheduling algorithm is introduced in [Ma et al. 2003], presenting a new priority scheme, where a message’s priority depends upon its length, its deadline and a constant indicating different types of messages. Simulations indicate a higher deadline miss ratio than if only scheduling according to MLF. In a continuation of this work, Huang et al. [Huang et al. 2007a] use this priority scheme for QoS prediction at run time. The weights of the different components of a message’s priority (length, deadline, type) can be dynamically adapted in order to keep the experienced QoS at a constant level. However, this run-time priority adaptation is not accompanied by a deterministic real-time analysis.

In [Ma et al. 2000a], [Ma et al. 2000b] and [Ma et al. 2000c], a mechanism for guaranteed real-time performance is provided, including admission control, traffic regulation and a reservation-based scheduling algorithm. The admission control is done by the help of a schedulability test, ensuring that new application streams only are accepted if their bounded delay can be guaranteed (together with the delay bound of all already accepted streams). Traffic regulation is implemented by delaying messages that do not conform to their traffic specification. The scheduling algorithm proposed is a combination of an adaptive round-robin scheme, servicing the already accepted traffic streams, and the earlier mentioned EATS protocol [Jia et al. 1995] for assigning wavelength channels to the messages to be transmitted. A delay bound analysis is provided, and the scheme is evaluated by a trace-driven simulation. In the simulation data it becomes obvious that the delay bound guarantee for the traffic results in a poor network utilization, only approximately 50% at an input traffic load near one [Ma et al. 2000a].
Kuri et al. [Kuri et al. 2001] developed a MAC protocol for IP differentiated services on a control channel-based network. As this MAC protocol targets differentiated service for IP traffic, the QoS management is coarse grained (only a limited number of traffic classes), not making a flow-based traffic prioritization possible. Additionally, the scheduling algorithm described in the paper includes random aspects, which makes a deterministic analysis of the delay bound impossible.

In [Diao et al. 2001], a priority-differentiated scheduling scheme is introduced which makes it possible for high-priority packets to preempt prescheduled low-priority packets in order to decrease packet delay for real-time traffic and alleviate the head-of-line effect. Transmissions are coordinated by a shared control channel with time-multiplexed access. Using distributed scheduling control, each node has to maintain global knowledge of the network. Simulations indicate a decrease in mean packet delay compared to, e.g., the EATS algorithm, but no analytical prove is provided. Furthermore, neither throughput guarantees nor bounded delay are discussed.

[Kim et al. 2002] describes a transmission scheduling scheme for a CC-FTTT-FRTR architecture, supporting three traffic classes. The scheme includes a self-collision prediction mechanism in order to foresee and avoid collisions, implemented by the means of a control database in each access node. According to [Kim et al. 2002], the computationally complex configuration of this database makes the protocol more suitable for networks in the metropolitan scale than on the local one. Unfortunately, no real-time analysis is provided.

Choi introduces in [Choi 2003] a MAC protocol offering bandwidth guarantees. The protocol combines both contention-based and reservation-based medium access, where traffic in need of bandwidth guarantees uses the reservation option and best-effort traffic has to content for the medium. Only a fraction of the bandwidth, however, is allocated for high-priority traffic leading to the traffic suffering of unnecessary high delays, increasing the probability for deadline misses, while best-effort traffic is send instead. An analysis of the throughput of real-time traffic is given, together with the calculation of throughput and average delay for best-effort traffic. The question of deadline guarantees is not treated.

In [Wang et al. 2003] and [Wang et al. 2006], a MAC protocol targeted towards multimedia traffic is described (M-WDMA, Multimedia Wavelength Division Multiple Access). The protocol tries to combine the advantages of TDMA-based, reservation-based and random access approaches in order to meet the needs of traffic types with different characteristics. Both static and dynamic bandwidth allocation is suggested, used in M-WDMA and M-WDMA+ respectively. Each network node is fitted with one fixed-tuned receiver and three tuneable transmitters, one for each
traffic type, i.e., constant bit rate and two types of variable bit rate. Additionally to those, a fixed transmitter/receiver pair is needed for establishing a control channel. Each traffic type is serviced by one of three subprotocols (based on time multiplexing, reservation and random access respectively) integrated into M-WDMA. In [Wang et al. 2003], the mean packet delay and maximum throughput are analysed and in a simulation M-WDMA+ is compared to the CONRAD protocol [Bengi 2001] [Bengi et al. 2001] and shown to reach both higher throughput and lower average delay. In [Wang et al. 2006], the approximation of the deadline miss rate is added to prior results and an “admissible region”, depending upon, e.g., traffic load and traffic type, is calculated, indicating to the admission control whether a certain QoS can be provided for the application. However, based upon approximations and targeting multimedia traffic, this cannot provide a deterministic delay bound guarantee for periodic hard real-time traffic, as needed by the applications targeted in the work presented in this thesis.

Bellón et al. [Bellón et al. 2004] presents an EDF-based protocol for a CC-FTTT-FRTR network architecture. The reservation-based protocol is designed to support real-time traffic and depends upon distributed scheduling control, which in its turn is accompanied by a large number of tables in each node keeping track of the global state of the network at each time instance. A simulation indicates a low deadline miss rate for the case of HDTV traffic, but the results are not substantiated by any deterministic real-time analysis. Therefore neither throughput nor bounded delay can be guaranteed.

In [Huang et al. 2004], a scheduling scheme is introduced which tries to alleviate the problem of longer messages blocking shorter ones when using purely deadline-dependent scheduling. This blocking can result in deadline misses for the blocked packets. By dropping packets according to the described algorithm, simulations show that the throughput in the network can be increased and the packet loss rate decreased. However, this provides only long term statistical QoS, as no throughput and/or delay bound guarantee on the packet level can be given. In [Huang et al. 2005], analytical comparisons to other scheduling algorithms are provided. Further work in [Huang et al. 2007b] integrates the scheduling algorithm with the dropping scheme into a framework, the goal of which is to optimize the ordering of the packets to minimize packet loss. Extensive simulations indicate an optimal performance in this regard. [Huang et al. 2008] follows up this work analyzing the average message delay (however, not using the dropping scheme). Although targeting a minimum of deadline misses, neither of the mentioned results include the possibility of guaranteeing deadlines on a per packet level.
Petridou et al. [Petridou et al. 2008] present a scheduling approach based upon the clustering of nodes into those with long or short messages, while still keeping the classification into high-priority and low-priority messages. Long messages will be scheduled ahead of short one, and all high-priority messages before any of the low-priority messages. Experimental evaluation indicates an increased network throughput and shorter mean delay compared to a basic pretransmission coordination-based scheduling algorithm. No discussion on delay bound guarantees is provided.

3.3 AWG-based architectures

In the WASPNET (Wavelength Switched Packet Network) project [Nizam et al. 1998][Hunter et al. 1999], the main aim is to investigate the potential of optical packet switching, and the switch fabric is chosen to be implemented by the help of an AWG. Both hardware design and network management are studied, and a test bed is implemented. The suggested network is a reconfigurable multiwavelength transport network. The switch architecture uses feedback delay lines in cooperation with a wavelength-selective router, as, e.g., an AWG. The usage of feedback delay lines makes it possible to implement multiple packet priorities and a functionality where higher priority packets can preempt lower priority packets is included in the design [Chia et al. 1999]. However, the complex architecture also includes the use of a large number of other components as, e.g., tuneable wavelength converters at each AWG input port and output port, adding both cost and possible points of failure. Early tests showed the potential of AWG-based switches to outperform switches based upon broadcast-and-select technology [Hunter et al. 1999]. In [Chia et al. 2000], the WASPNET switch architecture is compared to other optical packet switch designs and found to reduce losses due to optical splitting and combining. An analytical framework for the calculation of packet loss probability and mean packet delay is presented in [Chia et al. 2001], but real-time demands are not taken into account.

Cheyns et al. [Cheyns et al. 2003] propose a possible architecture, based upon an AWG, for a switching node for optical packet-based networks. The AWG is used to implement the switching matrix in the node and it is connected with the input and output fibres by the means of tuneable wavelength converters. A switch controller is used to assign wavelengths according to different suggested algorithms, including certain random decisions, and the architecture is evaluated by the help of simulated Poisson traffic. While the application area of the AWG is similar to the suggestion for the distributed router in this thesis, the approach is different. As the research presented here aims towards the support of real-time communication, the AWG’s
property as a wavelength router is taken advantage of by letting each wavelength serve as an address to a specific destination node. Excluding any randomness from the design described here makes it possible to provide a deterministic delay bound analysis.

Gripp et al. [Gripp et al. 2003] describe an optical switch fabric, based on an $N \times N$ AWG, for a high-capacity IP router. Tuneable transmitters and fixed-tuned receivers are connected to the input and output ports of the AWG respectively. A scheduler can change the connection pattern between input and output ports by configuring the wavelengths of the transmitters. A hardware implementation shows the suitability of these kinds of architectures. However, by limiting the usage of tuneable components to the transmitter side and not integrating also tuneable receivers into the architecture, the amount of concurrency is confined to a lower level.

In [Ngo et al. 2004] and [Ngo et al. 2006], Ngo et al. present a switch fabric based upon AWGs and wavelength converters connected in a nonblocking multistage architecture. Each input fibre is connected to a splitter, demultiplexing the wavelengths which in their turn continue through the limited range wavelength converter before entering the first stage of numerous parallel AWGs. Before each new stage of AWGs, and before the wavelengths are multiplexed onto the output fibre, the wavelengths again need to pass through the same type of converter. This architecture contains a high number of components and therefore, the implementation cost and difficulty will increase as the number of wavelengths increases.

The IRIS (Integrated Router Interconnected Spectrally) project targets the problems of developing all-optical packet routers. As a partial solution, [Neilson et al. 2005] and [Bernasconi et al. 2006] suggest an architecture based upon wavelength switching and load balancing. The function of wavelength switching is implemented in a multistage architecture, where the first and the last stage incorporate an AWG. Even this is a complicated structure containing numerous components, increasing implementation difficulties and possible points of failure. However, the amount of concurrency is higher than in the switch architecture described in [Ngo et al. 2004] and [Ngo et al. 2006]. It also surpasses in this respect the suggestion presented in this thesis.

Kyriakis-Bitzaros et al. [Kyriakis-Bitzaros et al. 2006] present an optical backplane based on a TT-FR architecture with an AWG as a passive wavelength router at its centre. The nodes are directly attached to the AWG, and the tuneability of the transmitters ensures the all-to-all connectivity. However, no arbitration mechanism in order to avoid receiver collisions is provided. Implementation tests and bit error rate measurements indicate good scalability.
4 The Real-Time AWG Network

This chapter introduces the network architecture and its protocol on a more general level, while details about the two suggestions, for a SAN and a distributed router, and their individual differences are described in the two following chapters. For natural reasons there will occur some minor overlap between these descriptions.

4.1 Network requirements

In the design process of any network, certain fundamental prerequisites have to be considered before making any decisions upon its architecture. The two exterior circumstances which have the most decisive influence on the network requirements in this case are, on the one hand, the fact that the network has to be suitable as an interconnection network in an embedded system and, on the other hand, the strict real-time performance demands on this interconnection network.

The reliability of the network is highly prioritized so that all connections are accessible for each node whenever it might need to send any data. Besides that, full network connectivity is mandatory in the network designed in this work, i.e., each node has to be able to reach any other node in the network. As a single-hop network is chosen, full network connectivity implies in this case that no two nodes will be further apart from each other than one single hop. This minimizes propagation delays and requires lower resources in terms of transmission capacity and hardware complexity per node, a fact especially important in embedded systems with, e.g., size, cost, and complexity requirements.

As two of the three present traffic classes, hard real-time traffic and soft real-time traffic, have real-time requirements, being able to guarantee QoS for real-time traffic is an essential requirement on the network. The goal is to schedule the different traffic classes in a fair manner, where all packets with an identical absolute deadline should have the same actual probability of being sent. More about the characteristics of the different traffic classes can be found in Chapter 4.1.1.

The issue of network scalability is not a major concern in this case as the number of nodes in an embedded system is rather fixed and new nodes are added negligibly seldom. The embedded system that this architecture is aimed for is thought to have a maximum of 64 nodes that will be connected by the AWG-based communication system.
4.1.1 Real-time traffic classes

The network proposed in this thesis is required to handle different traffic classes, in other words, traffic with different demands regarding real-time guarantees. Traffic can generally be divided into three groups: hard real-time traffic, soft real-time traffic and nonreal-time traffic.

Hard real-time traffic has to meet strict timing requirements. Failing to meet deadlines can have serious consequences for the entire system and a transmission that misses its deadline has lost its value. In many cases, it is equally disadvantageous to have data arrive too late at their destination as not sending the data at all. So it is vital to be able to guarantee the success of a transmission in advance, which is made possible by the help of the schedule provided by the MAC protocol. However, this is under the assumption of error-free transmission.

Soft real-time traffic also has deadline constraints, but meeting them is not vital to the system. After having transmitted all of the hard real-time traffic, the remaining bandwidth can be used for soft real-time traffic. Timeliness does not have to be guaranteed, but the system should try its best to send the data on time and, in some sense, to give priority over nonreal-time traffic.

Nonreal-time traffic can be described as traffic with an infinite deadline. Those messages can, e.g., be queued until the network capacity allows them to be sent without disturbing any real-time traffic.

4.2 Network architecture

The proposed network architecture, which is shown in Figure 8, consists of one \( N \times N \) AWG, one control node, further denoted as protocol processor (PP), and \( N - 1 \) end nodes. The AWG component is the centre of a physical WDM star network. The benefit of exchanging a normal PSC with the AWG, and of sacrificing one node for scheduling calculations, is the efficient support for deterministic services. More concretely described, simultaneous transmission of control information from all end nodes is implemented in order to gain short scheduling delays in the network. One of the nodes functions as a control node (PP) and is incorporated together with the AWG component in a hub. All \( N \) nodes, including the protocol processor, are connected to the \( N \times N \) AWG by two fibres, one for transmission and one for reception. All fibres used between the nodes and the AWG are defined to be of equal...
length in order to result in the same propagation delay for each packet which is send between any source-destination pair.

The network is based upon the principle of CC-TT-TR-FT-FR [Mukherjee 1992a]. The fixed-tuned receiver and transmitter are used to establish physical control channels between the end nodes and the protocol processor. Each of the $N-1$ fixed-tuned transmitters will send on an individual wavelength and each fixed-tuned receiver will be fixed to the corresponding wavelength, i.e., a node's fixed-tuned transmitter will send on the same wavelength as its fixed-tuned receiver receives on. All traffic from and to the fixed-tuned transceivers are control packets. On the control channel, the control processor receives all information from the end nodes that it needs in order to run its scheduling algorithm, and it sends out control information back to all the other nodes. This means that all intelligence about real-time traffic scheduling is embedded in the protocol processor, while the other nodes

*Figure 8. The network architecture.*
and the AWG component are completely passive in this respect. However, the end nodes do schedule their packet queues. Of the $N$ possible wavelengths out from each node, one is always used to send control information as described above. In addition to the control wavelength, one of the remaining $N-1$ wavelengths at each end node can, at each instance, be used for transmission of data traffic. The AWG is used to route the packets from source to destination based on the chosen wavelength.

The protocol processor transmits and receives data via wavelength array components, while the other $N-1$ nodes are equipped with one tuneable transmitter and receiver and one fixed-tuned transmitter and receiver. The array component on the transmitting side of the PP is attached to an $(N-1) \times 1$ combiner, which multiplexes the $N-1$ different wavelengths onto one fibre. Correspondingly, a $1 \times (N-1)$ splitter separates the incoming wavelengths before they enter the $N-1$ array of receivers of the protocol processor (see Figure 9). Due to the fixed size of the array components used at the PP, it is not possible to dynamically expand the number of nodes in the network. This lack of scalability is, however, not considered to be a problem in this case, since the target applications are embedded systems with networks that normally do not have to grow during operation.

The fixed-tuned and tuneable transmitter on each of the $N-1$ remaining nodes are connected to a $2 \times 1$ combiner in each node which multiplexes the two wavelengths (one fixed wavelength to the protocol processor and a tuneable one for data transmission to any of the remaining nodes) onto one fibre. The corresponding design on the receiving side of each node is a $1 \times 2$ splitter that is connected to the fibre with incoming traffic which is distributed to the node’s two receivers, the fixed and the tuneable one (see Figure 10).

![Figure 9. The $(N-1) \times 1$ combiner and the $1 \times (N-1)$ splitter at the PP.](image)
All of the end nodes are able to send and receive data traffic on all the available wavelengths due to their tuneable transmitters and tuneable receivers. As none of the end nodes has global knowledge of which other node will send in which time slot and on which wavelength, the PP has to employ a MAC protocol in order to organize the data traffic in a centralized fashion. The main task of the MAC protocol is to delegate the available wavelengths between the sending nodes based on control messages from these nodes to the PP, and to make sure that the hard real-time message with the shortest deadline is sent first.

The proposed MAC protocol assumes an $N \times N$ AWG using one FSR containing $N$ contiguous wavelengths. The tuneable transceivers in all end nodes are tuneable over those $N$ wavelengths. The array components in and out of the PP are able to receive and send on a range of $N-1$ wavelengths.

Each of the $N-1$ end nodes has different message queues for hard, soft, and nonreal-time messages; nonreal-time is in this context defined as soft real-time with an infinite deadline. The nodes sort the messages in the queues according to deadline, i.e., the message with the shortest deadline will be placed first in its queue and thereby assigned the highest priority. The EDF scheduling algorithm is chosen due to its optimality in certain circumstances [Liu et al. 1973]. After completing the sorting process, the node sends out a control message to the protocol processor. This control message sent by the end nodes contains information about one or more packets in its queue, depending upon which of the developed MAC protocols it is using. In more detail, it will specify source node, destination node, deadline and traffic class (hard, soft or nonreal-time) for the packet(s) in question. More details on how the packets
specified in the control packet are chosen can be found in the detailed descriptions of the control traffic in Chapters 5.2 and 6.2.

All traffic, both control and data traffic, is organized in time slots, where the duration of each slot corresponds to the transmission time of one (maximum sized) data packet, i.e., the time measured between when the first and the last bit of the packet leaves the source node. However, tuning times might not be negligible, so this delay has to be added to the length of one time slot. As control packets are fairly short, because of the small amount of information they contain, and as short distances, and thereby a short propagation delay, in the network are assumed, one time slot provides enough time to send control traffic to the PP, for the PP to run the scheduling algorithm, and to send control messages back to the end nodes. At the end of each time slot, all end nodes tune their respective transmitters and receivers accordingly. Data transmissions take place in the following time slot.

Due to the fixed-tuned transceivers at each node and the individual wavelength of the control channel, it is possible to handle data traffic and control traffic in parallel during each time slot. Furthermore, the AWG allows control traffic from all the end nodes to be sent simultaneously. Therefore, the medium access control does not result in more than one single time slot of delay (see Figure 11).

After having sorted their queues according to the EDF algorithm, each node gives highest priority to the first packet in the hard real-time queue(s). Only in case of an empty hard real-time queue, any soft or nonreal-time packets are considered.

The protocol processor has the responsibility of accepting or denying each node’s request to send. In order to avoid receiver collisions, there must be no more than one data transmission to any single destination in any given time slot. When the protocol processor has received control messages by all the end nodes, it sorts those messages by deadline with the shortest deadline given highest priority. It checks the control

![Figure 11. Control packet propagation.](image-url)
messages one by one, according to priority, and determines whether the requested transmission can be accepted or not. As soon as the transmission of a data packet to a certain destination is accepted, any other request for communication with this destination in the next slot is denied, as this request would lead to a receiver collision. A corresponding procedure is necessary to avoid that two packets from the same sending node are scheduled for the same time slot in the protocol presented in Chapter 6. As soon as the algorithm has found the right set of data packets to send during the next time slot, the PP sends out an individual control message simultaneously to each node. In this control packet, each node will be informed about whether it is allowed to send in the up-coming time slot, if it will receive any data traffic and, in that case, from which node.

In order for the traffic model to work properly and avoid collisions, the nodes have to be synchronized. Synchronization in the proposed AWG network is easier than in many other networks because the protocol processor has the possibility to send out traffic, in this case control packets, to all end nodes simultaneously through its transmitter array. As all the fibres between the nodes and the AWG component are assumed to be of equal length, the nodes are synchronized on incoming control packets. Further, more exact or more frequent, synchronization is not needed. As each node has its own control channel, clock synchronization on the packet level is not required. Interconnection distances in the targeted networks are assumed to be short, so the difference in propagation delay due to dispersion is negligible. (A discussion of timing and dispersion in WDM star networks can be found in [Semaan et al. 1993].)

For error detection, the implementation of a checksum or similar on a lower layer is assumed, as no mechanism for error checking of the control packets is included in the presented MAC protocol. In case of an error in the control message so that the protocol processor or end node cannot use the information, a retransmission has to be attempted in the next time slot. If a control packet is lost on the way from the protocol processor to an end node, the end node simply has to send the request once again in the next time slot. Additionally, the transmission to this end node announced in this control packet will probably experience an error as the end node will not be tuned to the right wavelength (except for the case that it will receive from the same source as in the ongoing time slot). This will add a delay of at least two time slots for this packet as its transmission will have to be requested again. If the control packet from the end node does not reach the protocol processor, the PP will still send out a control packet as usual, leaving the first part of the control packet empty. The end node will interpret this answer simply as a denied request and the requested transmission will have an added delay of (at least) one time slot.
5 Using the Real-Time AWG Network in System Area Networks

This chapter describes in more detail the network and protocol design solution for a short range communication system like, e.g., a SAN [Böhm et al. 2004] [Böhm et al. 2005b] [Kunert et al. 2005a]. The main focus lies on guaranteed support for hard real-time traffic at the same time as soft real-time traffic present in the system is supported as well. The single-hop network with the AWG at its centre uses control channel-based, centralized traffic scheduling. The AWG’s property of spatial wavelength reuse and the combination of fixed-tuned and tuneable transceivers in the nodes enable simultaneous control and data transmission. A case study with defined real-time communication requirements for a RSP application was carried out in order to investigate the suitability of the suggested solution for this kind of application.

5.1 Network architecture

The suggested network architecture for SANs, as earlier described in Chapter 4.2, is a single-hop star network with the AWG as the central component. Together with the PP it is integrated in a hub, and the PP and the end nodes communicate through physical control channels. All end nodes have a pair of fixed-tuned and a pair of tuneable transceivers, where the fixed wavelength is used for the control communication. Each node uses an individual control wavelength. In each node there are two message queues, one for hard real-time traffic and one for soft and nonreal-time traffic (see Figure 12). Again nonreal-time traffic in this context is seen as soft real-time traffic with infinite deadlines.

5.2 Protocol details

As the nodes do not have global knowledge of which other nodes that want to send in which time slot and on which wavelength, the protocol processor has to employ a MAC protocol and organize the traffic in a centralized fashion. The main task of the MAC protocol is to delegate the available wavelengths between the sending nodes based on control information, and to make sure that the hard real-time message with the shortest deadline is sent first. As the presented system is developed for a SAN, the fibres interconnecting the end nodes and the protocol processor are assumed to have a maximum length of a few tens of meters. In other words, the propagation delay is assumed to be a fraction of a time slot.
5.2.1 The control traffic

Each end node sends out a control message to the protocol processor on its control channel, an individual, fixed control wavelength. The node chooses the first packet from the hard real-time queue, or, in case the hard real-time queue is empty, the first packet from the soft real-time queue, and sends a control message containing information about the packet. The information contained specifies the identity of the source node and the destination node and furthermore also specifies the deadline of the packet in question and its traffic class. The protocol processor’s decision on which packet is allowed to be sent is based upon a simple algorithm explained in Chapter 5.2.2.

The control traffic is organised in two phases, i.e., control packets on their way to the protocol processor and control packets on their way from it. These two phases span one time slot, defined by the length of the transmission time of one (maximum sized)
data packet, as mentioned earlier. The control messages sent from the protocol processor to the end nodes contain two pieces of information. The first part of the control message tells the node whether the request to send the packet in the next time slot is accepted or not. The second part of the control message from the protocol processor to the end nodes contains information about whether they can expect any traffic in the following slot, and if so, from which other node. If any specific node is not a destination for data traffic in the next time slot, this second field will be empty.

As there is only one single, dedicated wavelength connecting each pair of nodes, each node knows to which wavelength to tune its transmitter and receiver as soon as it has the information about what node(s) it is to communicate with in the following time slot. After the tuning of the transceivers, the nodes are ready for transmission and reception which takes place in the next time slot. Nodes that are not allowed to send their requested data and that are no destination node either, receive an empty control packet from the protocol processor for synchronization reasons.

After having received a control packet, each node tunes its tuneable transmitter to the wavelength dedicated to the destination node for the accepted message scheduled to be send during the following time slot. At the same time, also the destination node has to tune its tuneable receiver to the specified wavelength in order to be able to receive the scheduled data. Each node can send and receive traffic at the same time. In the time slot that follows, each node sends the message it has got permission to send via the AWG to the destination, and receives any potential traffic. During the same time slot, new control packets from the end nodes reach the PP and the scheduling algorithm produces a new traffic scheme for the next time slot coming up.

The parallel handling of data and control traffic is possible because of the individual, fixed wavelength of the control channel from each node, as it therefore does not interfere with data traffic sent on any of the remaining $N - 1$ wavelengths. Furthermore, the characteristic properties of the AWG allow simultaneous control traffic from all the nodes to be sent to the PP. Therefore, the medium access control results in no more than one single time slot of delay, and does not have any impact on the data traffic flow. This principle is illustrated in Figure 11, Chapter 4.3.

*Timing analysis*

For a $64 \times 64$ AWG, the length of a control packet, to the protocol processor is less than 100 bits: 6 bits signifying the source node, 6 bits to identify the destination node, 20 bits to state the deadline, 1 bit indicating the traffic class, and some additional bits of overhead. Under the assumption of a bit rate of 2.5 Gbit/s, this results in a transmission time of 40 ns, denoted by $T_{\text{pack1}}$. The control message from the PP to an end node is even shorter: one flag bit, to designate whether the request to send has
been accepted or not, and 6 bits to identify the sending node from which it will receive traffic in the next time slot. $T_{pack2}$ denotes the transmission time of this control packet and, in this case, is 20 ns. At the end of the control slot, the transceivers at the end nodes are tuned according to the information in the control packet. The tuning time, $T_{tune}$, is approximated to 100 ns. Still assuming a bit rate of 2.5 Gbit/s, a processing time ($T_{proc}$) of 100 ns at the PP, and defining the length of one time slot ($T_{slot}$) to be 1 $\mu$s, i.e., 2500 bits as the maximum data packet length, results in the following calculation of the propagation time, denoted by $T_{prop}$:

$$T_{prop} = \frac{T_{slot} - T_{pack1} - T_{proc} - T_{pack2} - T_{tune}}{2} = 370 \text{ ns}. \quad (1)$$

In other words, an assumed length of a time slot of 1 $\mu$s leaves enough communication time to support a fibre length of 74 m, assuming a propagation speed of $2 \times 10^8$ m/s, between the PP and each regular node (see Figure 13 for details). The speed of light through a fibre is around two thirds of the speed of light in vacuum, which is approximately $3 \times 10^8$ m/s.

### 5.2.2 The scheduling algorithm

Both the end nodes and the PP are involved in the decision on which node is to send any data and during which time slot. The end nodes decide which packet has to be sent next, while the PP is responsible for distributing the network resources by deciding which requests from the nodes to accept or reject.
Each end node has two packet queues, one for hard real-time traffic and one for soft and nonreal-time traffic, which are sorted according to deadline, i.e., by EDF. After having sorted their two queues according to the EDF algorithm, each node has to decide on giving priority either to the first packet in the hard real-time queue or the first in the soft real-time queue. In order to be able to give guarantees on delay bounds for hard real-time traffic, this traffic always is to be sent first. In other words, as long as there are packets queued in the hard real-time queue, the soft real-time traffic has to wait. This guarantees that the hard real-time traffic is not disturbed or compromised in any situation by lower priority traffic. The protocol processor will treat the incoming requests as described in Chapter 4.3.

### 5.3 Case study with simulation analysis

In order to verify the feasibility of the proposed network and protocol, the system was evaluated according to specific application requirements. A simulation based on a radar signal processing (RSP) case was carried out and delay, throughput and deadline miss ratio for the network were analysed. RSP is an application area requiring System Area Networks with support for heterogeneous real-time services. With its three traffic classes, the proposed AWG-based network suits this purpose very well.

#### 5.3.1 Case definition

In [Bergenhem et al. 2002], Bergenhem et al. describe a full case definition for a RSP case scenario. The evaluation described in the following is inspired by the straight pipeline case, one of three possible alternatives. Due to its clear traffic pattern, this case simplifies analysis and understanding of the simulation results, while still requiring three traffic classes.

![Figure 14. Master node and straight pipeline of slave nodes according to RSP case definition.](image)
A network architecture with a $16 \times 16$ AWG at its centre, consisting of one PP and fifteen communicating end nodes, was chosen for the simulation of the RSP system. One node serves as a master node, while the remaining fourteen slave nodes are used for the pipelined data flow (see Figure 14). Further assumptions made are periodic traffic, a propagation speed of 2.5 Gbit/s and a time slot length of 1 $\mu$s, i.e., one time slot corresponds to one data packet of 2500 bits. In the context of this simulation, the delay caused by tuning was defined to be negligible. The traffic in the RSP case definition consists of three main traffic types: control traffic, which flows in two directions between the master node and each slave node in the straight pipeline, data traffic from node to node in a pipelined manner, and other traffic, such as logging of data or long term statistics from the pipeline nodes to the master node. These three traffic types and their individual deadline requirements are conveniently represented by the three traffic classes in the suggested protocol: hard real-time, soft real-time and nonreal-time traffic. The relative delay bounds for the three traffic classes are equal to their individual period times.

5.3.2 Simulation results

The parameters chosen for analysis are throughput, average delay and deadline miss ratio of the traffic classes by using both fixed, predefined traffic parameters, and by varying the amount of SRT or HRT traffic. The predefined traffic class parameters are listed in Table 1, while the simulation results for this set of parameters is presented in Table 2. All traffic that any channel will send during one period is generated and queued for sending at the start of the period. There is no smoothing of the incoming traffic over the whole period. As seen in Table 2, there are no deadline misses for neither soft nor hard real-time traffic.

Simulating HRT traffic only leads to a throughput of 0.28 HRT packets per time slot, which is the maximum possible for HRT traffic as can be concluded from the 0.00 deadline miss ratio. This means that all generated HRT traffic is delivered in time. Solely SRT traffic in the network leads to a throughput of 11.20 packets per slot. Using all three of the traffic classes, with the parameters stated above, leads to a total throughput of 12.71 packets per slot, which comprises 0.28 HRT, 11.20 SRT and 1.23 NRT packets per time slot. This indicates that, in spite of three different traffic types in the system, the maximum throughput for the HRT and SRT traffic can be reached. As will be discussed later in Chapter 7.1, in case of the period equalling the deadline, a throughput guarantee of one packet per slot can be given for HRT traffic, and therefore its deadline miss ratio at the simulated traffic loads (smaller than one) will always be zero.
### Table 1: Traffic type parameters

<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>Traffic Class</th>
<th>Communication Pattern</th>
<th>Period [slots]</th>
<th>Payload [packets]</th>
<th>Delay Bound [slots]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>HRT</td>
<td>Master/Slave</td>
<td>100</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>Data</td>
<td>SRT</td>
<td>Straight Pipeline</td>
<td>5000</td>
<td>4000</td>
<td>5000</td>
</tr>
<tr>
<td>Other</td>
<td>NRT</td>
<td>Many-to-one</td>
<td>5000</td>
<td>950</td>
<td>None</td>
</tr>
</tbody>
</table>

### Table 2. Throughput, average delay and deadline miss ratio at fixed traffic loads.

<table>
<thead>
<tr>
<th></th>
<th>Throughput [packets/slot]</th>
<th>Delay [slots/packet]</th>
<th>Deadline Miss Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total *</td>
<td>12.71</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HRT *</td>
<td>0.28</td>
<td>4.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SRT *</td>
<td>11.20</td>
<td>2057.00</td>
<td>0.00</td>
</tr>
<tr>
<td>NRT *</td>
<td>1.23</td>
<td>6225.80</td>
<td>-</td>
</tr>
<tr>
<td>Max HRT**</td>
<td>0.28</td>
<td>4.64</td>
<td>0.00</td>
</tr>
<tr>
<td>Max SRT***</td>
<td>11.20</td>
<td>2000.50</td>
<td>0.00</td>
</tr>
</tbody>
</table>

*) Using the traffic load stated above  
**) HRT traffic only  
***) SRT traffic only

Figures 15 to 19 show throughput, average delay and deadline miss ratio when changing the traffic load in the system. The traffic load is increased for all traffic classes at the same rate. The curves are attained by running the simulator for 20000 time slots and the statistical results are computed from slot 5000 and forward.

Figures 15 to 17 indicate the system’s behaviour when varying the SRT traffic from 500 to 5000 packets per channel with a period of 5000 slots, while the HRT traffic is kept at a constant level of one data packet per channel and period of 100 slots. A constant amount of nonreal-time traffic (950 packets per period of 5000 slots) is
introduced to make use of the remaining bandwidth that is not needed by the two real-time classes.

The throughput of the three traffic classes, together with the total throughput, is shown in Figure 15. As the intensity of the SRT traffic increases, the throughput of this traffic class reaches its maximum of 13.6 packets per slot at a traffic intensity of 5000 packets per period. The throughput of the HRT traffic is not affected and remains constant at the maximally possible 0.28 packets per slot. Due to its high priority, it is not affected by the increasing SRT traffic intensity or the NRT traffic in the system. The NRT traffic, on the other hand, is starved as the network starts to get saturated at an SRT traffic intensity of about 3000 packets per channel and 5000 slots.

Figure 16 shows the average delay of the traffic classes. As the traffic load in the system increases, the SRT traffic experiences a slight increase in delay, while the HRT traffic remains at a constant average delay of 4.64 slots per packet throughout the duration of the simulation. Because of its low priority, the NRT traffic has to wait for the real-time traffic classes to be sent, which increases its delay considerably. For high
SRT traffic intensity, NRT traffic gets starved and too few packets are sent to provide statistically reliable results.

Figure 17 verifies that the HRT traffic meets all its deadline requirements. In spite of the increasing SRT traffic load, the deadline miss ratio of the HRT traffic remains zero, while the deadline miss ratio of the SRT traffic increases slightly to 6% as SRT traffic intensities reach close to 5000 packets per period.

In the second simulation setup, the results of which are shown in Figure 18 and 19, the HRT traffic is varied between 1 and 16 data packets per channel with a period of 100 slots. This time, the SRT traffic intensity remains constant at 4000 packets per period of 5000 time slots, while the NRT traffic load is identical with that in the previous simulation setup.

Figure 18 shows how varying the HRT traffic intensity influences the three traffic classes in terms of throughput. An increasing HRT traffic load leads to a steady increase in HRT throughput, while SRT and NRT traffic experience a slight decrease in throughput due to their lower priority. The total throughput reaches its maximum

Figure 16. Average delay of the different traffic classes against SRT traffic intensity.
value at about 13.2 packets per slot at a HRT traffic intensity of 7 packets per 100 time slots.

The behaviour of the delay in Figure 19 can be explained accordingly. With an increased HRT traffic intensity, the traffic class with the highest priority, HRT traffic, is affected the least, while the remaining two traffic types experience a more significant increase in delay. The average delay for the NRT traffic is not plotted in the figure, as the network already is saturated with NRT traffic which would lead to irregularities in the curve due to insufficient statistical data.

The deadline miss ratio of the HRT and SRT traffic are not affected by the increasing HRT load and the increasing delay, and remain at zero throughout the simulation (not shown in a figure). This indicates that the HRT and SRT capacity of the network is not challenged by the imposed traffic intensities.

Figure 17. Deadline miss ratio of HRT and SRT traffic against SRT traffic intensity.
Figure 18. Throughput of the different traffic classes and total throughput against HRT traffic intensity.
Figure 19. Average delay of the different traffic classes against HRT traffic intensity.
6 Using the Real-Time AWG Network in a Distributed Optoelectronic Router

In this chapter, the network architecture and MAC protocol are enhanced for the usage in a distributed router instead [Böhm et al. 2005a][Kunert et al. 2005a][Kunert et al. 2005b]. Even in this application, guaranteed support for hard real-time traffic is essential, while soft real-time traffic still has to be supported. The traffic pattern in a router is naturally dependent upon the applications using the router, but the possibility of bursty traffic or traffic arriving at another rate as the internal speed in the router has to be taken into consideration. The network is still a single-hop star with an AWG at its centre and a PP is responsible for the scheduling of the traffic. Control communication between the PP and the end nodes is sent through physical control channels, and data transmissions are possible in parallel due to the advantageous properties of the AWG. This chapter is concluded by a simulation study of periodic real-time traffic in the suggested solution.

Figure 20. Example of a router architecture.
6.1 Router architecture

A general router can be viewed as a number of input and output ports and their corresponding queuing systems (see Figure 20). The switch fabric is the heart of the router and is responsible for carrying packets from input ports to desired output ports. In low-performance routers, the switch fabric is often just a simple bus, while high-performance routers often have some kind of complex multistage interconnection network to implement an experienced all-to-all connectivity [Tse 2005].

In the suggested router architecture, the AWG network acts as the switch fabric, while the nodes in the network correspond to the router’s input and output ports. The network architecture consists of one $N \times N$ AWG, integrated in a hub with the protocol processor used for centralized scheduling, and $N-1$ pairs of input and output ports (see Figure 21). The AWG component acts as the centre of a physical WDM star network, and the PP and each pair of input and output ports are connected to the AWG by two fibres, one for transmission and one for reception.

The difference compared to the SAN’s network architecture is twofold. One, a memory to store control information has been added to the PP, and two, the queue architecture in the end node has been changed. Each of the router’s input ports has four different message queues per possible output port. One queue is for HRT messages whose control information is already stored at the PP and one queue is for the remaining hard real-time packets, further on denoted as control queue and noncontrol queue respectively. Corresponding queues exist for soft and nonreal-time traffic, where nonreal-time traffic still is defined as SRT traffic with infinite deadline.

6.2 Protocol details

The main task of the MAC protocol is to delegate the available wavelengths between the router’s input and output ports based on control information, and to make sure that the hard real-time message with the shortest deadline is sent first. The fibres interconnecting the ports and the protocol processor are assumed to have a maximum length of a few tens of meters, which would constitute the case of a distributed router. In other words, the propagation delay of each packet is assumed to be a fraction of the length of a time slot. An even more compact router with even shorter fibres is of course possible as well.
Figure 21. The proposed router and queuing architecture.
6.2.1 The control traffic

As already described in Chapter 4, the PP needs to receive information about the packets to be scheduled for the following time slot. Therefore, in each time slot, each input port sends a control packet to the PP over the dedicated control channel. In this design, the PP has a memory to store information about packets that did not get permission to be sent. Due to the memory, no control information has to be sent twice. In each time slot, an input port can receive one or more new packets that are put in the queue that corresponds to its destination and the desired traffic class. The decision to introduce multiple queues was made in order to avoid the effects caused by the head-of-line effect of single queues, where the packet at the head of the queue blocks all packets behind itself. The possibility of sending control information about multiple packets at a time was added to accommodate traffic entering the router at a higher data rate than the internal router speed.

The noncontrol queues are sorted by deadline, and a control message with information about four packets is sent. The four packets chosen are those that are at the head of their noncontrol queue, and at the same time have one of the four shortest deadlines amongst those. Information about those four packets is sent to the PP, requesting permission to send data. The HRT queues are always checked first. In case of empty HRT queues, the SRT queues are considered. The control message contains information about traffic class, source port, destination port and deadline for each of the four packets in question. Even here, as the control packet is simple and contains little information, one time slot can be used to send control traffic to the PP, to run the scheduling algorithm at the PP, and to send control messages back to the ports.

Timing analysis

Again looking at the case of a $64 \times 64$ AWG, the length of a control packet to the PP in this case is less than 200 bits, including 1 bit for the traffic class, 6 bits for specifying the source, 6 bits for identifying the destination, and 20 bits to state the deadline, all for each of the four packets, plus some bits of overhead. At an assumed bit rate of 2.5 Gbit/s, the transmission time of a control packet to the PP, $T_{\text{pack1}}$, is 80 ns. The control message from the PP back to the ports is even shorter as its approximately 50 bits include 6 bits to state the destination for the accepted packet and 6 bits to identify the sending port from which the end node will receive during the next time slot, plus some bits of overhead. This corresponds to a transmission time, $T_{\text{pack2}}$, of about 20 ns. Accessing the protocol processor’s memory to fetch and store information about unscheduled packets can be done during the transmission
time of the control packets and does not add any additional delay. At the end of the slot, the pairs of input/output ports tune their respective transmitters and receivers according to the control information, which takes about 100 ns, denoted by $T_{tune}$.

Assuming a bit rate of 2.5 Gbit/s, a 400 ns processing time ($T_{proc}$) at the PP and defining the length of one slot, $T_{slot}$, to be 1 $\mu$s, i.e., 2500 bits as the maximum data packet length, leads to the following calculation of the allowed propagation time, $T_{prop}$, between a router port and the PP:

$$T_{prop} = \frac{T_{slot} - T_{pack1} - T_{proc} - T_{pack2} - T_{tune}}{2} = 200 \text{ ns}$$ (2)

In other words, a time slot length of 1 $\mu$s leaves enough communication time to support a fibre length of 40 m, assuming a propagation speed of $2\times10^8$ m/s, between the PP and any router port (see Figure 22).

Data transmissions again take place immediately in the following time slot. It is still due to the fixed tuned and individual wavelength of the control channel that data and control traffic can be handled in parallel. Even in this system the AWG allows control traffic from all the input ports to be sent to the PP simultaneously and, therefore, the medium access control can be limited to one single slot of delay. Synchronization is accomplished in the same way as explained earlier in Chapter 4.3).
6.2.2 The scheduling algorithm

Each of the router’s input ports sorts all its noncontrol queues according to the EDF algorithm, sends information about its four chosen packets to the PP and moves the packets to the corresponding HRT control queue. As long as there are packets queued in a noncontrol HRT queue, traffic in the SRT queues has to wait, guaranteeing prime attention to the HRT traffic and thereby ensuring that it is not disturbed or compromised by SRT traffic in any situation.

Upon reception, the PP sorts the control messages together with the control information stored in its memory by deadline, checks them one by one, and determines which of the requested transmissions can be accepted without causing a sender or receiver conflict. To avoid those collisions, there must not be more than one transmission from any given source or to any given destination during any given time slot. As soon as a transmission from a certain input port or to a certain output port is accepted, any other request for communication with this source or destination respectively is denied. Having decided on a set of accepted requests, the PP sends out individual messages to all the ports. The PP will remove the accepted requests from its memory, while the remaining requests will be stored. When the input port receives information about which output port it is permitted to send traffic to, it picks the packet out of the control queue and sends it to the intended destination.

6.3 Simulation analysis

Simulations were carried out to analyze delay, throughput and deadline miss ratio of the HRT and SRT traffic by varying the intensity of the two real-time traffic classes respectively. For the simulations, a system model with a $64 \times 64$ AWG was chosen, which leads to a router with 63 input ports and 63 output ports, saving one pair of input and output ports at the AWG for the PP. Further, periodic traffic is assumed for both traffic classes with a period and deadline of 100 time slots. The curves are attained by running the simulator for 1000 time slots and the statistical results are computed from slot 400 and forward.

Periodic traffic channels in the simulator are treated as follows. In order to simulate incoming traffic over the whole period, a random offset is computed that decides in which time slot in each period a certain channel will be requested for transmission. The distribution of source addresses and destination addresses is randomized with an even distribution. Certain further assumptions were made.
All data messages are equally long with a transmission time of one time slot.

The queues in all ports and in the PP’s memory are of infinite length.

Packets are generated at the beginning of a time slot.

Delay and deadline are expressed in number of time slots, from the point in time when a packet is generated until its transmission starts.

The tuning delay is negligible.

Figures 23 to 25 show the router’s behaviour when keeping the SRT traffic intensity at a constant rate of 10%, i.e., 630 packets per period of 100 time slots. The HRT traffic intensity is increased from 0% to 90%, i.e., 5670 packets per period of 100 time slots. At the highest HRT traffic rate, the total traffic load in the system is 100%.

Figure 23 shows that an increasing HRT traffic load leads to steady increase in HRT throughput, while the SRT traffic remains on a constant level before experiencing a slight decrease at HRT rates higher than 4410 packets per period. Due to its lower priority, SRT traffic is filling the capacity of the network not needed by HRT traffic.
and at high HRT traffic intensities, SRT packets get starved. The total throughput reaches a value of about 58 packets per slot.

The behaviour of the average delay in Figure 24 can be explained accordingly. With an increased HRT traffic intensity, the prioritized HRT traffic is affected the least, while the SRT traffic type experiences a higher increase in average delay.

As can be seen in Figure 25, the slight increase in average delay at high HRT traffic intensities does not affect the HRT deadline miss ratio. This shows that the HRT capacity of the network is not challenged by the imposed traffic intensities. The sudden increase in average delay for SRT packets, as seen in Figure 24, leads to a deadline miss ratio of about 0.27 at the maximum HRT traffic intensity of 5670 packets per period.

The second set of figures, Figures 26 to 28, are obtained by having a constant HRT traffic rate of 10%, i.e., 630 packets per period, while the SRT traffic intensity is varied between 0% and 90%, i.e., between 0 and 5670 packets per period.

*Figure 24. Average delay of HRT and SRT traffic against HRT traffic intensity.*
The throughput of the two traffic classes, together with the total throughput, is shown in Figure 26. As the intensity of the SRT traffic increases, its throughput reaches a maximum of about 52 packets per time slot at its highest traffic intensity (5670 packets per period). Due to its high priority, the throughput of the HRT traffic is not affected and remains constant at 6.3 packets per time slot. Comparing the total throughput in this figure with the total throughput in Figure 23 shows that both cases have a total maximum throughput of about 58 packets per node and slot.

Figure 27 shows that the SRT traffic experiences an increase in delay, as the traffic load in the system increases, while HRT traffic remains at a delay of about 1.1 slots, not including actual transmission of the packet in question, per packet for all simulated SRT traffic loads. Due to its higher priority, it is not affected by the changing level of SRT traffic intensity.

Figure 28 verifies that the HRT traffic meets all its deadline requirements. In spite of the increasing SRT traffic load, the deadline miss ratio of the HRT traffic remains zero, while the SRT deadline miss ratio increases considerably to 18% as its intensity approaches 90%.

*Figure 25. Deadline miss ratio of HRT and SRT traffic against HRT traffic intensity.*
Figure 26. Throughput of HRT and SRT traffic and total throughput against SRT traffic intensity.
Figure 27. Average delay of HRT and SRT traffic against SRT traffic intensity.
Figure 28. Deadline miss ratio of HRT and SRT traffic against SRT traffic intensity.
7 Improved hard real-time traffic support

Supporting hard real-time traffic means being able to guarantee the throughput of a certain number of packets with timing constraints. Therefore, in order to determine the performance characteristics for hard real-time traffic, a deterministic analysis of the delay of the proposed network architecture is given below [Kunert et al. 2008a][Kunert et al. 2008b]. The required input parameters for the analysis are the traffic characteristics of a given application. Those will be provided in the form of real-time traffic flows, or real-time channels, where each flow $q$ has the following attributes

- source $S_q$
- destination $D_q$
- period $TP_q$ (minimum message inter-arrival time)
- deadline $TD_q$
- capacity $C_q$ (maximum packet length)

The total number of flows in the network is denoted by $Q$, and $1 \leq q \leq Q$. In order to increase the number of guaranteed packets which can be transmitted simultaneously over the single-hop network, traffic dependencies are analyzed on a per flow level. The goal of this method is to determine which packets can be sent simultaneously in case not all nodes want to send to the same destination node.

7.1 Real-time analysis

The task of analysing the interdependencies of these traffic flows, which resources they use and if they constitute a feasible system, i.e., if a schedulable traffic allocation is possible, can be mapped onto the problem of uniprocessor task scheduling [Hoang et al. 2003]. This means that during the course of this analysis, those traffic flows are looked upon as synchronous, periodic tasks which have to be scheduled on the network. The capacity $C_q$ of a traffic flow corresponds to the worst-case execution time (WCET) of the task to be scheduled.

Assuming a worst-case scenario, i.e., when all of the nodes in the suggested network want to send data traffic to the same destination in any given time slot, there is always at least one packet, namely the one with the earliest deadline, that can be guaranteed access to the medium due to the assumption of EDF scheduling in the end nodes and the PP. This means that a capacity of 1 always can be guaranteed for hard
real-time traffic. Assuming periodic traffic, basic EDF theory [Liu et al. 1973] can be used to analyse the system. According to EDF scheduling theory, the utilisation of a hard real-time system is defined as:

\[
U(q) = \sum_{q} \frac{C_q}{TP_q} \leq U_{\text{max}}.
\]  

(3)

Translated into this communication context, this means that \( U \) defines the utilization of the periodic traffic in the network, where \( C_q \) is the amount of data per period, \( TP_q \) is the period of the data traffic, and \( U_{\text{max}} \) denotes the maximum utilisation of the network by hard real-time traffic that must not be exceeded.

For further description of the feasibility test in detail, the following concepts from the area of real-time task scheduling have to be introduced into the discussion.

* **Hyperperiod**
  Given a task set consisting of periodic tasks, the hyperperiod is the least common multiple of all periods of those tasks, i.e., the length of time from when all tasks’ periods start at the same time until they start at the same time again. In the context of communication, a task corresponds to a communication channel.

* **Busyperiod**
  A busyperiod is any interval of time in which the resource is not idle, i.e., the busyperiod of a communication link is generally any time interval during which the link is not idle.

* **Workload function**
  The traffic demand on the analyzed link corresponds to the processor demand in a real-time system, and can be defined by a workload function, \( h(t,q) \). It is the sum of all the capacities of the instances of tasks \( q \) with an absolute deadline less than or equal to a point in time \( t \), where \( t \) is the time elapsed from the start of the hyperperiod. Mapping this on the communication scenario, the function is summing up the maximum packet lengths per period, \( C_q \), of all instances of real-time channels \( q \) that have an absolute deadline less than or equal to \( t \). The workload function is calculated as follows [Baruah et al. 1990a][Baruah et al. 1990b][Spuri 1996].

\[
h(t,q) = \sum_{t \leq TD_q} \left[ \frac{t + TP_q - TD_q}{TP_q} \right] \cdot C_q
\]  

(4)
Feasibility tests investigate if a system is in a feasible system state, i.e., if all the logical tasks in the system are feasible. Following the discussion from above, feasibility testing in a communication context spells checking if the admission of an additional logical real-time channel would still result in a set of feasible real-time channels. The feasibility testing is performed in two steps, each being a test of its own. The following two constraints have to be met.

**Constraint 1**

*The utilization of the link has to be less than or equal to one (100%), following the previous discussion about guaranteed access to the medium.*

In order for the task set to be schedulable, or, in this case, for the flows to be allocatable over the network link, the utilization parameter has to be less than or equal to 1 (100%). This means that $U_{max}$ is equal to 1 and the constraint therefore defined as

$$U(q) = \sum_{q} \frac{C_q}{TP_q} \leq 1$$

This condition is necessary, but not sufficient to be able to ensure a 100% success rate for the transmission of real-time traffic with guaranteed deadlines. This means, as it is hard real-time traffic that is analyzed, solely the fulfilment of this utilization constraint will not be sufficient to guarantee that no deadlines will be missed. However, for the case of all deadlines being equal to or longer than their corresponding periods, this test is enough. For schedulability reasons, it is assumed in the analysis that the utilization of each single task, i.e., flow or channel, is no higher than 100%.

**Constraint 2**

*For all values of t, the workload function $h(t,q)$ has to be less than or equal to t.*

$$h(t,q) = \sum_{\forall Q \in \mathbb{Q}} \left( t + \frac{TP_q - TD_q}{TP_q} \right)\cdot C_q \leq t \quad \forall t$$

(6)
This second condition, introduced in [Baruah et al. 1990a][Baruah et al. 1990b] and generalized in [Spuri 1996], was added in order to insure the continued feasibility of the system when adding a new task, or in this case, a new traffic flow.

However, this feasibility analysis, as mentioned earlier, is developed for real-time systems and therefore assumes fully preemptive tasks. In network communication, packets normally cannot be preempted and therefore the possibility of further delay has to be taken into account. A constant $T_{\text{blocking}}$ is defined which denotes the maximum blocking time that one packet can introduce into the system, i.e., $T_{\text{blocking}}$ equals the transmission time of a packet with the maximum packet size. Furthermore, the sending and receiving of control information will introduce an additional delay, represented here by the constant $T_{\text{control}}$. In accordance with earlier reasoning, $T_{\text{blocking}}$ and $T_{\text{control}}$ will be equally long. These compensations result into a further shortening of the delay bound.

$$\text{T}_{D_q}' = \text{T}_{D_q} - T_{\text{blocking}} - T_{\text{control}}$$

This means that the workload function is remodelled as follows.

$$h(t,q) = \sum_{tD_q \leq t} \left( \frac{t + TP_q - \text{T}_{D_q}'}{TP_q} \right) \cdot C_q$$

Unfortunately, the second constraint, in the form given above, does not lend itself to calculation particularly well due to the high computational complexity it introduces into the feasibility test. It is shown in [Stankovic et al. 1998] how it is possible to reduce the time and memory complexity of the second constraint check by reducing the number of instances of calculation to include merely a reduced number of integer time values during an interval upperbounded by $BP_1$, the first busyperiod during the first hyperperiod of the schedule where all periods start at time zero. If $h(t,q) \leq t$ in the first busyperiod of the hyperperiod in the supposed schedule to come, then $h(t,q) \leq t$ for all $t$. The following upper bound of the interval to be checked is therefore an improvement of the algorithm above.

$$t:1 \leq t \leq BP_1 \quad \forall t \in N$$

Furthermore, one needs to check not each integer from the first time slot, but only the integers $t$ where
\[ t \in \bigcup_q \left\{ m \cdot TP_q + TD_q : m = 0, 1, 2, \ldots \right\} \]  
\[ \text{(10)} \]

where

\[ t \in [1; BP_1] . \]  
\[ \text{(11)} \]

In the case of nonpreemptive communication, the local delay bound again has to be further adjusted due to the possibility of blockage and control traffic, which finally results in the following set.

\[ t \in \bigcup_q \left\{ m \cdot TP_q + TD_q' : m = 0, 1, 2, \ldots \right\} \]  
\[ \text{(12)} \]

where

\[ t \in [1; BP_1] . \]  
\[ \text{(13)} \]

The upper and lower boundaries of the interval for \( t \) stay unmodified. Only when both the utilization constraint and the workload constraint are fulfilled can a feasible traffic allocation be guaranteed.

### 7.2 Traffic analysis algorithm

When using the feasibility analysis on a set of real-time flows over a network, it will return a simple ‘yes/no’ answer, providing the result to the question if all flows could be allocated over the network as if the network was a single resource. This is due to the original application of this analysis to uniprocessor task scheduling. In reality, a network is a set of overlapping resources, depending upon the physical and logical network architecture and the medium access control method used. In the presented context, a resource is constituted by a sender, a receiver and the unidirectional light path connecting them. If any of those components is busy, the whole resource is seen as reserved. This leads to the following definitions.

**Definition 1:**

A **resource** consists of a sender, a receiver and the unidirectional light path connecting them.
**Definition 2:**

In this analysis, a resource is defined to be \textbf{busy} if either the sender at the source node is transmitting to any other node connected to it or the receiver at the destination is receiving from any other node connected to it, or both are participating in the communication (sending and receiving respectively) over the unidirectional link between them, i.e., they actually use the medium for which the current traffic flow competes.

The reason for this definition of the 'busy' principle is the fact that in the suggested network architecture, the sender at each source node only can send to one destination at a time, and the receiver at each destination node only can receive from one source at a time (not taking into account the communication with the protocol processor which is allocated on a separate control channel). This automatically leads to the consequence that the resource, and therefore the unidirectional link between those two nodes, cannot be used for other communication as soon as one of them is busy.

The usage of the original feasibility analysis contains a rather large amount of pessimism as it completely dispenses with the possibility of simultaneous transmissions over resources independent of each other. The approach presented here concentrates upon the construction of virtual overlapping subnets in order to analyze them individually according to the already described feasibility test.

**Definition 3:**

One subnet consists of one main traffic flow and all other traffic flows it shares at least one part of the resource with, i.e., the flows which have the same source or destination node (or both) as the main studied flow.

**Definition 4:**

Two subnets are \textbf{overlapping} if they share at least the sender at the source node or the receiver at the destination node. As long as two subnets are not overlapping, simultaneous transmissions in them can occur at any time.

A subnet, which is a set of real-time channels, has to be found for each individual traffic flow over the network, and both the utilization and the workload tests have to be applied on all these subnets. Each subnet contains solely those traffic flows that are directly competing with the main studied flow, not taking into account flows that in their turn are competing with those. The details are described in the following paragraph.
In order to analyze the interdependencies of the traffic flows contained in one subnet, it has to be investigated for each single flow which parts of the resource it shares with any other flow. Flow $F_i$, which is the studied RT channel, is assumed to be characterized by its source $S_i$, its destination $D_i$, its period $TP_i$, its deadline $TD_i$, and its capacity $C_i$. The link between $S_i$ and $D_i$ is denoted $L_i$. The resource including $S_i$, $D_i$ and $L_i$ is denoted $R_i$. $F_i$ might have to share either its source or its destination node (or in fact both) and therefore the first step has to be to create a set $M_i$, i.e., a subnet, with all traffic flows $F_j$ that either have $S_i$ as its source node $S_j$ or $D_i$ as its destination node $D_j$. These are the following:

$$M_i = \bigcup_{j=1}^{Q} \left\{ F_j \mid S_j = S_i \text{ or } D_j = D_i \right\}$$  \hspace{1cm} (14)

This set also includes the studied flow $F_i$ itself and those flows parallel to it, i.e., which share all parts of the resource with the currently studied flow (i.e., it is an inclusive OR not exclusive OR in the set definition). However, apart from $F_i$, no other flows competing with any flow $F_j$ are included in the set.

### 7.3 Improved, less pessimistic feasibility analysis

In order to be able to decide if all tasks or flows of the ones competing for the same resource can be allocated in a way so that no deadline will be missed, a feasibility analysis has to be conducted for each set of competing flows, i.e., subnet. The number of subnets is equal to the number of traffic flows, i.e., $Q$. The traffic interdependency analysis introduced in the previous chapter opens up the possibility to now calculate the utilization $U_i$ of $R_i$ by all flows $F_j$ in $M_i$ as

$$U_i = \sum_{j=1}^{Q} e_j \text{ where } e_k = \begin{cases} \frac{C_k}{TP_k} & \text{if } F_k \in M_i \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (15)

It is still a necessary condition for this utilization to be $\leq 1$. In order to satisfy the second constraint connected to the workload of the link, the workload function is applied as before, but upon the smaller subnet $M_i$. The result answers the question whether hard real-time traffic can be guaranteed for the studied flow $F_i$. 


Both the utilization and the workload check are carried out for each set of traffic flows, i.e., each single traffic flow will be studied as the main flow of a set. The feasibility checks will generate the necessary information to decide if the studied flow can be guaranteed to fulfil its timing constraint. A negative answer does in itself not mean that the flow automatically will miss its deadline, but simply that no guarantee for it can be given. \( X \) denotes the set containing all traffic flows for which hard real-time can be guaranteed, while \( Y \) is the size of this set, i.e., the actual number of flows, for which timely treatment can be guaranteed:

\[
X = \bigcup_{j=1}^{Q} \{ M_i : U_i \leq 1 \text{ and } (\forall t) (h_i(t) \leq t) \}
\]

\[
Y = |X|
\]

When analyzing a hard real-time system, \( Y \) must be equal to the number of traffic flows in the system, i.e., \( Q \), for the system to be feasible. However, for the analysis of soft real-time systems, \( Y \) can be used as a parameter that indicates the degree to which real-time traffic can be guaranteed.

There can still be pessimism contained in this analysis. Flows might be included in several subnets, but due to the possibility of intricate traffic interdependencies, the complexity of an analysis studying dependencies more than one step from the main studied traffic flow might grow immensely fast, and is, for now, outside the scope of this work. However, the presented approach can increase the capacity which can be guaranteed for hard real-time traffic from 1 to \( Y \), with \( Y \geq 1 \), due to the reduction of the number of flows participating in the feasibility analysis to those flows that actually compete in a worst-case situation.

How can the possibility be excluded that this analysis results are too optimistic? Being too optimistic would infer that there are cases were this analysis method gives a positive answer, while in reality the traffic flow studied will miss its deadline. That in its turn could only happen if the admission control had not taken into account all flows demanding capacity of the resource competed for. As the subnet is defined as
the set of all flows that share this resource, no relevant competitors are left out, and therefore the admission control and the feasibility analysis have all necessary information to produce a reliable, nonoptimistic prediction as they calculate on the worst-case situation.

7.4 Simulation analysis

In order to demonstrate the improvement made possible by the usage of the improved feasibility analysis, a simulation program was implemented in Java. The assumptions are a $16 \times 16$ AWG, which means that there will be 15 end nodes and the protocol processor. The traffic pattern was assumed to be the following. The period as well as the deadline of all real-time traffic flows is 100 time slots and their maximum packet length corresponds to the length of one time slot. The source of each traffic flow is randomized with an even distribution, while the choice of destination is limited to include nodes contained in subgroups of a certain (variable) size. The maximum number of channels requested in the system is set to 2000. Each data point in the evaluation curves is the result of 100 iterations in order to increase the statistical reliability of the result.

![Figure 29. Throughput by guaranteed traffic flows.](image-url)
In the first figure, Figure 29, the number of requested real-time channels is increased (in steps of one) from 1 to 2000, after which the network seems to be saturated by the traffic load for all curves, i.e., for all subgroup sizes. The parameter plotted in the figure is the throughput in packets per time slot experienced by the guaranteed real-time channels depending on the number of requested real-time channels. (The calculation of the actual number of accepted real-time channels is basic as each channel $q$ has a bandwidth utilization of 1% ($C_q=1$, $TP_q=100$)). Different curves are plotted reflecting different sizes of subgroups. The cases illustrated in the figures are when each source can send to one possible other destination, or 4, 7 or 14 other destinations, which is the maximum possible number. (All possible sizes of subgroups were simulated, but some are excluded from the figure for readability reasons.) Which destinations are included in each subgroup is randomized with an even distribution.

Looking at Figure 29, the maximum throughput by accepted real-time channels is reached at the network saturation point of about 800 RTC requests. Most prominent, however, is the curve for the case of a subgroup size of one, where the maximum

![Figure 30. Calculated versus simulated throughput.](image)
throughput is 9.53, instead of the remaining values of around 7. The reason for this deviating behaviour is the relatively low probability of overlapping subnets when each traffic flow from a particular source has to have the same destination. The interference between all traffic flows is, over a statistically relevant time seen, smallest in this case when all traffic flows are randomized with an even distribution of source-destination pairs.

The theoretical maximum throughput, denoted \( TMT \), reachable in a network with this configuration, and under the assumption of equally many traffic flows between any source-destination pair, can be calculated by

\[
TMT = \frac{N \cdot N_{\text{dest}}}{2 \cdot N_{\text{dest}} - 1}
\]  

(18)

where \( N \) is the total number of nodes in the network, and \( N_{\text{dest}} \) is the number of destinations per subgroup. This theoretical average throughput should however only be used for an approximate comparison since it is based on the assumption of a nonrandom even distribution. The theoretically calculated values are compared to the simulation results in Figures 30 and 31.

Figure 31. Difference between calculated and simulated throughput.
As can be seen in these figures, the difference between the simulated and calculated throughput decreases continuously as the number of possible destinations per source increases. For a small number of possible destinations, the lower simulated throughput is due to the relatively high probability of nonoverlapping subnets which reduces the actual throughput in comparison to the one that can be calculated. The turning point where the simulated throughput changes from a sinking to an increasing trend lies around a subgroup size of four.

In order to further investigate this behaviour, the throughput of the guaranteed real-time channels depending on the number of destinations per source, i.e., subgroup size, is studied in Figure 32. The curves illustrate the results for different traffic loads in the system. While 500 RTCs can be accommodated easily by the network, the higher traffic loads can be seen experiencing the same behaviour as described earlier regarding Figure 30. However, this curve shows also that the influence by the low probability of overlapping subnets (which was illustrated by the ‘Subgroup size = 1’ curve in Figure 29 earlier) is observable up to a subgroup size of four different destinations per source. In that point the three upper curves have their minimum after which they start increasing.

![Figure 32. Throughput depending on subgroup size.](image-url)
In summation, the simulations show that the number of real-time channels that can be guaranteed to meet their deadlines is considerably higher when using the suggested traffic analysis in combination with a real-time analysis compared to simply using the real-time analysis. The guaranteed throughput was increased from being one packet per slot to being around seven, i.e., a utilization of about 700%. For some traffic patterns, even higher guaranteed throughputs can be reached as shown by the simulations.
8 Conclusions

High-performance embedded systems with heterogeneous traffic, high throughput needs and strict delay bound demands are in need of new communication solutions providing the necessary performance. Optical technologies are an interesting contender due to their high bandwidth and low loss qualities. In this thesis, special interest was directed towards the AWG which due to its potential for high concurrency offers the possibility of short communication delays.

Network architectures, based upon the AWG and different queuing architectures, were developed for two application areas, namely System Area Networks and distributed routers. Corresponding MAC protocols were designed in order to make the best use of the AWG’s advantages, and reservation-based MAC protocols were chosen due to their well-known good performance for QoS requiring traffic. As a high grade of concurrency is possible in the suggested AWG-based network, the delay caused by the reservation process could be limited to one single time slot. Evaluation work done by the help of simulations showed the suitability of the presented solutions for the targeted applications, keeping, e.g., the deadline miss ratio for hard real-time traffic at zero in all studied traffic scenarios.

Targeting hard real-time systems, the need of guaranteed throughput and a limited delay bound is important to meet. In order to do that, a real-time analysis was included already when evaluating the developed solution by simulation. The admission control process used a busy-period analysis method originating from the area of real-time systems, but adapted for a communication context, in its decision process. Being aware of the pessimism incorporated in the analysis, a traffic dependency analysis was added to improve the performance of the analysis when used for the kind of multichannel networks in focus in this thesis. Simulating the analysis for different traffic patterns indicated that the throughput guarantee can be improved substantially.

Summarizing the accomplished results, the AWG-based network solution and the corresponding MAC protocols were found to satisfy the demands put up by the target applications in high-performance embedded systems. More research into different aspects of communication networks in embedded systems is needed, before comprehensive recommendations can be made however. Limiting the research on which this thesis is based to the real-time communication requirements of embedded systems, resulted in the conclusion that these kinds of optical network are very suitable for the studied high-performance embedded systems with heterogeneous traffic demands and strict timing constraints.
References

[Acampora et al. 1989]

[Banerjee 1998]

[Baruah et al. 1990a]

[Baruah et al. 1990b]

[Bellón et al. 2004]

[Bengi 1999]
K. Bengi, “Performance of single-hop WDM LANs supporting real-time services”, *Photonic Network Communications*, vol. 1, iss. 4, 1999, pp. 287-301.

[Bengi 2001]
[Bengi et al. 2001]

[Bergenhem et al. 2002]

[Bernasconi et al. 2006]

[Borella et al. 1999]

[Böhm et al. 2004]

[Böhm et al. 2005a]

[Böhm et al. 2005b]
[Cheyns et al. 2003]

[Chia et al. 1999]

[Chia et al. 2000]

[Chia et al. 2001]

[Choi 2003]

[Diao et al. 2001]

[Dolzer et al. 2002]
[Dong et al. 1998]

[Dragone 1999a]

[Dragone et al. 1999b]

[Fan et al. 2003]

[Fan et al. 2004]

[Gerstel et al. 1996]

[Gripp et al. 2003]

[Hoang et al. 2003]
[Huang et al. 2004]

[Huang et al. 2005]

[Huang et al. 2007a]

[Huang et al. 2007b]

[Huang et al. 2008]

[Hunter et al. 1999]

[Jeon et al. 1990]

[Jia et al. 1995]
[Jonsson et al. 1997]

[Jonsson 1999]

[Jonsson 2003]

[Kam et al. 1998a]

[Kam et al. 1998b]

[Kim et al. 1995]

[Kim et al. 2002]

[Kunert et al. 2005a]
[Kunert et al. 2005b]

[Kunert et al. 2008a]

[Kunert et al. 2008b]

[Kuri et al. 2001]

[Kyriakis-Bitzaros et al. 2006]

[Li et al. 1998]

[Liu et al. 1973]

[Lu et al. 1992]
[Ma et al. 1998]

[Ma et al. 2000a]

[Ma et al. 2000b]

[Ma et al. 2000c]

[Ma et al. 2001]

[Ma et al. 2003]

[Maier et al. 2000a]

[Maier et al. 2000b]
[Maier et al. 2001]

[Maier et al. 2002]

[Maier 2003a]

[Maier et al. 2003b]

[Maier 2004a]

[Maier et al. 2004b]

[Mukherjee 1992a]

[Mukherjee 1992b]

[Neilson et al. 2005]
[Ngo et al. 2004]

[Ngo et al. 2006]

[Nizam et al. 1998]

[Petridou et al. 2008]

[Selvakennedy et al. 1996]

[Semaan et al. 1993]

[Smit et al. 1996]
[Spencer et al. 2000]

[Spuri 1996]

[Stankovic et al. 1998]

[Takahashi et al. 1995]

[Tse 2005]

[Tyan et al. 1996]

[Tyan et al. 2001]

[Verbeek et al. 1995]
[Wang et al. 2003]

[Wang et al. 2006]

[Woesner et al. 2003]

[Yan et al. 1996]

[Yang et al. 2004]