Technische Universität München
Lehrstuhl für Medientechnik

A Resource-Efficient IP-based Network Architecture for In-Vehicle Communication

Dipl.-Ing. Univ. Mehrnoush Rahmani

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. sc. techn. Andreas Herkersdorf
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. Eckehard Steinbach
	2.	Prof. Dr. rer. nat. Ernst W. Biersack Grande Ecole Telecom Paris/ Frankreich

Die Dissertation wurde am 16.03.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 19.06.2009 angenommen.

Acknowledgment

This dissertation was written during my time as PhD student and research member in the network architecture group at BMW Research and Technology and at the Institute for Media Technology at Technische Universität München (TUM). I am grateful to both institutes to give me the opportunity for this work. This thesis is the result of three years of work whereby I have been accompanied and supported by many people. It is a pleasure to have the opportunity to express my gratitude to all of them.

First, I would like to thank my PhD adviser Prof. Dr.-Ing. Eckehard Steinbach for accepting to supervise me as an external PhD student and for his continuous support. I am grateful to him for his many suggestions and for being available whenever I needed his advise. His constructive way of work taught me how to carry on in all difficult situations. I owe him lots of gratitude for challenging me and showing me how to progress.

Special thanks goes also to my BMW supervisor, Richard Bogenberger, who kept an eye on the progress of my work and integrated me perfectly in his projects. I thank him for providing me with all facilities I needed for this work. I also would like to express my profound appreciation to Mr. Karl-Ernst Steinberg, head of the department for offering me this position and for his constant support also after the PhD period.

I would like to express my special gratitude to Prof. Dr. Ernst Biersack from Eurecom for accepting to be my second PhD adviser. I learnt a lot from him during our cooperative projects over the period of this dissertation. It is a pleasure for me to have him as my second thesis examiner.

I also would like to acknowledge the cooperative and friendly atmosphere in BMW Research and Technology which is certainly due to its staff. All of my colleagues supported me during this work. My team colleagues, Wolfgang Hintermaier, Joachim Hillebrand, Dr. Rainer Steffen, Dr. Daniel Herrscher and Andreas Winckler from the ZT-4 department, Dr. Klaus Gresser, Andreas Laika and Dr. Marc Walessa from the ZT-3 department contributed to the success of this thesis. I am grateful to all of them. I also would like to thank Isaac Trefz, Holger Endt, Champ, Ben Krebs and Martin Pfannenstein for proofreading this thesis. I appreciate the staff of the Institute at TUM for providing a pleasant environment to work during the periods I spent at the university.

All students I supervised did an excellent work and contributed a lot to this dissertation. It would not be possible to finish this thesis within three years without their contributions. I thank them all and hope to have the opportunity to support them too in the future.

Last but most notably, a great thank goes to my dear parents who always encourage me with the best advises for life. This work would never be done without their infinite support. Also, my younger brother and sister, Alborz and Delaram enrich my life with much love. In the end, I dedicate my most special thank to my future husband, Michael who is my best friend, mentor, ideal and stand in life.

Munich, January 2009 *Mehrnoush Rahmani*

Abstract

This dissertation investigates various aspects of network design and planning for future in-vehicle data communication. The major issues addressed are network architecture and topology design, network dimensioning, and resource-efficient streaming by means of traffic shaping and video compression in driver assistance camera systems. Concerning the network architecture and topology design, standardized communication protocols from the IT domain are analyzed with respect to the in-vehicle communication requirements. A heterogeneous and all-IP-based network architecture is introduced in two different representative network topologies as candidate parts of the future overall in-vehicle network. Motivated by the fact that the car is a closed network system where all applications and their transmission scenarios are known a priori, a static network dimensioning method is derived analytically and verified by a self-designed simulation model. Quality of Service and resource usage are analyzed in the proposed network topologies. Traffic shaping is used to reduce the required network resources and consequently the cost. A novel traffic shaping algorithm is presented that outperforms other traffic shapers in terms of resource usage when applied to variable bit rate video sources under certain topology constraints. Video compression algorithms are investigated in driver assistance camera systems to be configured such that negative effects on the system performance are avoided while the overall resource usage is reduced. Finally, an experimental prototype is introduced that demonstrates the applicability of the proposed IP-based network in a real car.

Zusammenfassung

In der vorliegenden Arbeit werden Konzepte für die zukünftige Datenkommunikation im Fahrzeug entwickelt. Dabei liegen die Schwerpunkte auf dem Entwurf der Netzarchitektur und -topologie, der Netzdimensionierung, der ressourceneffizienten Übertragung mittels Verkehrsgestaltung (engl. Traffic-Shaping) sowie der Videokompression in Fahrerassistenz-Kamerasystemen. Zunächst werden für den Entwurf der Netzarchitektur und -topologie standardisierte Kommunikationsprotokolle aus dem IT-Bereich bezüglich der Erfüllung der Kommunikationsanforderungen im Fahrzeug untersucht. Anschließend wird der Entwurf einer heterogenen und durchgängig auf IP-basierten Netzarchitektur für zwei Arten von Netztopologien vorgestellt. Diese sind repräsentativ für zukünftige mögliche Fahrzeugbordnetze. Ausgehend von einem geschlossenen Kommunikationsnetz im Fahrzeug mit a priori bekannten Applikationen und Übertragungsszenarien wird eine statische Netzdimensionierungsmethode analytisch hergeleitet und anhand eines eigens entwickelten Simulationsmodells validiert. Um die benötigten Netzressourcen und somit die Kosten zu reduzieren, wird die Methode des Traffic-Shaping eingesetzt. Hinsichtlich der erforderlichen Ressourcenbelegung ist der vorgestellte Traffic-Shaping-Algorithmus bei Anwendung in Videoquellen mit variabler Datenrate unter bestimmten Topologiebedingungen allen bekannten Traffic-Shapern überlegen. Weiterhin werden Videokompressionsalgorithmen für den Einsatz in Fahrerassistenz-Kamerasystemen untersucht und so konfiguriert, dass negative Einflüsse auf die Systemperformanz vermieden werden und gleichzeitig die Ressourcenbelegung reduziert wird. Abschließend wird ein prototypischer Aufbau vorgestellt und die Anwendbarkeit des neu entwickelten IP-basierten Netzes in einem realen Fahrzeug verifiziert.

Contents

1 Introduction

This chapter provides an overview of the issues that motivated this work and introduces the research areas of this dissertation: *Network architecture design*, *topology planning*, *network dimensioning*, *traffic shaping* and *video compression in driver assistance camera systems* together with all contributions. Finally, the structure of the thesis is presented to guide the reader throughout the work.

1.1 Motivation

Today's premium cars contain up to 70 electronic control units (ECUs) interconnected by different automotive specific network technologies such as FlexRay, MOST (Media Oriented Systems Transport), CAN (Controller Area Network), and LIN (Local Interconnect Network) providing limited transmission capacities. Point-to-point connections realized by analogue CVBS (Color Video Blanking Signal) and digital LVDS (Low Voltage Differential Signaling) cables are used to transmit real-time video streams from driver assistance camera systems. Among the in-vehicle ECUs, there are more than ten distributed audio and video ECUs such as visual sensors (e.g., Radar, FIR for night vision), driver assistance cameras (e.g., rear-, side-, top-view cameras), DVD player, DVB-T and audio sources such as FM- and HD-radio systems. Audio and video streams are sent to several receivers such as the CID (Central Information Display), Head Up Display as well as the audio amplifier, Dolby Digital 7.1 DSP and several loud speakers. Additional displays and headsets are provided for the rear seat passengers. Except for FlexRay, all other mentioned network technologies are currently used to interconnect the audio and video ECUs in upper class premium vehicles. The application of different network technologies and point-to-point links leads to an inflexible network architecture and a complex cable harness in the car, which is expensive and requires high validation and management effort. Due to the growing demand for new applications, especially in the driver assistance and multimedia fields, the in-vehicle network will become even more complex and costly in the near future. Thus, traditional automotive network technologies are no longer suitable.

The ISO/OSI-layer 3 protocol IP (Internet Protocol) is the most dominant network protocol in the world and is perfectly appropriate for achieving independence from individual network technologies. It means that different transmission technologies can be used below IP in the OSI model. It is supported by all modern operating and network systems. Transport protocols such as the well-known TCP (Transport Control Protocol) and UDP (User Datagram Protocol) extend the IP functionality with transport oriented characteristics for an appropriate data delivery. In other words, respective to the application requirements, the IP network can be enhanced with different network technologies in physical and MAC layers and with other transport protocols in the transport layer. Additionally, the IP protocol stack is under continuous development. For instance, hardware implementations of the TCP/IP stack have been introduced to reduce CPU

load. An example here is the TOE (TCP Offload Engine) that is used in NICs (Network Interface Controllers) to offload the entire TCP/IP stack from the CPU to the network controller. Also μIP stacks have been introduced, e.g., as in [15] to be used in sensor networks. Moreover, an automotive qualified TCP/IP stack has recently been presented that fulfills the strict memory and power requirements of automotive ECUs [36].
Ethernet [51] has become the dominant network technology in computer networks. It represents the layers 1 and 2 in the OSI model. With the introduction of full-duplex switched Ethernet, the applicability of Ethernet has improved even more. This is because in full-duplex switched Ethernet networks, collisions do not occur, different transmission rates can be applied for individual devices and sending and receiving data is possible simultaneously and collision free [122]. Due to its wide availability in computer networks, Ethernet has also become the most cost-efficient technology among all other broadband network technologies such as FireWire [52]. Accordingly, the application of an IP over Ethernet (IP/Ethernet) network in the car represents an interesting possibility for future in-vehicle communication. However, Ethernet fails in providing quality of service (QoS) and real-time guarantees. Several solutions have been presented to use IP and Ethernet with QoS and real-time capability as will be discussed in Section 2.2. They all either modify the standard Ethernet or are only adapted for a small group of traffic types. The in-vehicle traffic consists of several different traffic types, from real-time control data to real-time audio and video streams and best effort data. Besides the various traffic types with different QoS requirements, production cost is another significant factor in the automotive sector where a large number of samples is produced for a model range of cars. Therefore, the application of standard components and protocols is essential in the car to keep costs low. Also, other network parameters such as topology, link capacity and resource usage play an important role in the cost value and should be carefully selected in the network planning phase. In the planning of an IP-based network, integration of new services with an adequate and stable degree of quality has to be considered. According to the general definition [108], network planning seeks an optimal trade-off between QoS and emerged cost. Consequently, a network should be planned that provides the required QoS with convenient cost. The terms QoS and cost can be defined as follows.

- **QoS:** Guaranteeing QoS for all services in the network is the goal of network planners. IETF has defined several metrics in the service requirements in order to control and evaluate the QoS [35]. They can be summarized as delay time, jitter (delay variations), packet loss and throughput.

- **Cost:** The implementation and startup of communication networks is a complicated procedure where the cost is difficult to estimate. The network planning cost can be divided into three groups as follows.

  - Capital Expenses (CAPEX) are related to infrastructure and resource requirements. Network resources should not be wasted by overdimensioning.
  - Implementation Expenses (IMPLEX) contain the network construction, installation and license costs.
  - Operational Expenses (OPEX) contain mainly the maintenance, management and marketing costs.

In order to be independent of component manufacturers and their price offers, only CAPEX have been taken into account in the analysis of this work. Thus, the problem of network planning can be reduced to the definition of optimal values for the topology, link capacity and resource allocation. They should be computed considering cost and design restrictions, i.e., QoS requirements of different applications, various traffic flows, nodes and link failures. Because of its high complexity, the research community has subdivided the network planning into sub-issues. In [82] several design sub-issues are introduced such as capacity assignment that describes the link capacities required for high-quality media streaming with the lowest cost and the topology assignment that defines the node positions and transmission routes in the network. Figure 1.1 shows the process of network planning in a simplified manner. First, the system requirements have to be defined.

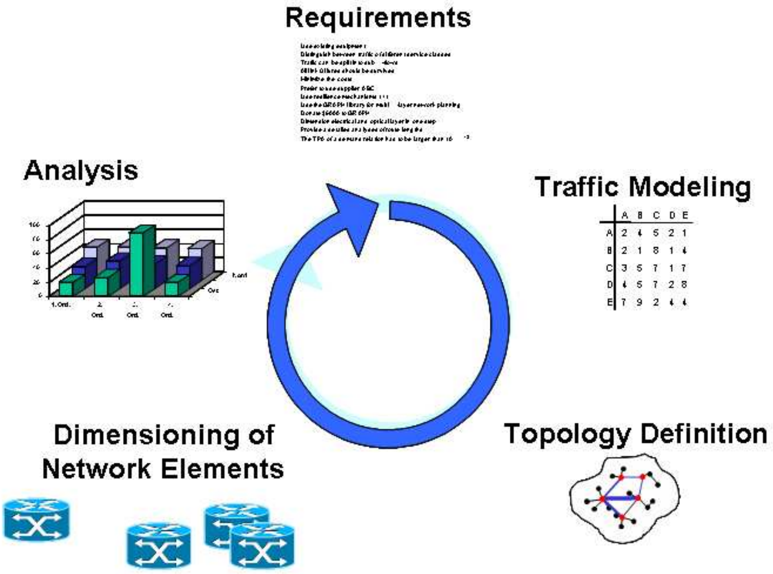


Figure 1.1: The process of network planning.

Thus, the considered transmission scenarios and data flows are determined for further analysis. Data flows have to be mathematically modeled to be interpretable in the following processing steps. Appropriate network topologies are then identified. Based on all known parameters such as transmission scenario, data flows and network topology, network dimensioning is now carried out. In the last processing step, resource usage and the achieved QoS are evaluated and if necessary, by relaxing some requirements, the whole network planning process can be repeated.

For a resource-efficient network dimensioning, it is required that the traffic entering the network conforms to certain characteristics. For example, the data rate should not exceed a pre-defined upper bound as not to overload the network, nor should it fall below a certain lower bound not to exceed the end-to-end delay requirement. In particular, variable bit rate video sources with large peak rates require a bit rate controller to fulfill the QoS requirements with a reasonable amount of resources. Such a rate controlling mechanism is called traffic shaping. A traffic shaper regulates its outgoing stream by delaying some packets until they can be sent without violating the desired traffic characteristics [83]. It seeks a trade-off between the required network resources and the achieved QoS. Various traffic shaping algorithms change traffic streams differently and therefore should be carefully selected and configured for each application.

Driver assistance camera streams are currently transmitted as raw data flows over separate links throughout the car. By using high video resolutions such as VGA resolution (640 × 480 pixels), a large amount of data is sent over each transmission link. In a convergent IP/Ethernet network, it is required to transmit several video streams over one link. This underlines the necessity of video compression in driver assistance camera systems to reduce the amount of transmitted data and realize all required transmission scenarios in future IP/Ethernet-based in-vehicle communication networks. However, it is a crucial task to find appropriate video compression algorithms and their configuration for different driver assistance services.

1.2 Contributions and Outline of this Dissertation

The focus of this thesis lies on the different areas of network planning as shown in Figure 1.1 such as network architecture and topology definition, traffic classification and modeling, resource planning and network dimensioning. Traffic shaping and video compression are applied to end-systems as mechanisms to reduce the required network resources. The goal of this work is to design a resource-efficient, convergent and QoS-aware network architecture for in-vehicle communication by means of standard components and protocols thus reducing the overall cost. According to the car manufacturers' requirement, the planned in-vehicle network should cover all traffic flows in the car except for the safety-critical control data from FlexRay applications. However, the network must be planned such that the integration of FlexRay data can easily be done when there is such a requirement in the future.

The following four paragraphs correspond to Chapters 2-5 and summarize the main contributions of each chapter. Chapter 6 concludes the thesis and gives an outlook on future research directions.

Related Works, Requirement Analysis and Traffic Modeling

In Chapter 2, an overview is given of the existing automotive network systems, their current application fields and limits. Except for the FlexRay bus, all other automotive networks are used in the interconnection of audio and video-based applications in the car. Due to the growing demand for more audio and video-based applications in the car, they are selected as the focus of this thesis. The main drawback of the current automotive networks is their limited transmission capacity, which is not enough to support future audio and especially, video-based applications in the car. As an alternative to the automotive networks, existing IP/Ethernet-based networks with different QoS approaches are discussed for their adaptability in the car. However, due to their application limitations and the lack of standard conformance, they are not adapted for in-vehicle communication. Afterwards, the basics of image and video compression are introduced and different applicable compression algorithms are reviewed to be used in driver assistance cameras. In the last section of Chapter 2, a requirement analysis is carried out for the in-vehicle traffic. Based on the defined requirements, the in-vehicle traffic is divided into the four classes real-time control data, real-time audio and video streams, multimedia data and best effort data. It is modeled by constant bit rate and variable bit rate data flows as follows.

- **Constant bit rate:** Periodic control data and uncompressed audio streams
- **Variable bit rate:** Compressed video streams

In addition, bulk FTP traffic representing the best effort data is inserted into the network. While the modeling of constant bit rate traffic is simple, an accurate modeling of the rate variability is challenging but essential for adequate resource planning. The fractional autoregressive integrated moving-average (F-ARIMA) model [108] with the Pareto distribution is used to model compressed video streams in the car.

Design of a Heterogeneous Convergent IP-based Network Architecture for In-Vehicle Communication

In Chapter 3, a heterogeneous network architecture consisting of a core wired IP/Ethernet network and a peripheral wireless IP/WLAN network is introduced to transmit all considered data flows throughout the in-vehicle network. A QoS-API is presented that works as a configuration API parallel to the BSD sockets and provides static priority tags for different applications. Thus, data packets are forwarded according to their priority tags and QoS is provided as long as the network is not overloaded. To guarantee the QoS at any time, also in network overload situations, a static network dimensioning method based on Network Calculus theory is presented. A simulation model is also developed to evaluate the analytical results in a more realistic way. Thus, required network resources (i.e., service rate and buffer size) can be computed for any application configuration in the car before data transmission is initiated to guarantee the QoS. For wireless multicast transmissions, an adaptive forward error correction (FEC)-based mechanism is presented that is required in addition to the static network dimensioning for providing the required quality at wireless receivers. The computed resources turn out to be very large for wired transmissions while wireless transmissions cannot be fully performed due to sudden channel saturations. Instantaneous large peak rates of compressed video streams are identified as the cause of these transmission issues. Also, small sized packets account for the wireless channel saturation. This issue is solved by the method of frame bursting. Both, analytical and simulation results are provided in Chapter 3.

Resource-Efficient Data Transmission by Means of Traffic Shaping

Solutions are presented in Chapter 4 to overcome the transmission issues identified in Chapter 3, namely the large resource usage in the wired network and the channel saturation in the wireless network. In order to reduce the instantaneous peak rates of video sources resulting in large bursts in the network, the mechanism of traffic shaping is proposed. Source shaping assigns desired characteristics to the data stream before entering the network. Thus, traffic bursts can be prevented and network dimensioning is simplified by knowing all source statistics. Reshaping compensates the jitter introduced by the previous network elements and thus, reconstructs the original data streams. Consequently, less resources are needed and the process of network dimensioning is simplified. Several well-known traffic shaping algorithms are analyzed in the video sources as source shapers and in the interconnected network elements (e.g., switches, access point) as reshapers. A novel traffic shaping algorithm called Simple Traffic Smoother (STS) is introduced that outperforms other traffic shapers in terms of resource usage when applied to variable bit rate video sources under certain topology constraints. Also, a new architecture design is presented for a traffic reshaper implementation, which operates on a per stream basis. All different traffic shaping algorithms are investigated and compared for their performance at providing QoS and reducing resource usage. The most appropriate network configurations are defined by means of analytical and simulation analysis. Finally, a prototype is implemented based on the theoretical results and evaluated through quality measurements. The results confirm that the proposed network architecture with the presented QoS mechanisms is well-adapted for in-vehicle communication. A general design guideline is introduced that can be applied to any other closed network system such as the car.

Introduction of Video Compression in Driver Assistance Camera Systems

In Chapter 5, different video compression algorithms are studied and verified to be applied to driver assistance camera systems. Quality metrics are defined to make the algorithms comparable. A testbed is implemented for QoS measurements with the focus on wired transmission. Analyses are performed for the direct display of decoded video streams, e.g., rear-view presentation on the HMI, that we call direct image-based driver assistance system and for the indirect representation of image information, e.g., the lane departure warning triggered by the front-view camera, that we call indirect image-based driver assistance system. In both cases, appropriate compression algorithms are selected through measurements. Finally, analysis results are presented for different system configurations and image processing algorithms applied in indirect image-based driver assistance systems.

Parts of this dissertation have been published in [1], [2], [4], [6], [3], [5], [9], [12], [11], [8] and [10].

2 State of the Art

The use of communication networks connecting electronic control units in the car dates back to the early 1990s. Since the 1970s, an exponential increase of the number of electronic systems has been observed that have gradually replaced those that are purely mechanical or hydraulic [93]. Different requirements of the different automotive applications have led to the development of various automotive specific networks. However, the existing automotive networks with limited transmission capacity cannot fulfill the growing demand for more applications, especially audio and video applications in the car. The Ethernet technology, on the other hand, offers enough transmission capacity for the integration of future applications in the car. In this chapter, an overview is given of automotive network systems that are currently in use for audio and video communication. These are compared with IP over Ethernet (IP/Ethernet) networks, especially those that are applied in other special communication environments. It is shown that Ethernet technology, when extended with an adequate QoS mechanism, is well-adapted to be applied for in-vehicle communication. Additionally, the in-vehicle traffic is classified and video compression is introduced as a key technology to reduce resource usage.

2.1 An Overview of the Existing Automotive Network Systems

Today's in-vehicle network system consists of several automotive network technologies such as FlexRay, CAN, MOST, LIN, and also point-to-point connections realized by analogue Color Video Blanking Signal (CVBS) and digital Low Voltage Differential Signaling (LVDS) cables. Video transmission systems such as driver assistance and multimedia systems represent the most complex network systems in today's high-class cars and are therefore the focus of this work. Figure 2.1 shows an example for the current in-vehicle video transmission system including a CAN segment, MOST, LIN, LVDS and CVBS connections. The above mentioned automotive network technologies are briefly explained in a chronological order in the following.

2.1.1 CAN: Controller Area Network

CAN is a multicast shared serial bus standard, developed in the 1980s by Robert Bosch GmbH, to connect ECUs originally for automotive purposes, i.e., as a vehicle bus in electromagnetically noisy environments [106], [144]. Nowadays, it is used for many embedded control applications in industrial environments.

The CAN data link layer protocol is standardized in ISO 11898-1 and describes mainly the data link layer, composed of the Logical Link Control (LLC) sublayer, the Media Access Control

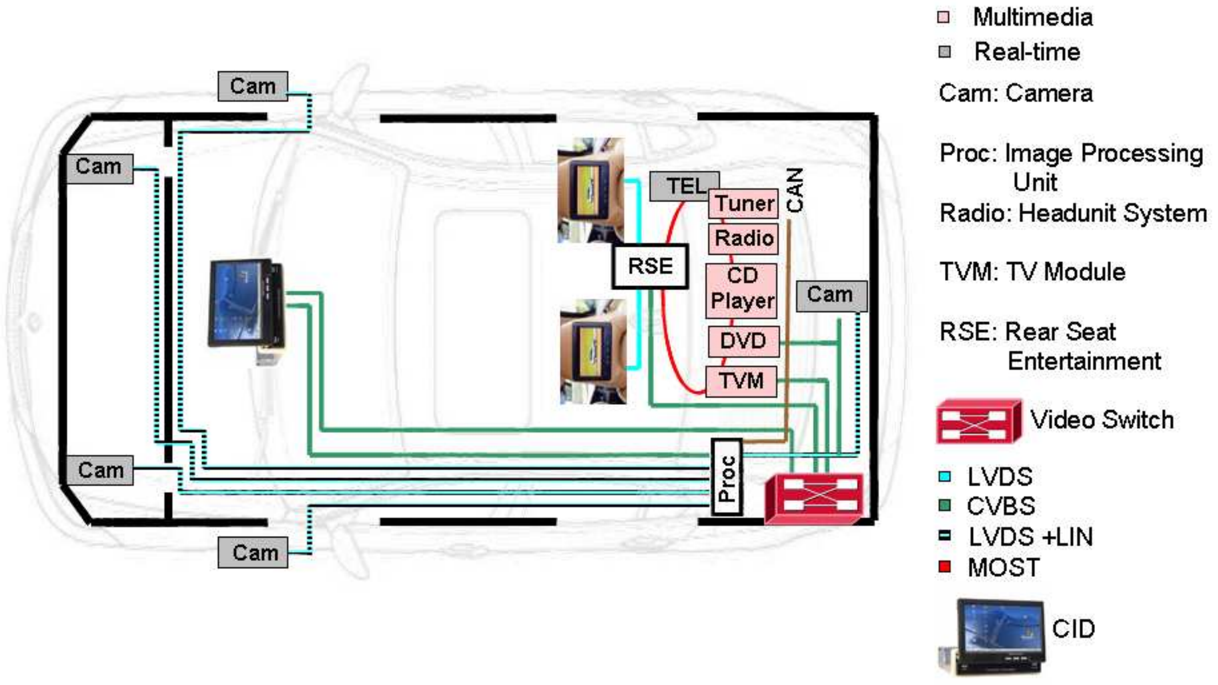


Figure 2.1: An example for the current in-vehicle video transmission system.

(MAC) sublayer and some aspects of the physical layer. All other protocol layers are left to the network designer's choice. The CAN bus can use different physical media such as electrical twisted-pair cables and fiber optics. However, the most common medium is the former one.

As commonly used with a differential bus, a Carrier Sense Multiple Access/Bitwise Arbitration (CSMA/BA) scheme is implemented in the CAN bus, i.e., if two or more devices start transmitting at the same time an arbitration scheme is accomplished based on the message identifier fields to decide which node will be granted the permission to continue transmission. During arbitration, each transmitting node monitors the bus state and compares the received bit with the transmitted bit in the message identifier fields. If a dominant bit is received when a recessive bit is transmitted the node stops transmitting and backs off, i.e., it has lost the arbitration. After the transmission of the identifier fields, all nodes bar one back off. The highest priority message that is the one with the dominant bit in the identifier field gets through unimpeded. This is how messages are assigned with different priorities in the CAN bus.

In CAN networks there is no addressing of subscribers or stations in the conventional sense. A transmitter sends a message to all CAN nodes (broadcasting). Each node decides on the basis of the message identifier whether it should process the message or not. The identifier defines the meaning of the data, i.e., the application type. Thus, all nodes in the network are able to filter messages and process only those that are targeted for them.

In addition to the CSMA/BA scheme that guarantees the collision avoidance, the CAN protocol provides several other important features. It guarantees the transmission delay time by using messages of limited lengths and fixed bit rates in each system. It also provides flexibility in the system configuration, since nodes can be added to and removed from the CAN network without changing the software and hardware of the other nodes. Since each message is first broadcasted and then filtered by the nodes, any group of nodes can simultaneously receive and react to a message. CAN provides the concept of multi-master, i.e., all nodes can transmit simultaneously when the physical medium is idle. The message with the highest priority gains the bus access. In order to ensure a safe data transfer, several error detection mechanisms such as Cyclic Redundancy Check (CRC), bit stuffing, bus monitoring and message frame check are applied. Corrupted messages are flagged by the node that detects them. They are aborted and retransmitted automatically. The recovery time from detecting an error until the transmission of the next message is at most 29 bits long, if there is no further error on the bus. Permanent errors can be identified and the defect nodes are autonomously switched off.

Each CAN message can transmit up to 8 bytes of payload data. Longer messages are accordingly segmented. A maximum of 127 nodes can be connected to a CAN bus. Bit rates up to 1 Mbit/s are possible at network lengths below 40 m. Decreasing the bit rate allows for longer network distances, e.g., 125 kbit/s at 500 m. Currently, CAN is used for the transmission of control messages in almost all automotive domains. Also, the sensor data for driver assistance services is transmitted by the CAN bus. Several types of the CAN bus that differ in their transmission rate and upper layer protocols are applied in the car, such as:

- S-CAN (safety-relevant CAN segment for automatic cruise control and anti-collision functions),
- PT-CAN (power train applications),
- FA-CAN (chassis and power train applications),
- K-CAN (CAN segment for driver assistance and infotainment applications),
- Body-CAN (comfort functions),
- Safety-CAN (safety-critical driver assistance).

2.1.2 LIN: Local Interconnect Network

The LIN specification was introduced in 1999 by the LIN Consortium. LIN was developed as a concept for low cost and low complexity automotive networks. It is a serial bus similar to the CAN and is applied where the transmission capacity and versatility of CAN are not required [143], [65]. LIN supports the single master concept. The media access is realized through a polling mechanism. The low cost characteristic of LIN is achieved by

- low cost silicon implementation based on common Universal Asynchronous Receiver Transmitter (UART) in hardware (an equivalent exists also in software),
- self synchronization without a quartz or ceramic resonator in the slave nodes, and
- low cost single-wire implementation.

LIN supports deterministic signal transmission with in advance computation of signal propagation times. The messages have a maximum latency and transmission time. The master determines the chronology and the frequency of messages with a scheduling table, which is switchable during runtime. LIN supports data rates up to 20 kbit/s with a maximum packet payload size of 8 bytes. Up to 16 devices can be connected by a LIN bus.
According to the mentioned characteristics, the LIN bus is adapted for simple networks with one master that controls several slaves. In current premium vehicles, the LIN bus is used for the communication of intelligent sensors and actors.

2.1.3 MOST: Media Oriented Systems Transport

MOST is a standard developed for interconnecting multimedia components. It was first introduced in the BMW 7 Series in 2001. In the following, the optical MOST ring that is currently applied for audio streaming in the car is explained. Other realizations of MOST can be found in [18].
Based on the optical fiber bearer, MOST provides a network system based on the multi-master concept at bit rates much higher than other present automotive network technologies. The MOST specification [141] defines all 7 ISO/OSI layers for data communication. The Application Programming Interface (API) of MOST devices is object oriented so that applications can concentrate on their functions. The API is able to control all features that MOST devices provide, such as those from audio equipments, GPS systems, telephones or telematic systems. Since all MOST devices use the same API, compatibility is assured. Accordingly, the MOST specification encompasses both the hardware and the software required to implement a multimedia network. It is produced by only one manufacturer.
MOST is a synchronous network with the system transmission frequency chosen to be 44.1 kHz, which is adapted to Audio CD and accordingly not appropriate for other applications such as DVDs (48 kHz). A timing master supplies the clock. All other devices synchronize their operations to this clock. Buffering and sample rate conversion are therefore not needed. In practice, there are small buffers at the receivers for decoding purposes.
Three channels are applied in the MOST specification:

- Control channel with a transmission rate of 700 kbit/s,
- Synchronous channel with data rates up to 24 Mbit/s,
- Asynchronous channel with variable data rates up to 14.4 Mbit/s.

The control channel is used to send control messages and setup connections. Once a connection is established, packets with a maximum of 60 bytes payload can flow over the synchronous or asynchronous channels and no further information processing is required.
MOST often employs a ring topology but star configurations and double ring are also possible. It can theoretically include up to 64 devices.
MOST is only used in the Multimedia domain and connects multimedia devices in vehicles. It should be mentioned that the current MOST in the cars does not have the capacity to transmit video data. Therefore, high speed point-to-point links, i.e., digital LVDS and traditional analogue CVBS cables are currently applied for video transmission in the car.

The MOST Cooperation has recently introduced the next MOST generation with 150 Mbit/s transmission rate [90]. Thus, all three mentioned transmission channels are extended and have a larger transmission capacity. In addition, two new channels, an isochronous channel and an Ethernet channel are introduced. Data streams that differ from the MOST system frequency of 44.1 kHz such as MPEG video streams are transmitted via the isochronous channel. By the Ethernet channel, connections to standard TCP/IP stacks are possible without any modification in the applications. Further on, no segmentation of large packets is needed in the new MOST standard as opposed to the old MOST generation, also called MOST25. By the so called preemptive acknowledgments, the receiver informs the sender about reception problems. Thus, no messages are sent that cannot be received and the resource usage is improved. However, this new MOST standard is not yet available in the market and consequently not yet verified for automotive use-cases.

2.1.4 FlexRay

FlexRay [139], [129] is a deterministic and fault-tolerant network technology, which supports data rates up to 10 Mbit/s[1] for advanced automotive control applications. Two transmission channels can be used in the FlexRay network. The second channel is used to double the transmission rate and for a redundant transmission in safety-critical applications. Both, bus and star topologies can be realized with the FlexRay technology. However, the bus topology is the widespread one. FlexRay uses a time triggered network access with a global time synchronization to guarantee message latency times. Each of its communication cycles contains a static segment, a dynamic segment, a symbol window and a network idle time. Within the static segment, static Time Division Multiple Access (TDMA) with constant time slot periods is used to arbitrate transmissions while a dynamic mini-slotting-based scheme with variable length time slots and a position dependent arbitration[2] is used for transmissions within the dynamic segment. The symbol window is a communication period in which a symbol can be transmitted on the network. The network idle time is a communication-free period that concludes each communication cycle. Up to 64 nodes are supported in a FlexRay network that can be realized either via electrical or optical cables. FlexRay is planned to be used in power train and chassis application domains in the car. It was first deployed in BMW X5 cars in 2006.

2.2 IP/Ethernet-based Networks with QoS Support

The Ethernet technology (IEEE 802.3) [51], an OSI-Layer 2 protocol, is a local area network (LAN) standard developed by Xerox Co. in cooperation with DEC and Intel in 1976. It uses star or bus topologies as shown in Figure 2.2 and supports data rates of 10 Mbit/s, 100 Mbit/s (Fast Ethernet), 1000 Mbit/s (Gigabit Ethernet) and 10.000 Mbit/s (10 Gigabit Ethernet). Even faster Ethernet networks with transmission rates up to 100 Gigabit/s or more are already planned. The Ethernet technology uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to access the physical medium. This means that in order to access the network, each device

[1] Effective bit rate is 8 Mbit/s, i.e., 4 Mbit/s per communication channel.

[2] Nodes with earlier time slots can block nodes with later time slots.

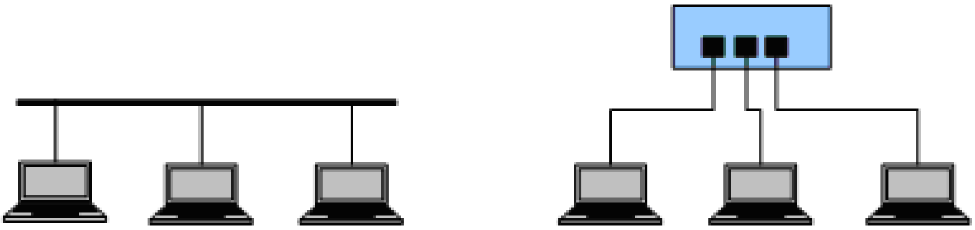


Figure 2.2: Bus and star topologies realized by the Ethernet technology

verifies whether the network is free. If the network is not free or signal collisions are detected, the device waits for a random amount of time before retrying.

Ethernet is the most widely used network technology in local area networks (LANs) and although, not initially implemented for media streaming is now beside FireWire (IEEE 1394) [52] the most widespread network technology, especially in the Internet for audio and video streaming. Because of the small royalty that Apple Computer and other patent holders have initially demanded from the users of FireWire, the more expensive hardware needed to implement it and also, the fact that only a few vendors offer consumer devices with FireWire interfaces, Ethernet has gained even more importance in the communication systems for audio and video transmission. However, as opposed to FireWire that enables isochronous real-time media streaming, the standard Ethernet fails in providing QoS and real-time capability for audio and video transmission and therefore, should be extended with other mechanisms.

The original Ethernet networks ran on coaxial cables and at 10 Mbit/s rates. These systems were known as 10base2 and 10base5. The coaxial cable was a shared medium for all connected devices where only one device could transmit at a time by using the CSMA/CD mechanism as mentioned above. When Ethernet was re-implemented on twisted-pair cables (UTP or STP), the CSMA/CD mechanism was retained, but for only two devices at each end of the cable. This type of Ethernet became known as half-duplex or semi-duplex, i.e., data packets flow in either direction, but only in one direction at once. Ethernet at 10 Mbit/s on twisted-pair cables is known as 10baseT. With the advent of twisted-pair cables between two devices, it became possible to separate the electrical signals traveling in each direction into different pairs of conductors in the cable and thus, realizing the full-duplex Ethernet, i.e., data packets flow in both directions at once without collisions. The full-duplex Ethernet requires improved Ethernet hardware at both ends where each end should know whether the other end is full-duplex capable. An Ethernet controller in a switched Ethernet network thus, never experiences any collisions if full duplex connectivity is used. However, packets can still be lost if

- the total network capacity exceeds the capacity of the switch engine, or
- the output buffer capacity is not sufficient when packets from several input ports compete for the same output port.

In order to solve the above mentioned issues in an IP/Ethernet network and provide QoS and real-time capability, several approaches have been proposed in the literature. Some of them, introduced by IETF, extend the OSI layer 3 as described in the following.

IntServ - Integrated Services [19] is a QoS mechanism for parametrization of IP packets. It reserves resources for each connection. RSVP (Resource reSerVation Protocol) is applied to initialize the resource reservation prior to the data transmission. It reserves required resources for unicast and multicast flows. Thus, it optimizes the bandwidth usage and eliminates congestions. The receiver sends a reservation message up the spanning tree to the sender. The message is propagated using the reverse path forwarding algorithm. At each hop the router examines the reservation request and if possible reserves the necessary resources and then passes the message. When the required resources are not available, the router sends back a failure. By the time the message is back at the receiver the resources are reserved all the way from the sender to the receiver and data can be transmitted on that path. Accordingly, intermediate routers have to calculate their available resources and accept or reject the incoming RSVP reservation requests. However, for dynamic network changes and a big number of participants, IntServ does not perform well because of the complexity due to resource reservation for each connection and tracking the continuous connection modifications in the network.

DiffServ - Differentiated Services [19] is a QoS mechanism based on the prioritization of IP packets. The priority flag of the IP packets is determined by the sender. The six bit priority flag is in the Type of Service field of the IP header. The first three bits define the class selector while the last three bits stand for the drop precedence. The intermediate routers decide only according to this priority flag how to handle the IP packets. In order to check the correctness of the assigned priority classes, some ”trust boundaries” are defined usually at the entrance to the routers. At these boundaries the administrator controls the correctness of the priority settings.

Some other approaches extend the Ethernet MAC with QoS and real-time capability. Those approaches that are widely applied, especially in the automation field, are briefly presented in the following.

PROFINET [110] is the successor of the field bus system PROFIBUS, based on the Ethernet technology. PROFINET is developed by Siemens and Phoenix Contact who founded the PROFIBUS International association (PI) and is available since 2002. It uses a TDMA scheme on top of the Ethernet MAC. A prioritization scheme based on the IEEE 802.1p and a hardware extension for clock synchronization are applied in the PROFINET protocol. In order to synchronize the cycles of the individual devices, PROFINET uses the IEEE 1588 protocol implemented in hardware. Thus, clock jitters are reduced to 1 μs. The necessary hardware is provided by only two manufacturers, Siemens and NEC.

EtherNet/IP (Industrial Protocol) [98] is a standard developed by Rockwell Automation and the OVDA (Open DeviceNet Vendor Association). It is mainly used in American automation fields, e.g., in manufacturing plants of General Motors. EtherNet/IP applies Ethernet switches, and TCP and UDP as transport protocols. The Common Industrial Protocol (CIP) is an application layer protocol that adapts Ethernet for industrial use. However, EtherNet/IP by itself does not guarantee hard real-time communication. Therefore, an extension of EtherNet/IP is under development, which is based on IEEE 1588 for the time synchronization and on the IEEE 802.1p for packet prioritization. Even though EtherNet/IP is widely deployed in production networks, there is no support organization for it and the majority of vendors integrate the components by themselves.

Ethernet Powerlink [38] is one of the oldest real-time Ethernet systems. It was specified by the company Bernecker&Rainer in 2001 and disclosed by the Ethernet Powerlink Standardisation Group (EPSG) in 2002. Ethernet Powerlink is based on a separate network without using conventional Ethernet devices. It follows another approach than the standard Ethernet, since it is used as a typical field bus system. The network structure does not include switches but hubs. It thus eliminates delay times caused by switches. One device in the network acts as the communication master and polls the device that is allowed to send in a certain time slot. This means a low bandwidth utilization due to the one active device at a time. Furthermore, the communicating devices and their time schedule should be pre-defined precisely, which requires a high effort. Also, the master should be protected and monitored in order not to fail the communication schedule. On the other hand, the system offers short deterministic delays due to the usage of hubs.

EtherCAT [37] is a product of the company Beckhoff and is supported by the EtherCAT technology group. It represents a strong mapping between a field bus and an Ethernet-based network. The devices are either connected to each other directly in one line or are connected via switches as a virtual line. The logical communication itself is based on a ring structure with a token rotating between the devices. However, EtherCAT does not take any advantage of Ethernet characteristics, e.g., it is not able to support priority traffic classes.

RTnet [39], [67] has been first defined in a diploma thesis from the University of Hannover. It is an open-source project for Linux and is freely available. RTnet implemented as a software framework, offers various functionalities on top of a real-time Linux system enhanced with RTAI (Real-time application interface) [102] and utilizes standard Ethernet hardware. It uses a TDMA-based access scheme. The time synchronization is performed by a timing master with the software implemented IEEE 1588 protocol. All timing slaves adjust their clocks and sending times according to the the first time slot in the network sent by the timing master. The sending times of time slots can be dynamically changed during the operation, which increases the flexibility and the bandwidth utilization. Priorities are assigned at the software level. RTnet can only be used in a reasonable way if hubs are used. But it also works with switches, however with a lower timing performance considering the switch delay times. Packet tunneling is applied for non-real-time data through the real-time network protocol stack of RTnet. Tunneled packets are encapsulated by the RTmac protocol that manages the real-time NIC (Network Interface Controller) in order to differ between otherwise identical real-time and non-real-time data packets. The main drawback of RTnet is that all nodes, i.e., real-time as well as non-real-time nodes should support the RTnet protocol stack as mentioned in [67].

Time-Triggered Ethernet [49] unifies the transmission of real-time and non-real-time data into one communication system. It distinguishes between the event-triggered (ET) standard Ethernet traffic and the time-triggered (TT) traffic that has strong time constraints. TT Ethernet messages are distinguished by the Ethernet type field with $0 \times 88d7$. They are further extended by a TT message header that contains the information about the TT control, message length and parameters. All TT Ethernet devices are synchronized and have the same time base through the applied precise synchronization mechanism in oder to guarantee a deterministic communication. TT Ethernet is based on a uniform time format that is characterized by the two parameters, granularity and horizon [95]. The granularity of the TT Ethernet clock determines the minimum interval between two adjacent ticks of a digital clock or the smallest time interval that can be measured with this time format, which is 60 nanoseconds. The horizon of the TT Ethernet clock determines the instant when the time will wrap around, which is about 30000 years and is even much higher than the horizon of the GPS to avoid the wrap around problem in a realistic future. In addition to the TT Ethernet messages, there are also the Protected Time Triggered (PTT) messages that define the data category with the strongest time, safety and precision requirements. For PTT messages, the so called Guardian devices continuously monitor and control the TT Ethernet switches for a correct functionality. The so called Unprotected Time Triggered (UTT) Ethernet messages are not controlled by the Guardians and have therefore, a lower priority than the PTT Ethernet messages. UTT messages could be from multimedia and streaming applications that have time constraints but are not as critical as the PTT messages. Both, the TT (the same for UTT) and the PTT Ethernet data require different hardware implementations for the NICs, i.e., network cards with two physical ports (one to the host and one to the TT Ethernet switch) for TT and with three physical ports (one to the host and two to two TT Ethernet switches) for the PTT Ethernet data [49] are required. TT and PTT Ethernet messages are forwarded by TT switches with a preemptive cut-through mechanism while the competing ET messages are stored and only forwarded when no other TT messages are left to be sent. The main disadvantage of the time-triggered approach is the proprietary hardware requirement, which indicates a high production cost.

2.2.1 Avionic Full-Duplex Switched Ethernet (AFDX)

Due to the continuous growing of safety-critical and passenger entertainment applications in avionic systems with the demand for larger transmission resources, the need for broadband on-board data buses is increasing. The requirement for rapid deployment concepts with minimal development and implementation costs has driven the avionic industry to explore existing off-the-shelf technologies. Both, Boeing and Airbus investigated the standard Ethernet technology, IEEE 802.3 to build a next-generation avionic data bus for the Dreamliner Boeing 787 and the Airbus A380, respectively. This investigation resulted in the development of Avionics Full-Duplex Switched Ethernet (AFDX)[3], which is based on the standard Ethernet technology and extended by specific functionalities to provide a deterministic network with guaranteed services [30], [14].

Although Ethernet offers high transmission rates and low cost due to the widespread commercial usage, it does not offer the required robustness for avionic systems and does not guarantee the QoS. AFDX attempts to solve these issues while applying as much commercial Ethernet hardware as possible. It addresses the shortcomings of Ethernet by using concepts from the Telecom standard, Asynchronous Transfer Mode (ATM). The major aspects of AFDX are as follows:

[3] AFDX is applied in the Airbus A380. Boeing applies Common Data Network (CND), which follows a similar concept as AFDX in the Dreamliner Boeing 787.

- Profiled and static network - System parameters are defined in the configuration tables that are loaded into switches at system startup. Also, end-systems are parametrized before data transmission is initiated.
- Full duplex Ethernet - The physical medium is twisted pair with separate pairs of wire for transmission and reception channels.
- Switched network - The wired Ethernet network supports the star topology. Each switch connects a maximum of 24 end-systems. Switches can also be cascaded to construct a larger network.
- Determinism - The AFDX network emulates a deterministic, point-to-point network by using virtual links (VLANs) with guaranteed transmission resources.
- Redundancy - Dual networks per connection provide a higher degree of reliability than a single network scheme.
- Data rate - The network operates at either 10 Mbit/s or 100 Mbit/s. The default mode is 100 Mbit/s.
- Traffic shaping and policing - End-systems and switches control their output and input rates according to the pre-defined configurations, thus guaranteeing the QoS.

[76] has shown and proven the deterministic bounds of an AFDX network by means of analytical derivations based on the Network Calculus Theory [137].
Although, AFDX provides a highly reliable and deterministic network with QoS guarantee, it requires high production cost due to the redundant networks. Since AFDX is a pure static network, dynamic topology changes, such as plug and play are not supported within this network.

2.2.2 Audio Video Bridging

Ethernet, as a widespread computer network technology has not been widely applied for real-time audio and video streaming. This is first, because Ethernet does not provide an isochronous and deterministic low-latency transmission that is required for streaming applications. Second, because the market was not large enough to adapt Ethernet for audio and video streaming. As a result, other technologies such as FireWire have been applied for media streaming in LANs. However, in the last few years, Ethernet or its modifications have repeatedly been applied for high quality audio and video (AV) networking. For example, the Cirrus Logic ”CobraNet” has been used for audio distribution in auditoriums, and the Gibson Guitar ”MaGIC” has been designed and demonstrated for use in live performances [142]. CobraNet uses a ”limited topology” in terms of a limited number of devices and hops to define delay, jitter and loss upper bounds, while MaGIC uses a non-standard bridge to stream high quality media. With the introduction of High Definition-TV (HDTV) in the beginning of 2004, and with the recent success of the digital audio, e.g., by ”iPod”, the idea appeared for a standardized AV network technology based on Ethernet. In July 2004, an IEEE 802.3 study group [13] was built from the institutions Pioneer, Nortel, Gibson Guitar, Samsung, Broadcom, NEC and others to verify the idea of an isochronous[4] Eth-

[4] Isochronous data should be transmitted periodically within certain time intervals to prevent jitter and provide synchronized audio and video at the receiver side.

ernet network, the so called *Residential Ethernet*. The Residential Ethernet project was moved to the IEEE 802.1 task group in 2005 and since then is renamed to Audio/Video Bridging (AVB) Ethernet[5]. Several well-known institutions such as Intel, Apple, Micrel, Marvell, Cisco, Sony, HP and Harman International have joined the standardization extending the working group that was built for Residential Ethernet.

In the AVB Ethernet, the MAC layer is extended by the 802.1 AVB MAC that supports three new functionalities for isochronous data streaming, i.e., IEEE 802.1AS (time synchronization), IEEE 802.1Qat (admission control and resource reservation) and IEEE 802.1Qav (traffic shaping and time-aware scheduling). During the network startup, after the auto negotiation phase, AVB devices exchange logical link discovery (LLD) messages to inform other AVB devices that they are AVB capable.

The IEEE 1588 precision time protocol (PTP) [47] is applied for the clock synchronization in the AVB network. A timing master is selected during the network startup. All other devices (timing slaves) adjust their clocks to the master clock. The master sends periodically a sync message with its local time to all slaves. This precise time synchronization has two purposes. First, it allows isochronous, i.e., time-aware and low jitter traffic shaping and scheduling that is packets are sent within certain time intervals. Second, it provides a common time base for sampling data streams at sources and reconstructing them at receivers with the same frequency. The isochronous transmissions are based on 8 kHz cycles, which is the rate commonly used in most isochronous networks such as IEEE 1394 and USB.

Each sender advertises its contents to all receivers in the network via a special higher layer protocol, e.g., the Zero Configuration Networking (Zeroconf) protocol [54]. The device that is likely to receive a stream sends a registration message to the sender. Thus, the sender and all intermediate switches know about the potential receivers and their locations in the network. Switches insert the information about the available resources into the registration message on the way to the sender. The sender answers with a reservation message that triggers admission control operations in the intermediate switches. It also reserves resources, i.e., data rate and buffer size for the required stream. If the admission control fails in an intermediate switch, the so called Status Indication (SI) bit of the reservation message will be set to zero (failed). All following switches do not reserve any resources and the receiver is informed by the SI flag. Otherwise, the stream will subsequently be sent to the receiver with guaranteed QoS.

Before transmission, the source shapes the data flow according to a pre-specified traffic shaping function to introduce certain desired traffic characteristics to the data flow [84]. The shaped data flow is also re-shaped in the intermediate switches to compensate the previously introduced jitter to the data, e.g., due to head-of-line blocking (HoLB) and to avoid bursts.

In an AVB network, different data flows are assigned with different priority levels from IEEE 802.1p. The network management data has the highest priority, the priority levels 6 and 7. Isochronous data packets are assigned with priority levels 4 (moderate latency streaming with 1 ms time cycle period) and 5 (low latency streaming with 125 μs time cycle period), while legacy Ethernet data uses the lower priority levels, 3 to 0. This means that in an AVB network, legacy Ethernet messages can be transmitted without any modifications.

It is possible to apply the AVB network without IP. The layer 2 protocol IEEE 1722 [32] and the layer 3 protocol IEEE 1733 [16] that are realized in hardware are used in an AVB network for session management, packetization, media format definition, in other words to enable isochronous data transmission without IP and layer 4 transport protocols.

[5] For more information, see `http://ieee802.org/1/pages/avbridges.html`.

Accordingly, the AVB standard realizes an Ethernet network that guarantees very low jitter and a maximum end-to-end delay time of 2 ms over seven cascaded switches for isochronous data transmission [87] and that supports plug and play, i.e., dynamic topology changes. However, the network complexity is high due to the time-aware data scheduling and the dynamic resource reservation. The AVB standard is not yet finalized and is supposed to be passed by the beginning of 2009.

2.3 Video Compression and Image Processing in the Car

2.3.1 Video Compression - Basics and Codecs

For a resource-efficient video transmission in the car, video compression is inevitable. Currently, raw camera video streams are first transmitted via digital LVDS cables to the processing ECU. The analogue output data is sent via CVBS cables from the image processing ECU to the receiver device, e.g., the headunit (Figure 2.1). The headunit converts the analogue signal into the digital format and sends it via LDVS to the display device. However, camera video streams can be more efficiently streamed as compressed digital data over one transmission medium all the way from the source to the display device. In the following, the basics of image and video compression are briefly introduced.

Compression means reduction of the number of bits required for data representation. Compression ratio is defined as $c = \frac{s_{compressed}}{s_{original}}$, where $s_{compressed}$ represents the size of the data after compression and $s_{original}$ is the size of the original (uncompressed) data. Due to their higher compression efficiency, lossy compression algorithms[6] are used in this work. Figure 2.3 shows the block diagram of a typical video encoder [46; 50; 70; 71; 101; 125]. In the following, all major processing steps of the video codec from Figure 2.3 are described.

Pre-processing and Color Space Conversion

Before data compression is initiated, it is essential to match the signal to the requirements of the receiver by removing any information that is not relevant to the receiver. For video, this includes the spatial resolution and color space conversion that are explained in A.1. Also, artifacts negatively influence the compression performance and should therefore be removed in the pre-processing steps. A ”smoother” image, especially the one containing less noise or sudden intensity changes, can be compressed more easily. Pre-processing steps for in-vehicle cameras additionally include the exposure time adjustments, adjustments to the varying lightning conditions, lens shading correction, camera calibration, and correction of the effects of the typical wide angle lenses. These are tightly coupled with the image sensor. For some cameras (e.g., the rear-view camera) the frames of the video stream have to be flipped horizontally for displaying.

Transform Coding and Quantization

In video compression, transform coding refers to a frequency transform used to exploit the spatial redundancy within a frame and to de-correlate information relative to its perceptibility by

[6] Lossy compression algorithms irreversibly change the original data during the encoding process as opposed to lossless codecs. The decoder can subsequently not fully restore the original data without any distortion.

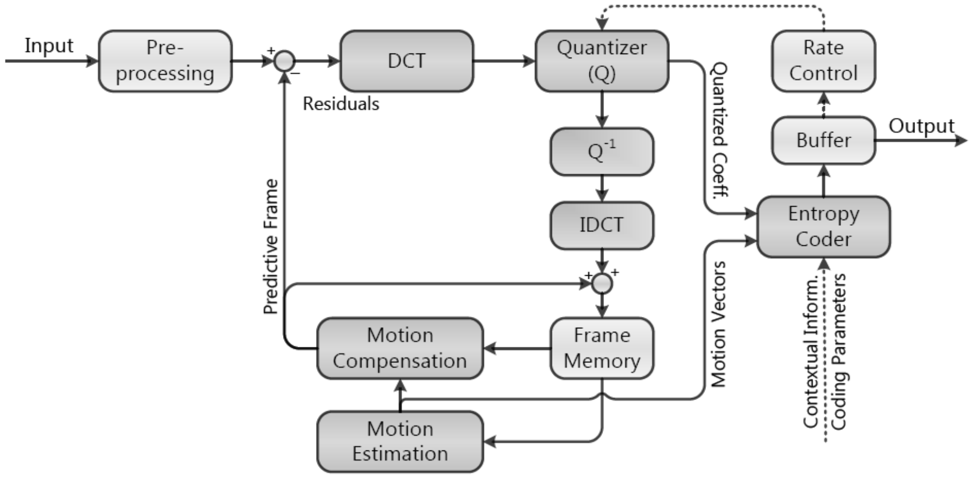


Figure 2.3: Typical video encoder based on motion compensated prediction and the discrete cosine transform (DCT) with the darker fields denoting the essential steps of encoding.

the human eye. Two types of transform coding are commonly employed, the discrete cosine transform (DCT) [79; 115] and the Discrete Wavelet Transform (DWT) [114].

Most commonly used video codecs, such as MPEG codecs use DCT [115]. They subdivide single frames in square blocks of typically 8 × 8 pixels and apply the DCT to convert the spatial representation of the image into the frequency domain[7]. Some other compression algorithms such as Motion JPEG 2000 apply DWT. According to [121], a ”wave” is mathematically expressed as a sinusoidal (or oscillating) function of time. Thus, a ”wavelet” is a small wave whose energy is concentrated in the time domain. More details about DCT and DWT can be found in [101].

The actual compression is achieved by quantization [72], i.e., taking advantage of the psycho visual or perceptual redundancy within the DCT or DWT coefficients. The quantization process (Q in Figure 2.3) reduces the number of possible values that have to be encoded by mapping the transform coefficients to discrete values based on their perceptual significance. For example, high-frequency transform values are quantized more coarsely than the low-frequency values. Also, the quantization factors usually differ between the luma and chroma planes as explained in Section A.1. Quantization is the main factor for the compression efficiency. By introducing a scale factor to the quantization process, the compression ratio can be controlled to either achieve a desired bit rate or compress an image or a video frame at a defined maximum quality degradation (distortion). Quantization is a lossy operation. Therefore, de-quantization (Q^{-1} in Figure 2.3) cannot exactly reproduce the original transform values.

Lossless (Entropy) Coding

The lossless entropy coding exploits statistics in the pattern of the quantized transform values in order to reduce the number of bits furthermore. Variable length coding (VLC) and arithmetic

[7] Generally, 64 multiplications are performed in the DCT matrix multiplication for an 8 × 8 block. In [133], an 8 × 8 DCT algorithm is presented in which the DCT matrix is factored into separate matrices. It performs only 5 multiplications, 8 scaling operations, and 29 additions. By being separable, the algorithm allows for efficient parallel hardware implementations.

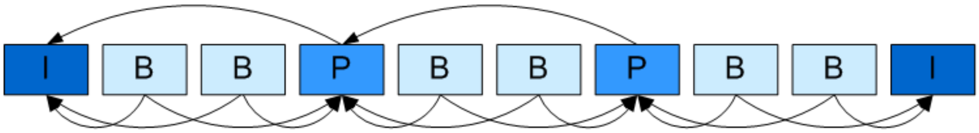


Figure 2.4: Inter-frame dependencies between I-, P-, and B-frames.

coding are the two widely used algorithms for entropy coding. More details can be found in [101].

Motion Compensated Prediction

All introduced processing steps so far exploit redundancies within a frame and are equally applicable to the individual frames of a video sequence. Changes in consecutive frames are typically very small in a video sequence, e.g., when the background is static movements are limited to single objects within a frame. Therefore, motion compensated prediction is used in video compression to take advantage of the similarities between two temporarily adjacent frames in a video sequence. The goal is to predict the next frame from the recently coded video frame. Therefore, each frame that is to be used as a reference has to be internally decoded in the codec and kept in the frame memory. Motion estimation is the process of finding an area in the reference frame that is most similar to each macroblock[8] of the current frame and leads to the smallest prediction error [25]. Generally, in the reference frame, the search area for motion estimation is shifted by integer pixel values. Enhancements in current video coding standards allow the motion estimation process to be performed with half- or quarter-pixel shifts to enable a more precise encoding of movements. The spatial displacement to the best match in the reference frame is referred to as the motion vector. In motion compensation, the motion vector is used to reduce the effects of motion [101]. Consequently, only the prediction error, also called the residual, has to be compressed by transform coding and quantization. The entropy coding is finally accomplished for the quantized residual coefficients, the motion vectors and the synchronization information. Thus, three different frame types **I (Intra frame)**, **P (Forward predicted (inter-)frame)** and **B (Bi-directionally predicted frame)** are produced as shown in Figure 2.4. A frame sequence starting with an I-frame and ending before the next I-frame, is called a **group of pictures (GoP)** and is typically continuously repeated throughout the video sequence.
Motion estimation and compensation require the highest computational power in video compression by consuming 60–80% of the total computational time [100]. The key factors that determine the complexity of motion estimation and compensation are the search precision, the search mode, and the search range[9].

Rate Control

The output bit rate of an encoder with a constant quantizer usually varies with changes in the spatial domain and the amount of movement in a video scene up to an order of a magnitude. This is referred to as a variable bit rate stream and is explained in Section 2.4.2 in more detail. The changes of the amount of transmitted data over time can be determined as minimum bit rate, mean bit rate, and instantaneous or peak bit rate. However, all codecs can also be configured to

[8] In block-based motion compensated prediction, the current frame is subdivided into macroblocks with a typical size of 16×16 pixels.

[9] The search mode defines the algorithm and the matching criteria that are used while the search range defines the number of surrounding macroblocks on which the search is performed.

produce an almost constant bit rate stream. In a typical video encoder as shown in Figure 2.3, a buffer is placed at the output of the entropy encoder. The rate control module shown in Figure 2.3 measures the current bit rate. It acts as a feedback mechanism to the quantizer and regulates the quantization degree in order to adapt the video stream to the desired bit rate. If the output bit rate is higher than desired, the quantizer scale will be increased by the rate control algorithm to increase compression and vice versa. The drawback of the constant bit rate compression approach is the lower video quality in complex scenes compared to the variable bit rate compression for the same desired bit rate.

Artifacts and Post-Processing

Similar to pre-processing in the encoder, post-processing is often used in the decoder. Different filters can be applied for post-processing to reduce the coding artifacts. Post-processing filters usually reduce blockiness, the most common artifact in DCT-based compression algorithms. Advanced codecs (e.g., H.264/AVC) apply the so-called in-loop de-blocking filter during encoding and decoding. There is also a wide range of errors that may occur over error-prone channels, including single or multiple bit errors, packet loss or burst losses. Depending on the type and the location of the errors, different schemes of error concealment are applied in the decoder [123].

Video Codecs

In the following, video compression algorithms that have been selected for analysis in this work are shortly introduced.

MPEG-2 (Video) [58] was jointly standardized by MPEG and ITU-T in 1994. The ITU-T norm is H.262. The main advances of MPEG-2 compared to previous standards, e.g., MPEG-1 include the addition of tools for interlaced coding mode, the support of higher frame resolutions, scalable video coding (SVC), and wider motion compensation ranges. The search ranges are adapted to the larger frame sizes and the interlaced encoding mode. MPEG-2 covers the typical standard definition TV resolutions PAL and NTSC with 720 × 576 pixels at 50 fields per second (25 frames/s) and 720 × 480 at 60 fields per second, respectively.
As the most prominent applications, the Digital Video Disc (DVD) and Digital Video Broadcast (DVB) employ the MPEG-2 compression standard with variable bit rate at typical rates of 4-8 Mbit/s. In later standard refinements, MPEG-2 was extended to support high definition TV (HDTV) resolution. It is currently available in premium cars through the DVD-players, e.g.
MPEG-2 introduces the concept of profiles and levels. It defines combinations of five profiles and four levels, which are summarized in Tables 2.1 and 2.2. More details can be found in the

Profile	Frames	YUV	Remarks	Levels
Simple (SP)	P, I	4:2:0	no interlaced coding, not scalable	ML
Main (MP)	P, I, B	4:2:0	most widely used, not scalable	all
4:2:2 (422P)	P, I, B	4:2:2	not scalable	ML
SNR scalable	P, I, B	4:2:0	quantization scaling	LL, ML
Spatial	P, I, B	4:2:0	spatial scaling	H-14
High (HP)	P, I, B	4:2:2	spatial or SNR scaling	ML,H-14,HL

Table 2.1: Profiles defined in MPEG-2 (Video).

Level	Frame Size (pixel) and Rate (frame/s)	Bit Rate
Low (LL)	352 × 288 at 25 or 352 × 250 at 30	4 Mbit/s
Main (ML)	720 × 576 at 25 or 720 × 480 at 30	15 Mbit/s
High 1440 (H-14)	1440 × 1152 at 30	60 Mbit/s
High (HL)	1920 × 1152 at 30	80 Mbit/s

Table 2.2: Typical frame sizes, frame rates, and bit rates for the levels defined in MPEG-2 (Video).

standardization document or in [81]. The most common combination is the Main profile at Main level, which is designed for standard definition TV. The combination of Main profile at High level is applicable for high definition TV.
The MPEG Licensing Authority (MPEG LA) [138] provides a way to license a portfolio of required patents for the use of MPEG-2 Video as an alternative to negotiate separate licenses. The typical license fee is US-$2.50 per encoding and per decoding unit. However, it does not assure that all possibly essential patents are included.

MPEG-4 (Part 2: Visual) [60], in the following referred to as MPEG-4 was first released in 1999 with corrections and enhancements that followed in later years. It was primarily designed for low bit rate streaming applications, but extended to support high quality and high bit rate streaming applications as well as file storage. MPEG-4 builds upon MPEG-2. It supports progressive and interlaced video coding with resolutions smaller than QCIF (176 × 144 pixels) and beyond HDTV (1280 × 720 pixels), i.e., bit rates varying from the kbit/s range to more than 1 Gbit/s. The main new features of MPEG-4 compared to MPEG-2 are summarized in the following [50; 71; 105].
Advanced motion compensated prediction: Unrestricted motion vectors, an extended range of quantized coefficients, motion estimation with half-pel accuracy per default with an option to use 1/4-pel accuracy, and global motion compensation (GMC) to predict the average motion of a frame as a whole with the possibility of defining a global translation, scaling (zoom), and rotation of the frame.
MPEG quantization type: All AC coefficients[10] are quantized using a constant scalar value while the DC coefficients are quantized using a partially linear mapping. Generally, the MPEG quantizer type preserves more details and is better suited for medium to high bit rate encoding.
Error Resilience: Sliced encoding with resynchronization, data partitioning, reversible VLC and new prediction (feedback channel for channel status announcements).
Scene graphs and content-based functionalities: Definition of a multimedia framework with frames as so-called video object planes (VOPs) being of rectangular or also arbitrary shapes.

MPEG-4 Visual defines a total of 19 profiles covering different applications in video compression. The most common profiles for rectangular and natural video sequences are the Simple

[10] The scaled average intensity value of an image block is called DC coefficient. For natural images it is typically the largest value among all DCT coefficients. All other coefficients are called AC coefficients.

Profile (SP), the Advanced Simple Profile (ASP), and the Advanced Real-Time Simple profile (ARTS). The SP is intended for basic error resilient coding using I- and P-frames. It does not cover the advanced motion estimation tools mentioned above. In the SP, four levels are defined with a maximum bit rate of 384 kbit/s and CIF frame resolution (352 × 288 pixels). The ASP extends the SP by adding support for the enhanced coding features, such as B-frames, 1/4-pel motion compensation, user defined quantization tables, and global motion compensation. Its highest level defines a typical frame size of 720 × 576 pixels and a maximum data rate of 8 Mbit/s. The ARTS profile is defined for the use in real-time video transmission systems with a feedback channel from the decoder. It additionally supports advanced error resiliency features. It also provides four levels with CIF frame size and a maximum bit rate of 2 Mbit/s in the highest level. The other 16 profiles of MPEG-4 include support for higher bit rates, higher bit depths, scalable video coding, and arbitrary shaped video objects [40].

Similar to MPEG-2, also MPEG-4 uses patents from several companies requiring the payment of license and royalty fees. A license portfolio for MPEG-4 is also provided by the MPEG LA. Among others, the terms depend on the number of units sold and as the encoder and decoder units are generally acquired from a supplier, the exact terms may vary. Yet, as a general figure, a typical fee per unit is US-$0.25.

H.264/AVC (Advanced Video Codec) [59] is a development of the Joint Video Group (JVT), a collaboration of MPEG and ITU-T. H.264/AVC is the state-of-the-art in video coding technology. It was developed to provide better coding performance and a higher flexibility for use in different networks and applications, reaching from high quality, low bit rate streaming and conversational services over various physical networks to High Definition storage and broadcasting over cable and satellite. Just as its predecessors, H.264/AVC is a block-based and motion compensated video codec that by using its enhancements reaches up to 50% bit rate reduction with no perceptual quality loss compared to MPEG-2 [45] at the cost of a higher computational complexity[11]. To realize its flexibility, H.264/AVC defines a video coding layer (VCL) that represents the coded video content and a network abstraction layer (NAL) [113]. The network abstraction layer defines a clear syntax structure in the form of NAL units that can be used for direct mapping to various transmission networks (e.g., to IP packets) without the need of parsing the encoded bit stream. Therefore, the frame sub-partitions (syntax elements) created during encoding are mapped, together with contextual information, into NAL units consisting of a one byte header and the syntax element. The units are assigned with a type tag and an importance tag according to their degree of negative impact the loss of that unit would have for decoding. For example, the loss of an I-slice would result in a higher quality degradation than the loss of a B-slice. Depending on the transmission network or channel, strategies could be applied to better secure the transmission of more important NAL units.

The new tools of the H.264/AVC standard include small block size, short word-length, and exact-match inverse transform; arithmetic and context-adaptive entropy coding (e.g., CAVLC and CABAC); spatial “intra” prediction; quarter-pixel precision and macroblock partitioning

[11] In [77] and [119], complexity analyses for major H.264/AVC encoding and decoding tools are presented. According to these tests, the overall encoder complexity of H.264/AVC (MP) increases with an order of magnitude as compared to MPEG-4 (SP). The results also show the asymmetry of H.264/AVC. By using a “basic configuration” the encoder/decoder complexity ratio is about ten and can increase considerably with the addition of more advanced tools from the standard. A complexity analysis for a Baseline profile decoder can be found in [88].

with variable block sizes; weighted prediction, multiple reference frames, and reference decoupling; in-loop de-blocking filter; switching slices; error resilience tools. More details about the above mentioned tools can be found in [101].

Profiles and Levels: Three profiles have been defined in the initial H.264/AVC standard release, the Baseline profile, the Main profile, and the Extended profile. As the simplest profile, the Baseline profile is intended for applications with low delay and low complexity requirements, such as real-time services. The Main profile supports the core of the H.264/AVC tools and is intended for asymmetric applications, e.g., broadcasting. The Extended profile provides all tools supported by either the Baseline or Main profiles, except for CABAC. The tools defined for each profile are summarized in Table 2.3. The profiles are not subsets of each other as in previous standards. For instance, a Main profile decoder cannot decode all videos encoded by a Baseline profile encoder. H.264/AVC defines a total of 15 levels, which can be used in combination with any profile of the standard. An overview of the profiles and levels can be found in [44].

Table 2.3: H.264/AVC Baseline, Extended, and Main profile with the supported tools.

Tool	**Baseline**	**Extended**	**Main**
I and P Slices	+	+	+
B Slices	–	+	+
Switching Slices	–	+	–
Quarter-pixel motion compensation	+	+	+
Different Block Sizes	+	+	+
Intra Prediction	+	+	+
In-Loop De-blocking Filter	+	+	+
Multiple Reference Frames	+	+	+
CAVLC	+	+	+
CABAC	–	–	+
Error Resilience: FMO, ASO, RS	+	+	–
Error Resilience: DP	–	+	–
Weighted Prediction	–	+	+
Interlaced Coding	–	+	+

Also the H.264/AVC standard is covered by several patents. Besides MPEG LA, a second patent portfolio can be licensed from Via Licensing [126] covering some overlapping and some different patents. To have a general idea about the codec price, a typical fee is US-$0.20 per unit.

Motion JPEG In Motion JPEG, the frames of a video sequence are encoded separately as JPEG images [57] using its Baseline profile. The basic compression algorithms of JPEG are the same as for intra frame coding in the MPEG standards [46]. Motion JPEG is mostly applied for video editing, surveillance, and the movie function in digital cameras. It has a lower compression efficiency than the MPEG standards. But it has the advantage of being simple and low complex. Since JPEG is an established standard, there is a large number of implementations that are highly optimized and available at low cost. Motion JPEG does not require any license fees. The main disadvantage of Motion JPEG is the lack of a standard syntax description. Thus, compatibility problems between different implementations may occur.

Motion JPEG 2000 has been standardized as the Part 3 of the JPEG 2000 standard [62] by the Joint Photographic Experts Group (JPEG), a sister organization of MPEG and a sub-organization of ISO/ICE. Motion JPEG 2000 is based on compression of individual frames by using the JPEG 2000 [61] still image format. JPEG 2000 employs the DWT and achieves up to 50% better compression ratios than the DCT-based JPEG. JPEG 2000 and Motion JPEG 2000 are targeted to be royalty free [75]. The main drawback of JPEG 2000 is its required processing power that is up to 10 times higher than the DCT-based JPEG using standard PC equipment due to the use of DWT and the enhanced feature set. One major source of complexity in JPEG 2000 is the entropy coding, because it uses an arithmetic approach adding complexity to both, encoder and decoder. With hardware specialized codecs, the computation time can be reduced. The major improvements of (Motion) JPEG-2000 over (Motion) JPEG are as follows:

- Higher quality for a given data rate or more efficient compression at a given distortion limit
- Support for lossless and lossy compression
- Scalability can be achieved for both, SNR and resolution, without the need to save redundant or additional data.
- Variable choice of color spaces and higher bit depths per component (up to 16 bit)
- Errors during the transmission result in blurring and are distributed throughout the frame. This is often less disturbing than the blocking artifacts caused by the DCT-based coding schemes.

In [77], a comparison between H.264/AVC coding restricted to I-frames and Motion JPEG 2000 shows a similar coding efficiency. The authors state, that for frame sizes up to 1280×720 pixels, the compression efficiency of H.264/AVC (MP) intra mode coding is only slightly better than the efficiency of Motion JPEG 2000.
No license fee is required for the use of JPEG 2000 standard.

Dirac The open source and license-free Dirac video codec [120] is being developed by the British Broadcast Corp. (BBC) and uses the wavelet transform coding. The goal is to provide a royalty-free, simple, general-purpose video coding algorithm for a wide range of resolutions (from QCIF to HDTV), which is competitive to other state-of-the-art codecs in terms of compression efficiency. Open source VHDL hardware implementations of both encoder and decoder are available (`http://www.opencores.org/projects.cgi/web/dirac/overview`). However, all these projects are still under development. Due to the insufficient timing performance of the Dirac software codec, it is not considered in the time delay analysis of Section 5.1. The main techniques used in Dirac are:

- Wavelet transform coding
- Bit depts up to 16 bits per component with 4:2:0, 4:2:2, and 4:4:4 chroma subsampling
- Motion estimation and compensation based on overlapped blocks with variable block sizes to reduce block artifacts

- Global motion compensation, such as in MPEG-4, to describe translation, scaling, rotation, and three dimensional rotation
- Frame modes: I, L1, and L2 frames, similar to the MPEG frame modes I, P, and B
- Rate-distortion optimization mode
- Entropy coding with adaptive arithmetic coding
- Support for a lossless coding mode

In [78], a comparison with H.264/AVC shows that Dirac, even in its current development status, is competitive to the state-of-the-art standards in terms of perceptual image quality and compression efficiency.

Theora Theora [131] is an open source codec being developed by the Xiph.Org Foundation. Open source FPGA hardware implementations are also available [130]. Theora is based on the VP3 codec by On2 Technologies [97]. The Theora codec operates in a similar way as the MPEG standards. It uses an 8×8 DCT and block-based motion compensation. Besides I-frames, Theora supports P mode inter-frame prediction with multiple reference frames. But it has no equivalence to B-frames of MPEG. Theora uses 1/2-pel precision for motion estimation. Generally, Theora is intended to be used with the Ogg Format for file storage, but can also be implemented for streaming in networks. Its drawback is that it does not provide special tools for error resiliency. Other main features of the Theora syntax are the support of 4:2:0, 4:2:2, and 4:4:4 chroma subsampling with 8 bits per component, the support for custom quantization matrices and non-linear scaling of quantization values. Theora uses a flexible VLC scheme and an adaptive in-loop de-blocking filter. Due to the insufficient timing performance of the Theora software codec, it is not considered in the delay analysis of Section 5.1.

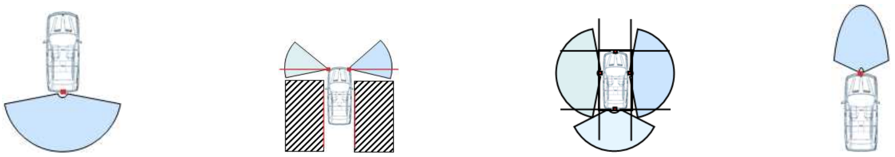
2.3.2 Image Processing in Driver Assistance Systems

Driver assistance camera systems can be divided into two application groups, direct and indirect image-based driver assistance services [10].

Direct Image-based Driver Assistance Services

The term direct image-based driver assistance comprises all image-based assistance functions that utilize the presentation of the image itself to assist the driver. This assistance service category implies that all applications share a display resource and therefore cannot work in parallel. The most common driver assistance services that belong to this category are listed in the following.

Rear-View Camera This driver assistance camera represents the rear area of the car via one video stream to the driver as shown in Figure 2.5(a). The presentation of the video usually takes place in the central display (CID in Figure 2.1). The frame rate is 30 frames/s.



(a) Rear-view *(b)* Side-view *(c)* Top-view *(d)* Night-vision

Figure 2.5: Direct image-based camera systems in the car.

Side-View Cameras This driver assistance service provides the picture of the right and left sides of the car's front area by two video streams to the driver (Figure 2.5(b)). The videos should help the driver to avoid collisions while entering into a road with unclear sight conditions. The two video streams are combined and presented at the central display. The frame rate is 30 frames/s.

Top-View Cameras This driver assistance function presents the right, left and rear areas of the car via three video streams to the driver (Figure 2.5(c)). The videos should help the driver in parking situations. The three video streams are combined and presented at the central display. The frame rate is 30 frames/s. Fish eye distortion algorithms are used because of the fish eye lens, which is deployed within the left and right cameras [28].

Night Vision Camera This driver assistance service presents the front area of the car via one video stream to the driver as shown in Figure 2.5(d). This functionality supports the driver with infrared images from the front side of the car while driving at night. There are two ways to implement this feature. Either two infrared spotlights emit infrared light and a camera within the car records the illuminated street or a far infrared camera mounted in front of the car is used. The frame rate is 30 frames/s.

Indirect Image-based Driver Assistance Services

The term indirect image-based driver assistance comprises all image-based assistance services that instead of using the image itself extract the needed information from the camera images and only use this information for driver assistance, e.g., to trigger an action based on this information. The most common driver assistance functions that belong to this category are listed in the following.

Lane Departure Warning This driver assistance function warns the driver when the car unexpectedly crosses the lane. The operation of this functionality is like all indirect image-based driver assistance functions not limited by any other functions within the car. So far, this function is realized with the front-view camera that is installed near the rear view mirror in the car. The frame rate can be set to 30-60 frames/s.

Traffic Sign Recognition Because of the rapid change of traffic signs it is hardly possible to keep the navigation data updated. The traffic sign recognition function tries to overcome this problem by detecting traffic signs while driving. Similar to the lane departure warning, also the traffic sign recognition function uses the front-view camera. The frame rate can be set to 30-60 frames/s.
Accordingly, indirect driver assistance services apply image processing algorithms. Image processing in the automotive domain means object recognition and is the task to identify an object and its type within pictures of a video sequence [94]. To achieve this goal, the so called Regions Of Interest (ROI) are identified by globally searching images for objects or by evaluating data from external sensors. In the car, the ROI are commonly selected by external sensors such as radar or 3D-cameras. The next processing steps are feature extraction and classification. The Edge Histogram Descriptor (EHD) of MPEG-7 [109] and Histogram of Oriented Gradients (HOG) [92] have been chosen for the analysis in the spacial domain of images, i.e., they are applied to every single video frame. The Optical Flow [66], [24] has been additionally applied to take the temporal domain of a video sequence into account. The Support Vector Machine (SVM), a supervised learning method, has been used to classify objects after feature extraction as this method achieves good classification results [33]. SVM is trained with features of well-known objects before it can classify unknown objects. EHD, HOG and Optical flow are briefly described in the following.

EHD: The Edge Histogram Descriptor has emerged to be the most suitable among the MPEG-7 descriptors in the context of object recognition. EHD subdivides each image into 4×4 non-overlapping equally sized sub-images. For each sub-image a histogram of five different types of edges is calculated and stored in five bins, a vector of 80 histogram bins is generated. Each sub-image is further subdivided into a fixed amount of equally large blocks of size of a multiple of two. Digital filters are applied to the spatial domain of each block in order to determine the predominant edge type of that block. In the case no edge is found in a block, this certain block is classified as a non-edge block and is not represented within the histogram. To generate the edge histogram, the number of blocks is counted and normalized by the total number of blocks within a sub-image.

HOG: This approach classifies objects based on the distribution of local intensity gradients or edge directions. Each image is subdivided into 8×8 pixel cells. The gradient of each cell is calculated using a gradient filter mask. For color images separate gradients are calculated for each color component. The one with the greatest norm is chosen to be the gradient vector of a specific cell. After calculating the gradient orientation, it is binned to one of nine sectors. This means that sectors of the size $\frac{180^\circ}{9} = 20^\circ$ are generated while the 'sign' of the gradient is ignored and the gradient orientation is mapped into one of those sectors. The pixel bins are accumulated into orientation bins for each cell and weighted by the respective gradient magnitude (i.e., the intensity of a pixel at position x, y). To generate the final descriptor, each four cells are grouped together to 16×16 pixel blocks and all blocks are put together with the contrast information.

Optical flow: This metric is calculated based on differences between two consecutive video frames. According to [24], optical flow is the "distribution of apparent velocities of movement

Table 2.4: Summarized domain classification.

Automotive Domains	Domain Functions
Chassis	Data communication for the operation of the Chassis (stability, agility and dynamics of the car)
Power train	Data communication for the operation of the power train (engine, gearbox etc.)
Driver assistance	Autonomously operating data and communication (without user intervention), supporting the driving situation
Infotainment/HMI	Data communication interacting with the driver concerning the operation and driving situation of the car (Route and traffic related information)
Comfort	Non-driving related data, communication concerning well-being of driver and the passengers
Entertainment	Car and driving unrelated data, audio and video for entertainment

of brightness patterns in an image". It is computed either from movements of the objects in a scene or from movements of the viewer (camera). $I_x u + I_y v + I_t = 0$ is the basic equation to be solved using side constraints in order to calculate the optical flow [80] where $I_{x,y,t}$ are intensity values of a pixel at position x, y at time t. The optical flow vector is subsequently denoted as $v = (u, v)$. There are a lot of different approaches that define different side constraints [66]. For the analysis in the automotive domain, a texture-driven, table-based indexing scheme that applies the Census transform [43] is used.

2.4 In-Vehicle Traffic

Traditionally, automotive networks are divided into several domains that correspond to different functionalities, constraints and structures. The classification and the number of automotive domains have not been standardized yet. Therefore, many different approaches can be found in the literature such as [17], [42], [48], and [22]. It can be seen that the number of domains is increasing with the publication date of these approaches as the number of automotive ECUs is increasing as well. Furthermore, it is apparent that the domain borders are not well defined, because some applications in different domains differ only in a few aspects. By incorporating the different definitions and considering future applications, six automotive domains can be defined as shown in Table 2.4. The domains are sorted according to their safety significance from top to bottom. A classic automotive domain is interconnected via an appropriate communication network. With the growing number of ECUs, a clear assignment of one automotive domain to one network system is not possible anymore. Figure 2.6 shows a mapping of the current automotive network systems to the introduced domains in Table 2.4. It can be seen that almost every domain is covered by more than one network technology. The reason is mainly the growing number of ECUs. Some ECUs connect to more than one network system in order to distribute their data. Accordingly, there is an increasing amount of traffic relayed between the networks by the so

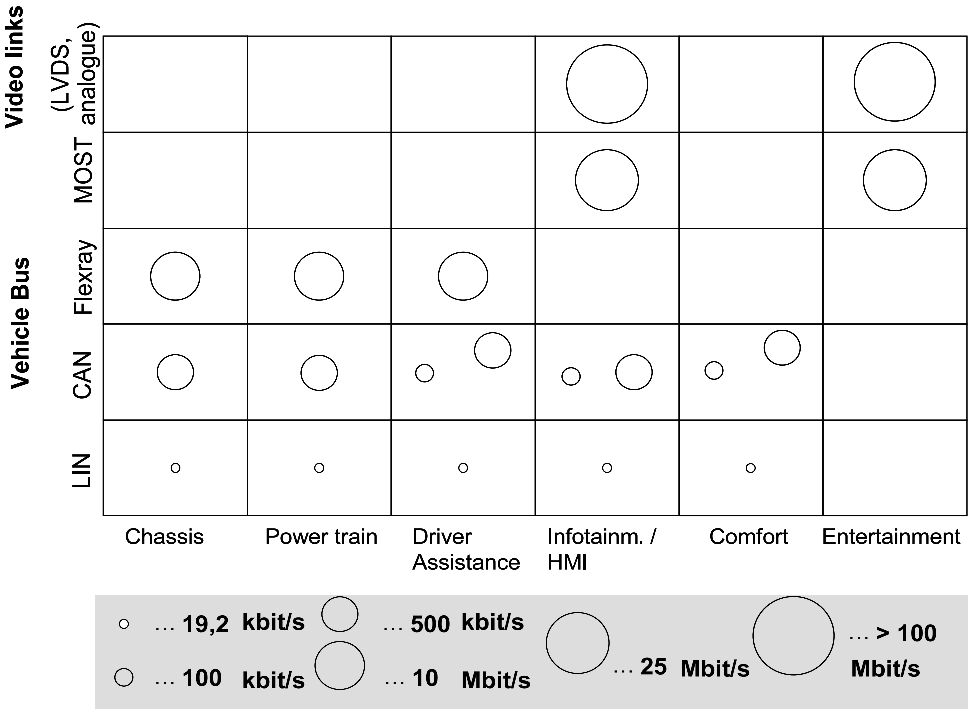


Figure 2.6: Mapping of automotive communication networks to the different domains

called gateways. Gateways are relay units that convert data types at the application layer. As a result, the increasing number of ECUs leads to an overlap of the domains, i.e., similar applications can be found in various domains.
In the following, a new classification of the in-vehicle traffic is presented with all requirements that are considered in the analysis of this work. Further on, an analytical method is introduced and configured to model the in-vehicle traffic.

2.4.1 Requirement Analysis

Today's in-vehicle traffic comprises of applications with different characteristics and QoS requirements as mentioned in [6] and [9]. The QoS requirements define the bounds for the throughput, end-to-end delay, delay jitter, loss and error rate. Depending on these requirements, the in-vehicle traffic can be classified into four different groups as follows.

Real-Time Control Data:

This traffic class defines the highly safety-critical control data with the strongest end-to-end delay requirement of 2.5 ms [12] for applications such as X-by-Wire. FlexRay applications belong to this data category. It also includes control data with lower delay requirements such as PT- and K-CAN applications. They transmit sensor data, but also control information for driver assistance. The maximum end-to-end packet delay derived from CAN cycle times amounts to 10 ms.

Real-Time Audio and Video Streams:

Audio, but mostly video data from camera systems or video transmitting sensor systems of driver assistance services belong to this data category. Interactive audio and video applications such as Voice over IP (VoIP) and video conferencing are also part of this data category. We assume a maximum of 5 cameras in the car that stream compressed video simultaneously. As mentioned before, video compression is applied for an efficient use of network resources in future cars. System requirements are given as follows.

- Transmission rate: 7.4 Mbit/s for video data (average data rate of MPEG-4 compressed video streams from [5]), 50.8 kbit/s for VoIP when using the G.726 voice codec which emits a voice packet every 20 ms
- Frame rate: 30 frames/s
- I-frame interval: 15 frames [5]
- Frame resolution: VGA (640×480 pixels)
- End-to-end delay: Max. 33 ms for cameras excluding the video processing time, max. 150 ms for VoIP according to ITU-T G.114
- Packet loss rate for an adequate media quality: very low

As mentioned before, MPEG-coded videos consist of 3 frame types, i.e., I-, P- and B-frames. However, the presence of B-frames imposes extra delay to the coding and decoding processes due to the fact that B-frames make use of forward and backward prediction [108]. For this reason, we apply the MPEG-4 compression algorithm without B-frame coding for the real-time video streams[12]. Our analyses based on software implemented video codecs have shown that compression and decompression amount to 10 to 20 ms of time. This time can considerably be reduced by application of hardware implemented codecs as described in [5].

Multimedia Data:

Multimedia systems transmit audio and video data for entertainment of the car occupants. QoS requirements of the assumed in-vehicle multimedia applications are defined in the following.

- Transmission rate: 4-8 Mbit/s (MPEG-2 is mostly applied for multimedia video applications.), 128 kbit/s (MP3), 1.4 Mbit/s (Audio CD)
- Frame rate: 25 frames/s
- I-frame interval: 12 frames[124]
- Frame resolution: 720×576 (PAL)

[12] In this work, the real-time criterion for a camera application implies that each of the processing steps, particularly encoding and decoding, has to be completed before the next frame is available, i.e., within one frame interval. This should be understood as a theoretical upper bound.

- End-to-end delay: max. 100 ms for Audio CD, max. 200 ms for DVD[13]
- Packet loss rate for an adequate media quality: very low

The assumed resolution levels for the multimedia streams do not exclude the future application of higher resolution levels.

Best Effort Data:

These applications do not require any QoS. They are not delay sensitive so that lost packets can be retransmitted again without any constraints on delay or jitter. This type of traffic includes web-browsing data, system maintenance or downloading data such as downloading a digital map to the navigation system.

The traffic categories real-time control data, real-time audio and video streams, multimedia and best effort data have been considered in the analysis of the IP/Ethernet in-vehicle network. The analysis, however, excludes the highly safety-critical control data that is required to be transmitted separately by the FlexRay bus and some CAN segments according to the current requirement of the car manufacturers.

2.4.2 Traffic Modeling

According to 2.4.1, the in-vehicle traffic is defined through different data classes. Understanding the nature of in-vehicle traffic classes is important for an appropriate network design. Statistical models are used in the following to characterize the in-vehicle data for further analysis. In-vehicle traffic is divided into two main groups of constant bit rate (CBR) and variable bit rate (VBR) data.

Constant Bit Rate Data

Constant bit rate (CBR) means that the amount of data sent over the time is always constant. In order to analyze a highly loaded network in the worst case transmission scenario, the real-time control data and the audio streams are both modeled as CBR applications with the highest possible bit rate.
For the real-time control data, it is assumed that the 8-byte K- and PT-CAN messages are packed into 64-byte Ethernet frames to be sent over the IP/Ethernet network at data rates of 800 kbit/s and 4 Mbit/s, respectively[14]. These rates are 8 times higher than the original K- and PT-CAN

[13] [104] suggests that a synchronization time of 240 ms between video and image sources is acceptable. In [118], the speed of visual attention was studied implying the time period of shifting the fovea focus. The paper cites different sources in which delay times of 75 to 175 ms are reported and presents experimental results with an attention delay of about 140 ms for peripheral events. In [85], the impact of delays for multi-user games has been researched where both, audio and video data are important and suggests delays up to 100 ms as absolutely acceptable. The maximum delay times for in-vehicle multimedia audio and video streams are derived from the mentioned references as approximated values.

[14] 4.8 Mbit/s is the highest achievable transmission rate of Ethernet packets containing K- and PT-CAN messages and is modeled here as CBR traffic for the worst case analysis. In reality, K- and PT-CAN have a VBR traffic and reach only sporadically the mentioned maximum bit rate.

rates of 100 kbit/s and 500 kbit/s (Figure 2.6), which is the result of larger Ethernet frames. Also, in-vehicle audio sources such as Audio CD or VoIP are modeled as CBR applications. In both cases, the transmission rate is constant, i.e., fixed size data packets are transmitted at constant time intervals according to [108].

Variable Bit Rate Data

MPEG videos imply a variable bit rate due to the different frame types, i.e., I-, P- and B-frames with different sizes as shown in Figure 2.7. An accurate modeling of the rate variability is essential for an adequate resource planning. Several source models have been introduced in the literature [26], [86] that assess the MPEG-compressed video structure with I-, P- and B-frames. According to [108], the introduced source models can be classified into two main groups, the Markov-based models and the self-similar models. In this work, we apply self-similar models, because they need less input parameters than the Markov-based models, which means that they have a lower configuration uncertainty. The long range dependence (LRD) property of a compressed video sequence is the autocorrelation function of its GoP (shown in Figure 2.8) and shows a proof for time correlation in a video stream, especially when the lag is less than 80 GoPs. [26] and [86] suggest that this characteristic of a video stream has important effects on the network performance and should therefore be included in the video source model. We apply the fractional autoregressive integrated moving average (F-ARIMA) model as a realization of self-similar models.

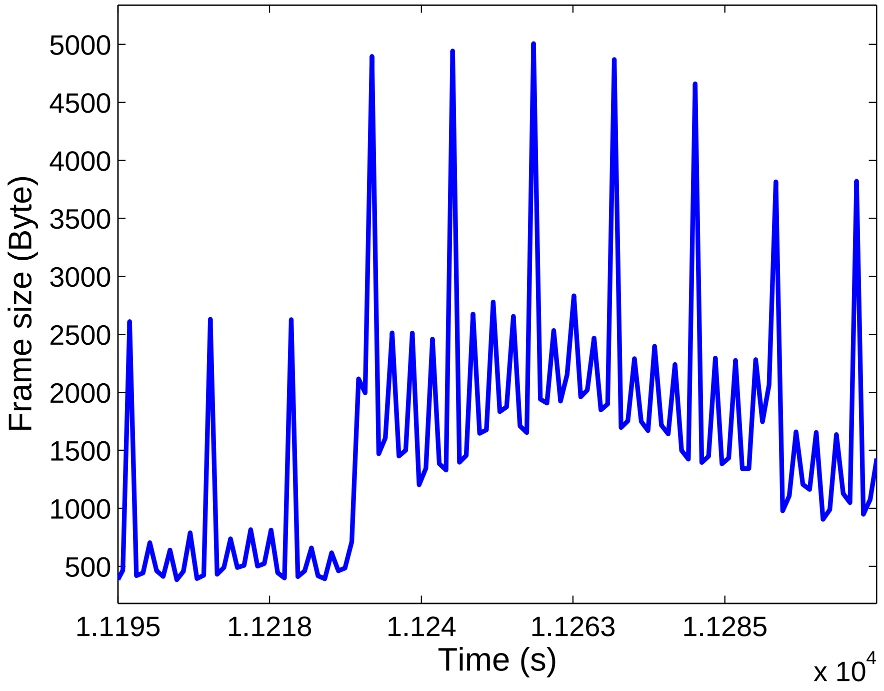


Figure 2.7: The traffic envelope of VBR applications.

The F-ARIMA Video Model The F-ARIMA model is a statistical model to generate a video stream. It is able to faithfully generate a stream with the LRD characteristic and different frame

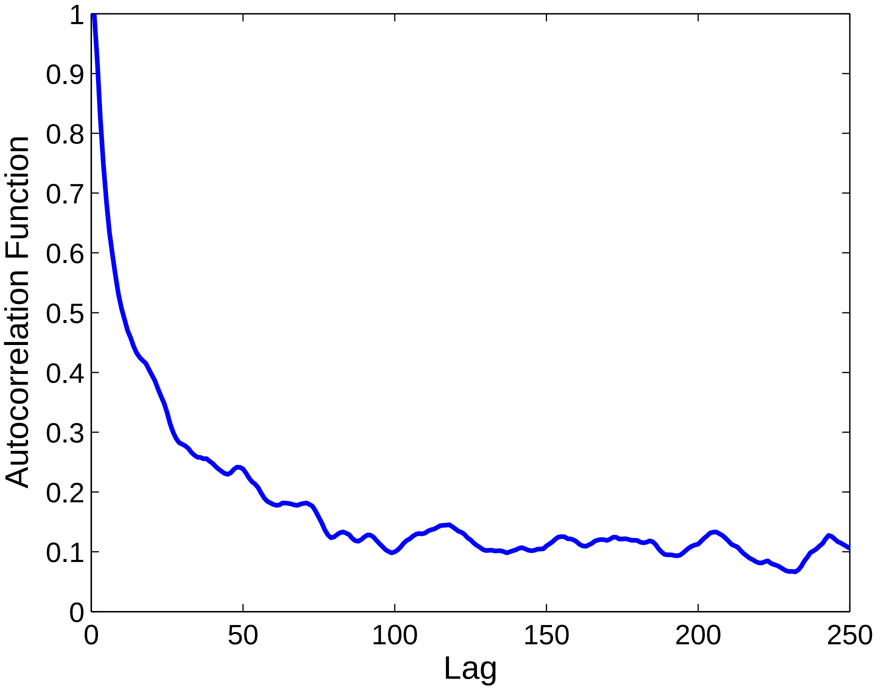


Figure 2.8: Autocorrelation function of a real compressed video stream at the GoP level.

sizes to model different frame types. Additionally, the distribution of frame sizes in a GoP can be selected to be any desired distribution. Due to the special interest of network planners in the tail distribution of the MPEG-coded video representing large frames and its frequent modeling by the Pareto distribution, we applied the Pareto distribution to model the GoP as explained in the following.

Modeling the LRD Characteristic The LRD characteristic is included in the F-ARIMA model by estimating the autocorrelation function of a real video stream. The autocorrelation function is partly determined by the Hurst parameter (H), which can be estimated in several ways from real frame sizes. More details can be found in [26] and [86] about how to estimate the H parameter from empirical data graphically. According to [26], after the H parameter has been estimated, the autocorrelation function at GoP level can be modeled as

$$r(k) = \begin{cases} \sum_{i=1}^{j} w_i exp(-\lambda_i k) \ \text{ for } k < K_t \\ \\ Lk^{-B} \ \text{ for } k \geq K_t. \end{cases} \tag{2.1}$$

where

$$B = 2(1-H)$$

$$\sum_{i=1}^{j} w_i = 1 \tag{2.2}$$

$$LK_t^{-B} = \sum_{i=1}^{j} w_i exp(-\lambda_i K_t)$$

The K_t in these equations represents the ”knee”, which is a transition point of the autocorrelation function from an exponentially decreasing function to a gradually decreasing function.

Constructing a Background Sequence with LRD The background sequence X is a sequence of normally distributed random variables x with the LRD property, i.e., $X = \{x_1, x_2, x_3, ..., x_v\}$. Each x can be transformed into a GoP as explained in the following steps. Accordingly, v is equal to the number of GoPs in the video sequence. Given the estimated autocorrelation function $r(k)$ from Eq. (2.1), X can be computed by

1. x_0 is generated from a Normal distribution $N(0, \sigma^2)$, $N_0 = 0$ and $D_0 = 1$.

2. For $k = 1, ..., v$, calculate the following variables recursively

$$\begin{aligned} N_k &= r(k) - \textstyle\sum_{j=1}^{k-1} \phi_{k-1,j} r(k-j) \\ D_k &= D_{k-1} - N_{k-1}^2 / D_{k-1} \\ \phi_{k,k} &= N_k / D_k \\ \phi_{k,j} &= \phi_{k-1,j} - \phi_{k,k}\phi_{k-1,k-j} \quad j = 1, ..., k-1 \end{aligned} \tag{2.3}$$

3. Use the above parameters to compute the mean and variance of each x and then to calculate the x_k by

$$\begin{aligned} m_k &= \textstyle\sum_{j=1}^{k} \phi_{k,j} X_{k-j} \\ \sigma_k^2 &= (1 - \phi_{k,k}^2)\sigma_{k-1}^2 \\ x_k &= N(m_k, \sigma_k^2) \end{aligned} \tag{2.4}$$

Constructing a GoP with LRD $G = \{g_1, g_2, g_3, ..., g_v\}$ defines the sequence of GoPs (g) in a video stream and can be transformed from the normally distributed X. As mentioned before, the distribution of G is chosen to be the Pareto distribution $f(x) = \alpha \frac{\beta^\alpha}{x^{\alpha+1}}$ (for $x \geq \beta$, α the steepness of the slope and β the lower bound of the random variable x) in this work. The transformation is done as follows.

$$G = F_p^{-1}(F_N(X)) \tag{2.5}$$

where $F_N = 1/2 \left(1 + erf\left(\frac{x-m}{\sigma\sqrt{2}}\right)\right)$ is the cumulative distribution function of the Normal distribution with the mean m and variance σ^2 and $F_p^{-1}(x) = \beta \left(1/(1-x)\right)^{1/\alpha}$ is the inverse cumulative distribution function of the Pareto distribution[15]. The transformation in Eq. (2.5) preserves the LRD characteristic of X in the G.

[15] erf is the error function. For n = 1,..., 6, $erf\left(n/\sqrt{2}\right)$ is defined to be smaller than 1.

Changing the Sequence of GoP into a Sequence of Frames According to [108], the GoP sequence G can be reformed into a sequence of picture frames by applying ratios between I-, P- and B-frames over the GoP size to each element of G. Let $ratio_I$, $ratio_P$ and $ratio_B$ be ratios of mean I-, P-, and B-frames of a video stream with an IBBPBBPBBPBB structure over the mean GoP size, i.e.,

$$ratio_I = I_{mean}/GoP_{mean} \tag{2.6}$$

$$ratio_P = 3 \cdot P_{mean}/GoP_{mean} \tag{2.7}$$

$$ratio_B = 8 \cdot B_{mean}/GoP_{mean} \tag{2.8}$$

Accordingly, the i^{th} GoP can be constructed by

$$GoP_i = \left\{ ratio_I \cdot g_i, \frac{ratio_B \cdot g_i}{8}, \frac{ratio_B \cdot g_i}{8}, \frac{ratio_P \cdot g_i}{3}, \frac{ratio_B \cdot g_i}{8}, \ldots,, \frac{ratio_B \cdot g_i}{8} \right\} \tag{2.9}$$

The three ratios for the three frame types can be obtained from a real DVD stream. However, the ratios for the driver assistance cameras without B-frames and with an I-frame interval of 15 [5] must be modified from the statistics of the real DVD stream (I-frame interval: 12 including B-frames [124]).
In the following, the F-ARIMA model is computed for the DVD application in the car based on the video sequence ”Star Trek: First Contact” from [124].
The H parameter needed to construct the autocorrelation function is provided by [124] to be 0.831 for the movie ”Star Trek: First Contact”. The autocorrelation function for this stream is estimated by trial and error as shown in Figure 2.9 to be

$$r(k) = \begin{cases} 0.4 \cdot exp(-0.15k) + 0.6 \cdot exp(-0.038k) \text{ for } k \leq< 26 \\ \\ 0.6963 \cdot k^{-0.338} \text{ for } k > 26. \end{cases} \tag{2.10}$$

Once the autocorrelation function has been estimated, X can be generated according to Eq. (2.3) and Eq. (2.4) for this video stream. In order to transform X to the Pareto distributed G, α and β parameters are required. α can be defined by comparing the Complementary Cumulative Distribution Function (CCDF) of the real video stream and the Pareto distributed stream with different values of α. As shown in Figure 2.10, $\alpha = 5.0$ turned out to be the most appropriate value. The β parameter can be calculated from the mean of the Pareto distribution as follows.

$$E(x) = \frac{\alpha\beta}{\alpha - 1} \tag{2.11}$$

Note that x in Eq. (2.11) is a Pareto distributed random variable. The mean of this Pareto distribution is the average GoP size. In other words, $GoP_{mean} = E(G)$ and GoP_{mean} can be calculated from the desired bit rate of the simulated stream. For example, in the case of the DVD application with an average bit rate of 4.7 Mbit/s at 25 frames/s, $GoP_{mean} = ((4.7 \cdot 10^6)/25) \cdot 12$ = 2.256 Mbit = 282 kByte for an I-frame interval of 12 frames. By substituting GoP_{mean} and $\alpha = 5.0$ in Eq. (2.11), β is found to be 225.6 kByte.
After computing all required parameters of Eq. (2.5), the background sequence X can be transformed to the GoP sequence G. However, initial results showed that by simply transforming to G, some GoPs turn out to be too large, which have adverse effects on the network performance. Therefore, an upper bound has been introduced for the resulting GoP size by computing the ratio

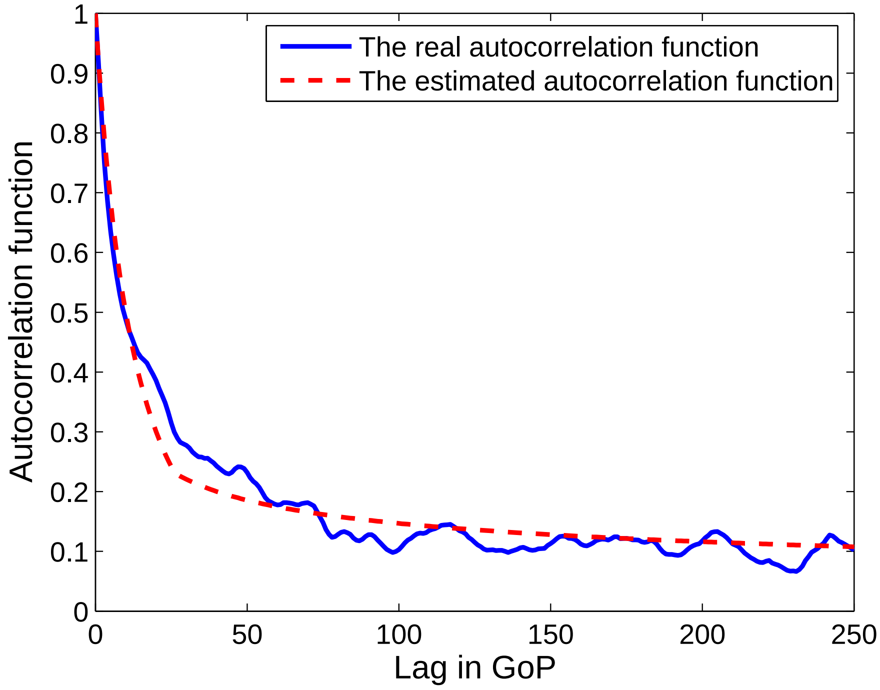


Figure 2.9: The estimated autocorrelation function and the real autocorrelation function from the video stream "Star Trek: First Contact".

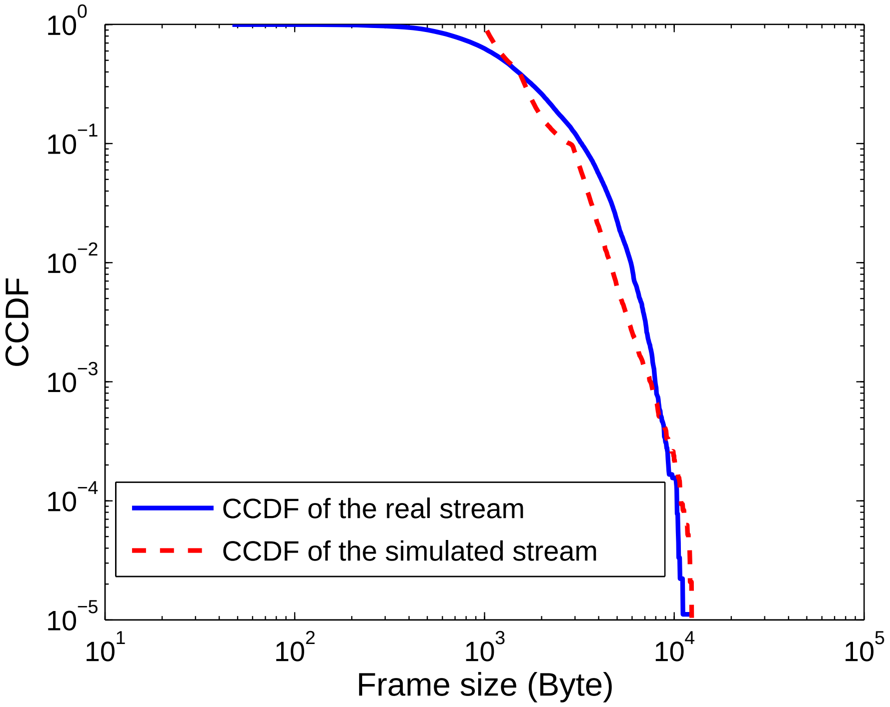


Figure 2.10: Comparing the CCDF values of the generated frame sizes and the real frame sizes given in bytes. The axes have a logarithmic scale.

between the largest GoP to the mean GoP size in the real video stream. This ratio defines the upper bound on the modeled GoP size in the sequence G. Any modeled GoP that is larger than this upper bound will be regenerated again based on the Pareto distribution. The upper bound for the GoP size has been determined to be 3.399 for the movie ”Star Trek: First Contact”.
The $ratio_I$, $ratio_P$ and $ratio_B$ used in Eq. (2.9) to construct a DVD stream are determined from the trace file ”Star Trek: First Contact” [124] to be 0.186, 0.293 and 0.521, respectively. Finally, these ratios have been applied to each element of the sequence G to get the modeled DVD stream shown in Figure 2.7.
Table 2.5 summarizes all statistical values applied to video sources in the analysis of this work.

Table 2.5: Configuration table for in-vehicle video sources (e.g., DVD and driver assistance cameras). a and b are the number of P- and B-frames in a GoP, while P_{mean} and B_{mean} define the average P- and B-frame sizes, respectively.

Parameter	Value		Description
	Camera	DVD	
I_{max} [Byte]	128510	178121	The size of the largest I-frame in the stream
I_{mean} [Byte]	54904	57191	The average I-frame size
F_{mean} [Byte]	24412	25630	The mean size of all frames in the stream
n [frames]	15	12	The number of frames per GoP
f [frames/s]	30	25	Frame rate
$T_{e2e,max}$ [s]	0.033	0.1	Maximum allowed end-to-end delay per frame
GoP_{mean} [Byte]	456569.38	282197	Mean GoP size
$GoP_{upperbound}$	2.331	3.399	GoP_{max}/GoP_{mean}
GoP_{max} [Byte]	1064263.22	957638.88	Maximum GoP size
$ratio_{I-frame}$	0.136	0.186	I_{mean}/GoP_{mean}
$ratio_{P-frame}$	0.864	0.293	$a \cdot P_{mean}/GoP_{mean}$
$ratio_{B-frame}$	0	0.521	$b \cdot B_{mean}/GoP_{mean}$
α	2.5	5.0	Pareto Shape parameter

2.5 Summary

In this chapter, an overview is given of the existing automotive network systems and their application fields. Due to the growing number of applications in the car, the requirements on automotive networks are continuously increasing. The deployment of different network technologies and point-to-point links interconnected via gateway systems leads to an inflexible network architecture and a complex cable harness in the car, which is expensive and requires a high validation and management effort.
IP over Ethernet is the most widespread network technology in computer networks and is increasingly gaining importance in QoS-aware communication systems such as its application in aircrafts, automation systems and also home networks. Ethernet offers enough transmission capacity, e.g., for the simultaneous transmission of several video streams, which is one of the transmission scenarios in the car. As described in [4], Ethernet also has a good error detection

capability compared to automotive network systems. Accordingly, IP over Ethernet represents a serious alternative to the current automotive networks and is therefore concretely analyzed for its adaptability as an in-vehicle communication system in this work.

The basics of video compression and several widespread video codecs are introduced. Video compression is assumed to be applied to automotive video systems, such as multimedia and driver assistance camera systems for an efficient resource usage. Typical automotive image processing algorithms are briefly presented to be investigated in Chapter 5 for compressed camera video streams.

The in-vehicle traffic is described and classified into four groups based on QoS requirements of the automotive applications. Precise traffic modeling is essential for adequate network planning. In this chapter, the fractional autoregressive integrated moving average (F-ARIMA) model has been selected for further analysis due to its frequent use in the literature and its comparably simple configuration. The adjustment and employment of the F-ARIMA model for in-vehicle video sources have been described on the basis of the DVD application. All configuration parameters applied for in-vehicle video sources (e.g., cameras and multimedia sources) have been summarized.

3 Proposed Network Architecture

In order to cope with growing complexity and to comply with future automotive demands, a new network architecture for in-vehicle communication is needed. In this chapter, an IP/Ethernet-based network architecture is proposed for future in-vehicle communication. Adequate network topologies are introduced. The QoS performance of the proposed network topologies is evaluated analytically and via simulations. Both evaluation methods are described and adapted to the in-vehicle requirements from Section 2.4.1. The QoS performance is computed based on the provided cost in terms of network resources and components. The wired core network is extended with a wireless peripheral network to increase user flexibility and reduce cabling effort. The introduced analytical and simulation models are extended for the wireless communication and evaluation results are presented for the selected worst case transmission scenarios.

3.1 Heterogeneous IP-based In-Vehicle Network

As discussed in [6] and [12], to realize a consistent network architecture IP can be considered as an abstraction layer between the applications and physical network technologies as shown in Figure 3.1. While the data link layer differs for wired and wireless physical layers and different transport protocols such as TCP and UDP can be used in layer-4 to enable reliable and

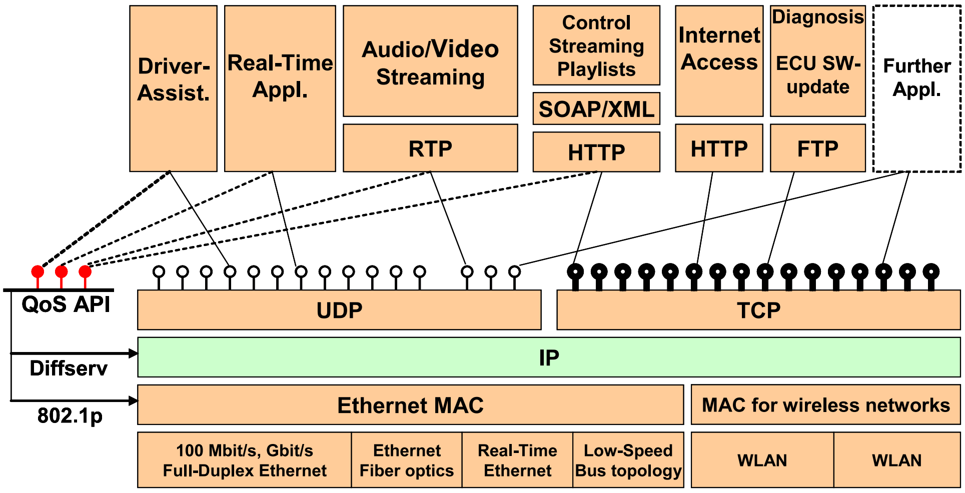


Figure 3.1: Proposed layered model (Filled circles represent the introduced QoS-API, empty ones represent the ports for fast and low delay transmissions, double lined ones represent the ports for reliable transmissions).

non-reliable but also connection-oriented and connectionless transmissions (See [3] for more details.), the layer-3 is consistent throughout all application domains in the car. If the data rate requirements grow between the vehicle generations and the old network technology does not fulfill the demands anymore, it can easily be replaced with a faster network technology without the need to change the application software. An example is the migration from Fast Ethernet (100 Mbit/s) to Gigabit Ethernet in the vehicle communication backbone. The modular and standard-based network architecture from Figure 3.1 provides different standard protocol suite implementations for different applications, for example Universal Plug and Play (UPnP) for controlling consumer electronic devices, the Real-time Transport Protocol (RTP) for streaming audio and video data or the Simple Network Management Protocol (SNMP) for network management.

Besides software, the IP-layer can be realized in hardware in order to avoid software-caused jitter. While present automotive network standards such as CAN and MOST require statically predefined functions and use a complex API for the communication between ECUs, IP-based standards take advantage of using standardized IT-protocols to transmit data without any function predefinition. Also, gateways will be resigned from the in-vehicle network, since in the IP-based network architecture there is no need for data conversion. Thus, the overall in-vehicle communication complexity and consequently cost can be reduced. Additionally, IP-based networks provide a wide range of development and measurement tools with lower costs. A prominent example is the open source protocol analyzer *Wireshark* which is widely used in the Internet area and is available free of charge.

3.1.1 Considered Network Topologies

Cars typically have a mixture of standard and client-specific electronic equipments. This leads to a large number of different possible variants for each car that has to be covered by an appropriate network topology. The topology should also be extensible to incorporate novel equipments that have to be added to the car during its lifetime. The goal is to find a flexible network topology that covers all different equipments and variants. For this purpose, the network topologies star, double star, daisy chain, and unidirectional ring shown in Figure 3.2 have been taken into account. Daisy chain turns out to be unappropriate due to high stockpile cost requirements for the maintenance of cable harness that is needed for different car equipments. Among star and double star, the double star topology is more adapted for in-vehicle communication. First, because the traffic load is distributed over two switches, i.e., the switch load can be better controlled. Second, there is already the possibility to integrate two switches in the car, e.g., in the headunit and in the central gateway. This means that the installation cost does not increase a lot for the double star topology. The unidirectional ring topology requires a lower cabling effort (2 wires instead of 4 wires of the twisted-pair Ethernet cable) and a smaller number of ports compared to the double star network, because the required 3-port switches can be integrated in the ECUs. The ring topology requires a low installation cost, because it can easily substitute one of the present widespread automotive networks, the MOST ring.

Consequently, double star and ring topologies have been selected for further analysis in this work. They are indicated as double star and ring networks from now on. The final overall in-vehicle network topology will probably be a combination of double star and ring, since both of them have certain advantages. The following discussions and evaluations are to better understand the benefits and drawbacks of these topologies. The results shall provide insights into the choice

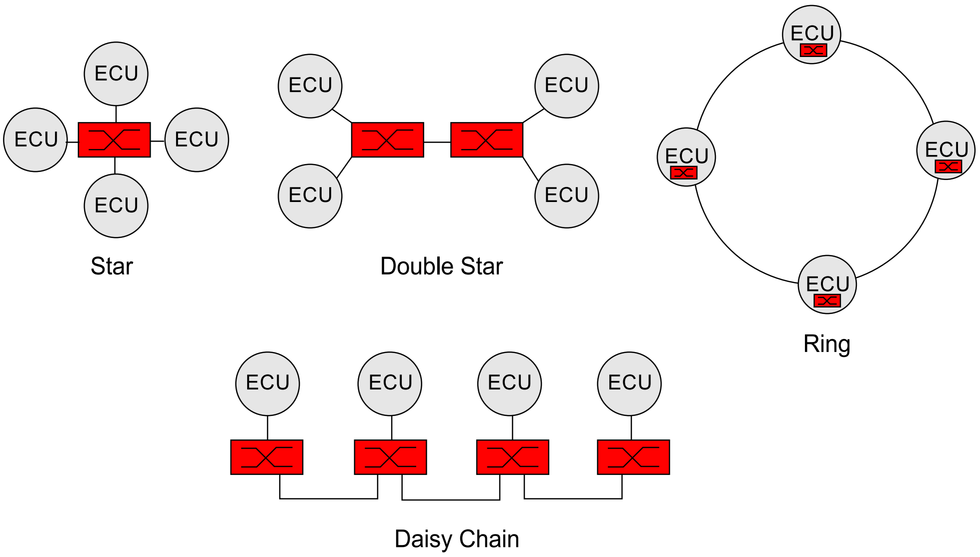


Figure 3.2: The network topologies, star, double star, ring and daisy chain.

of the future in-vehicle network topology. It should be mentioned that the low-cost bus topology will also be integrated in future cars to interconnect low data rate ECUs. However, this is not the subject of this work and will not be further discussed.

Double Star

In the star topology, each device is directly connected to a switch. For connections 4-wire twisted pair cables are needed. Due to the electromagnetic compatibility issue in the car, a Fast Ethernet network is assumed. In the analyzed double star network shown in Figure 3.3, all five cameras are connected to Switch B. A simultaneous transmission of these five cameras leads to a large burst at the output port of Switch B towards Switch A. For the worst case analysis, it is assumed that all five cameras transmit 100 Mbit/s at the same time. The main advantages of the double star topology are the high network resilience in the case of client outage and the flexibility in choosing link capacities for different parts of the network. The main disadvantages are the high cabling effort and the need for external switches, which requires extra cost and installation space.

Unidirectional Ring

The unidirectional ring is not defined in the Ethernet standard. But it can easily be implemented with standard network equipments. Every client needs only two external ports (1 × Rx and 1 × Tx). The devices are connected in a way that the Tx-port of a predecessor client is connected to the Rx-port of a successor client building a ring (Figure 3.4). This implies that communication is only possible in one direction. As shown in the right picture of Figure 3.5, 3-port switches are applied and integrated in the client devices. 2-wire twisted pair cables are used for connections. The advantages of the unidirectional ring network are the low cabling effort, the small number of ports and thus, lower cost in comparison to the double star network. The main drawback of unidirectional ring is the reduced network resilience in the case of client outage. It is quite difficult to get reliable fault rates from component manufacturers. Accordingly, it is not possible to draw exact conclusions about the resilience of future ECUs. Therefore, network resilience has not been investigated in more detail. As shown in Figure 3.4, the traffic flows have to go through all interconnected devices and are hence delayed which is another disadvantage of the

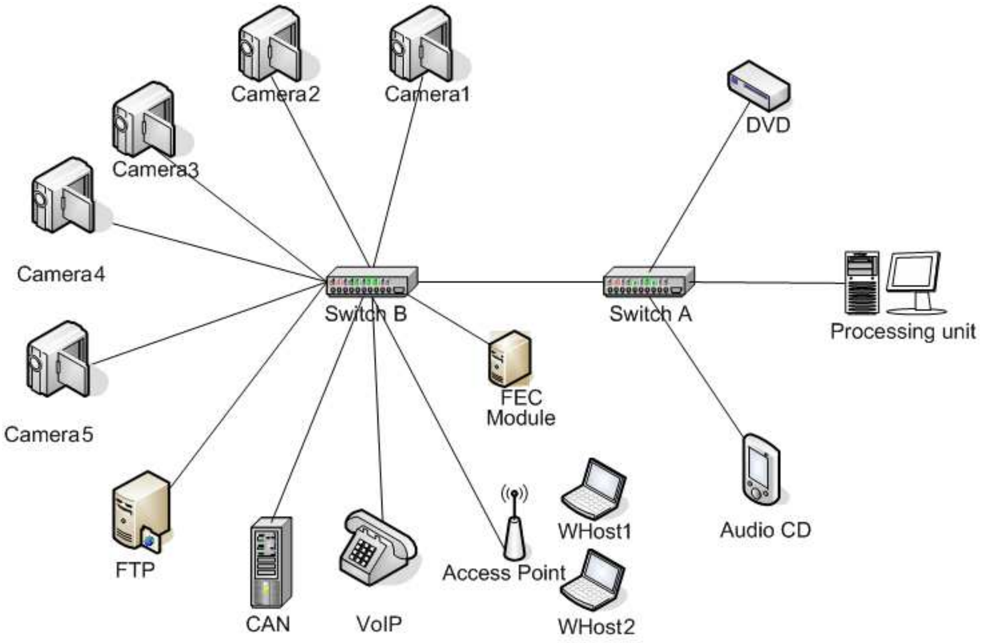


Figure 3.3: The analyzed double star topology. K- and PT-CAN applications run on one server. The FEC module supports wireless transmissions to the wireless hosts and is explained in Section 3.3.

unidirectional Ethernet ring.
The token ring technology has not been considered here mainly due to its high hardware cost and low bandwidth scalability compared to Ethernet.

3.1.2 Analysis of the Component Effort

In this section, the required production cost is described for the introduced IP/Ethernet networks from Section 3.1.1 based on the number of needed network components. It is compared with the equivalently equipped network system from Figure 2.1 that represents an example for the current in-vehicle network. The production cost discussed here is accordingly the component effort in terms of the number of required network components without considering their price values. Figure 3.5 shows possible realizations of the proposed IP/Ethernet-based network topologies that can be compared with the current in-vehicle network realization from Figure 2.1. The image processing unit abbreviated as ”Proc” in Figure 3.5 (the same as in Figure 2.1) is considered as a gateway in the current in-vehicle network, because it is an ECU that receives several video streams, processes them at the application layer and sends the resulting stream over the CVBS link to the receiver device. In the proposed IP/Ethernet networks, however, the Proc functionality is assumed to be integrated in another ECU, such as in the radio (headunit) in order to reduce the hardware effort as shown in Figure 3.5.

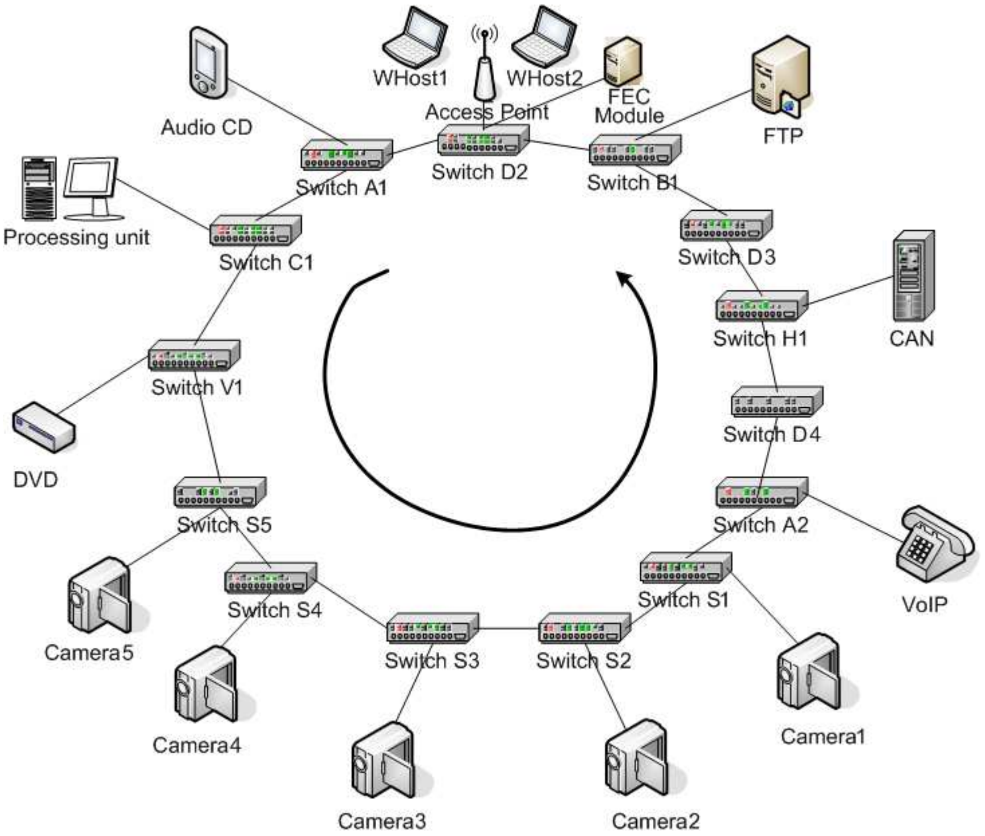


Figure 3.4: The analyzed unidirectional ring topology. K- and PT-CAN applications run on one server. The FEC module supports wireless transmissions to the wireless hosts and is explained in Section 3.3.

Comparison of the IP/Ethernet-based Double Star Network with the Current In-Vehicle Network

As mentioned in 3.1.1, the double star topology provides the highest network resilience in the case of client outage. It also offers a high flexibility for network resource planning and is the most appropriate topology from the network extensibility point of view. However, the number of required components such as cables/wires, switches, ports in the double star network is not as moderate as in the ring network. By simply comparing the number of required network components with the current in-vehicle network from Figure 2.1, the values in Table 3.1 are obtained. According to Table 3.1, the required number of network components is reduced in the proposed IP/Ethernet-based double star network which means reduction of production cost when component prices are equal in both networks.

Comparison of the IP/Ethernet-based Unidirectional Ring Network with the Current In-Vehicle Network

The unidirectional ring is more resource-efficient than the double star network because of the possibility to integrate the required 3-port switches into the end-systems. Thus, installation costs are omitted. Also, the number of required ports is reduced with device internal switches. The electrical full-duplex cables consist of only 2 wires ((T_x^+, T_x^-) seen from the sender, (R_x^+, R_x^-) seen from the receiver) in the ring network as shown in Figure 3.5. By comparing the number of

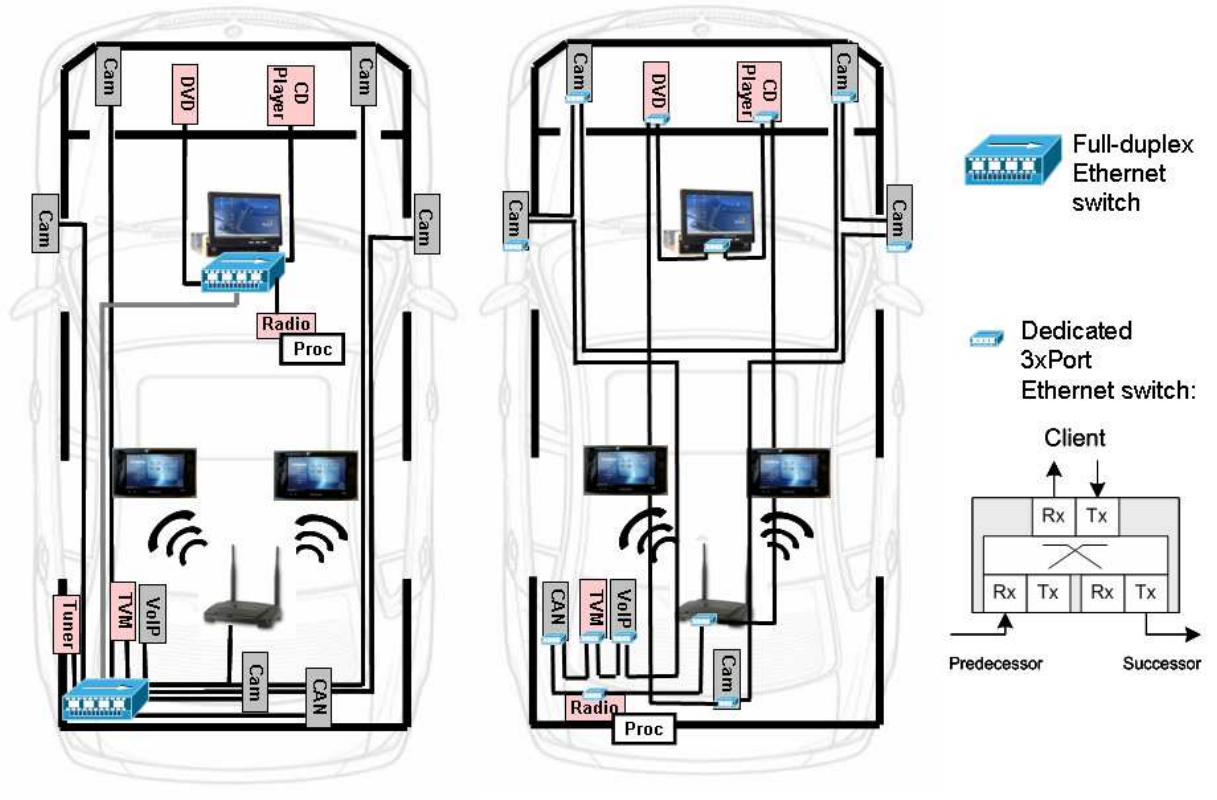


Figure 3.5: The proposed IP/Ethernet-based double star (left side) and Ring (right side) networks with equal equipments as the current in-vehicle network from Figure 2.1. Radio is another description of the headunit.

required network components in the proposed IP/Ethernet-based unidirectional ring network and in the current in-vehicle network (Figure 2.1) with equal equipments, the results in Table 3.2[1] are achieved. Table 3.2 shows a significant reduction of network components in the proposed IP/Ethernet-based unidirectional ring network. Assuming that component prices are equal in both networks, the production cost is thus reduced.

It can be concluded that both proposed IP/Ethernet-based network topologies reduce the production cost in terms of network components. The unidirectional ring topology requires a lower number of network components than the double star network due to integrated switches. The number of applied network technologies is significantly reduced from five, i.e., CAN, LIN, CVBS, LVDS and MOST in Figure 2.1 to two, Ethernet and WLAN in Figure 3.5, which indicates complexity reduction in terms of network interconnection effort (i.e., gateway). Further refinements can be done in the comparison scheme such as the consideration of shielded and unshielded cables, transmission capacity of the Ethernet cables, cable physical layer costs, installation costs, complexity of connector elements at device ports etc. that have an influence on the production cost. But in order to keep the comparison scheme simple and independent of the component manufacturers and their cost offers, the number of required network components has been chosen as the metric to evaluate the production cost in this work.

[1] Internal switches are not considered as additional network components.

Table 3.1: The required number of components for the IP/Ethernet-based double star network (Figure 3.5, left side) and the current in-vehicle network from Figure 2.1.

Networks	**# Wires**	**# Ports**	**# Switches, Access point, Gateways**
Current in-vehicle network	**54** (5x 2 wire LIN-, 5x 2 wire CVBS-, 7x 4 wire LVDS-, 7x Opt. (MOST)-cables)	**22** (4 (Video switch ↔ device), 5 (RSE ↔ device), 7 (Proc ↔ device), 6 (Device ↔ MOST))	**3** (1x Video switch, 1x RSE, 1x Proc)
Proposed IP/Ethernet double star network	**30** (15 x 2-wire cables (opt.)) or **60** (15 x 4-wire cables (elect.))	**18** (14 (switch ↔ device), 4 (switch ↔ switch, AP))	**3** (2 Full-duplex Ethernet switches, 1 Access Point)

Table 3.2: The required number of components for the IP/Ethernet-based ring network (Figure 3.5, right side) and the current in-vehicle network from Figure 2.1.

Networks	**# Wires**	**# Ports**	**# Switches, Access point, Gateways**
Current in-vehicle network	**54** (5x 2 wire LIN-, 5x 2 wire CVBS-, 7x 4 wire LVDS-, 7x Opt. (MOST)-cables)	**22** (4 (Video switch ↔ device), 5 (RSE ↔ device), 7 (Proc ↔ device), 6 (Device ↔ MOST))	**3** (1x Video switch, 1x RSE, 1x Proc)
Proposed IP/Ethernet ring network	**28** (14 x 2-wire cables (opt.)) or 28 (14 x 2-wire cables (elect.))	**14**	**1** Access Point

3.2 Wired Core Network

The switched full-duplex Fast Ethernet technology is used to realize the wired IP-based in-vehicle network, since it allows simultaneous data transmission and reception without collisions. Additionally, the electromagnetic radiation of the low-cost unshielded electrical Fast Ethernet cables is low enough for the automotive use. The concept of static IP-Diffserv [107] in combination with the IEEE 802.1p [53], [56] standard is proposed for in-vehicle communication as shown in Figure 3.1. This allows us to distinguish between the previously mentioned in-vehicle traffic classes in the car. Priority assignment is done via the QoS-API that is an extension to the standard protocol stack and works in parallel to the communication APIs (e.g., BSD socket interfaces). The QoS-API is either directly controlled by the application if it is a QoS-aware application or it is controlled by an external script for legacy applications without QoS support. The real-time control data is assigned to the priority 0, which represents the highest priority level in this work while the real-time audio and video streams are given the priority 1. Multimedia data is assigned to the priority 2 while the best effort data is given the lowest priority level, the priority 3. Layer-2 switches with the full-duplex capability are employed in the network. The switches support different traffic classes according to the IEEE 802.1p tags by parallel queueing at each output port. The strict priority scheduling mechanism is applied for the highest priority queue, i.e., the queue 0 while the queues 1 and 2 support the weighted fair queuing mechanism with weights $W1$, $W2$ and the queue 3 is served as best effort. This implies that as long as there are packets waiting to be sent in queue 0, other queues have to wait. When queue 0 is empty, then the other queues are served according to their predefined weights. The weights guarantee that the remaining queues are served with their minimum service rates and not starved by the higher priority queues [68].

The previously introduced double star and ring networks are investigated for a worst case transmission scenario when all ECUs from Figures 3.3 and 3.4 send data to the Processing Unit, representing the headunit in the car, at the same time. All five cameras are assumed to send video frames to the network simultaneously. They represent two top-view, two side-view and one rear-view cameras to be processed by one image processing unit integrated in the headunit [10]. Even though this transmission scenario represents a rare situation in the car, it is important to be considered if the network is expected to work flawlessly at any time.

In oder to guarantee QoS, the in-vehicle network should be properly dimensioned besides the introduced priority assignments. Thus, sporadic delay times due to, for example, head-of-line blocking at switch output ports, can be taken into account and the required transmission resources can be provided to avoid packet loss and quality degradation. Since all ECUs and their transmission scenarios are known before the network startup, it is possible to guarantee QoS by an accurate resource planning before data transmission is initiated. The analytical and simulation models are described in the following.

3.2.1 Analytical Model

The service curve framework introduced in [31] and [83] is the basis of the well-known Network Calculus technique [137]. As discussed in [122], Network Calculus is a widely accepted analytical technique for evaluating the performance of communication networks. The analytical model employed in this work uses the Network Calculus theory for resource planning and performance evaluation in the car. For ease of computation, the fluid model from [83] has been used as an approximation of the real system. In this section, the in-vehicle network is investigated for its QoS performance and resource usage.

Double Star Network

The **CAN applications** mentioned in Section 2.4.1 are CBR applications which are served with the highest priority. As long as their bit rates do not exceed the link capacity, they only need a small queue size just enough to store 1 or 2 packets in the case of jitter due to head-of-line blocking as shown in Figure 3.6. Thus, in order to compute the amount of required buffer, jitter should be defined first. From [83], the jitter introduced by a network element (NE) to the k_{th} packet can be calculated by

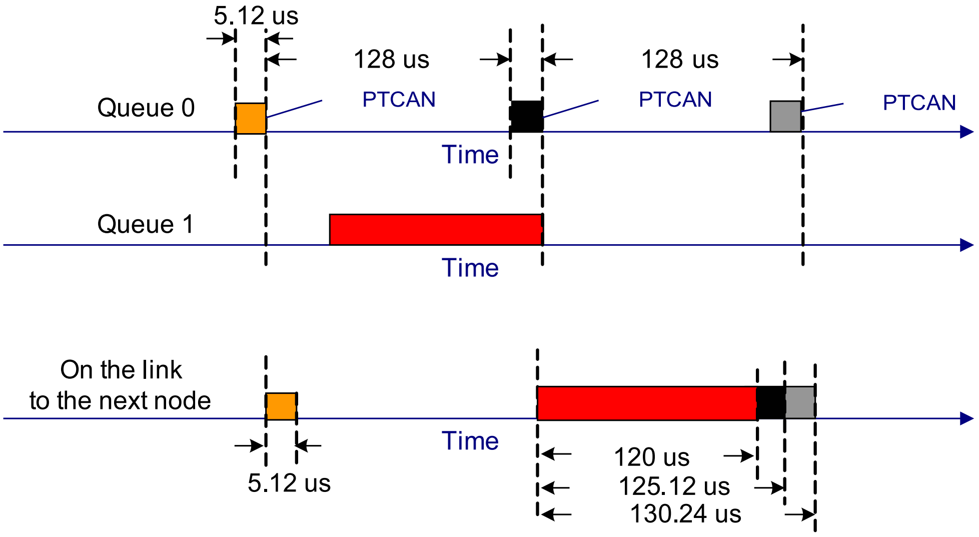


Figure 3.6: Head-of-line blocking for PT-CAN packets in queue 0. In the worst case, the blocking packet from the lower priority queue (here queue 1) is MTU-sized. The MTU size is defined here to be 1500 bytes. The packet order on the switch output link is indicated in the last line that shows the transmission toward the next node.

$$\text{Jitter}_{NE,k} = |(f_k - f_{k-1}) - (a_k - a_{k-1})| \tag{3.1}$$

where the k_{th} packet arrives at this NE at time a_k and departs at time f_k. According to Eq. (3.1), the jitter introduced to the last PT-CAN packet from Figure 3.6 is the largest and is equal to 122.88 μs per switch (See Appendix A.2 for details.). The queue size needed in Switch B (Figure 3.3), as a result, must be large enough to handle this jitter, i.e., to store the number of packets that fit into the 122.88 μs interval. Consequently, the maximum jitter in Switch A is twice as large and equal to 245.76 μs. Based on similar arguments, the maximum jitter for K-CAN packets is computed to be 120 μs per switch. Since the inter-arrival times of PT- and K-CAN packets are 128 μs and 640 μs, the required queue sizes in each switch can be calculated from

$$\text{Queue0}_{NE,k} = \left\lceil \frac{\text{max.jitter}_{PT}}{128\,\mu\text{s}} \right\rceil + \left\lceil \frac{\text{max.jitter}_{K}}{640\,\mu\text{s}} \right\rceil . \tag{3.2}$$

After substituting all values, the queue sizes are computed to be 2 and 3 MTU[2]-sized packets for switches B and A in Figure 3.3, respectively.

The **driver assistance cameras** emit packets composing picture frames in a burst with the instantaneous transmission rate of nearly 100 Mbit/s for a short period of time. According to the transmission scenario mentioned before, 5 cameras send data simultaneously to Switch B. The combined arrival traffic curve seen from Switch B is therefore quite high as shown in Figure 3.7. The I-frame obtained from the F-ARIMA traffic model has an average size of 54904 bytes (see Table 2.5) which takes 4.272 ms to be transmitted. By adding the arrival traffic curves of all 5 cameras, the instantaneous transmission rate at the beginning of each frame turns out to be 500 Mbit/s. The combined arrival traffic curve starts from 7500 bytes at time 0 (Figure 3.7), implying that there are 5 packets from 5 cameras ready to be sent simultaneously. The service rate R_1 for this traffic class is calculated by

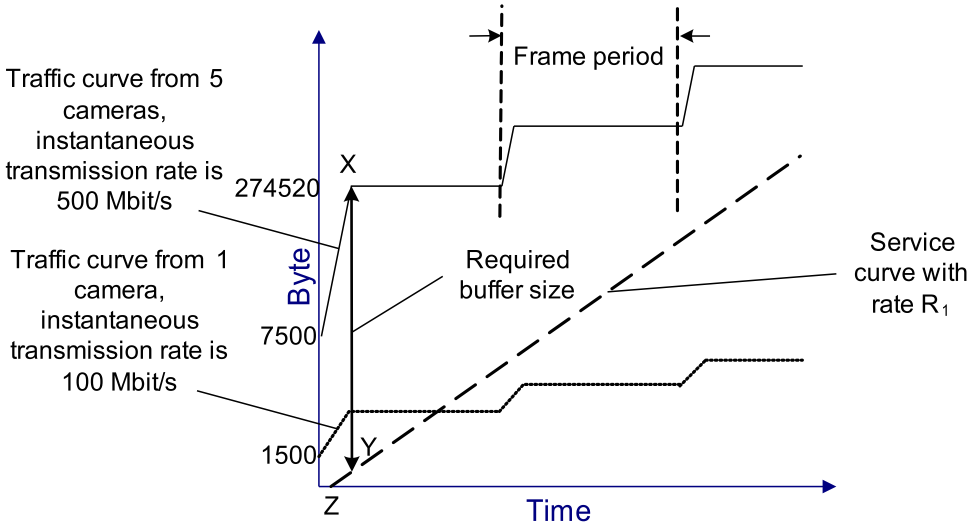


Figure 3.7: Traffic curve of driver assistance cameras in Switch B and the related service curve in the double star network.

$$R_1 = (100\,\text{Mbit/s} - R_0) \cdot \left(\frac{W_1}{W_1 + W_2} \right) \tag{3.3}$$

where R_0 is the service rate of the CAN applications and is equal to 4.8 Mbit/s. The weights W_1 and W_2 are computed according to

$$\left| \frac{W_1}{W_1 + W_2} - \frac{R_1}{R_1 + R_2} \right| \leq \varepsilon \cdot \frac{R_1}{R_1 + R_2} \tag{3.4}$$

to be 9 and 1 in the double star network with the precision factor ε being set to a small value, i.e., 0.1, resulting in R_1 = 85.68 Mbit/s. Since the weights are computed per priority queue, R_1 in Eq. (3.4) is defined by the service rate of one camera times 5, because of five simultaneous camera streams entering queue 1. Among all achieved (W_1, W_2) combinations, the one that gives

[2] The Maximum Transfer Unit (MTU) is defined to be 1500 bytes in this work.

the best trade-off between the packetization effect and the precision has been chosen[3].
The required queue size in Switch B for all 5 cameras is according to [31] the largest vertical distance between the arrival and service curves, i.e., the XY distance in Figure 3.7. Z represents the maximum deviation from an ideal fluid server or more simple, an upper bound on initial delay before the switch can serve the input stream as explained in [83]. In general, for a total of i upstream switches including the considered switch, Z can be calculated by

$$Z = \sum_{j=1}^{i} Z_j = \sum_{j=1}^{i} \left(\frac{C_j}{R} + D_j \right) \tag{3.5}$$

where Z_j is the initial delay of each switch, C_j is the maximum packet size representing the packetization effect at the j^{th} switch. R is the service rate offered to the considered stream and D_j is the maximum time the switch j has to wait until the link is free to transmit. In the worst case, Switch B has to take into account a head-of-line blocking by a lower priority MTU-sized packet and an additional delay due to interfering CAN packets. Considering the packet inter-arrival times of the CAN applications, at most 3 CAN packets can interfere in the mean time. Hence, D_1 in Switch B can be written as

$$D_1 = \frac{1500 \cdot 8\,\text{bit} + 3 \cdot 64 \cdot 8\,\text{bit}}{100\,\text{Mbit/s}} = 135.36\,\mu\text{s}. \tag{3.6}$$

By substituting D_1 in Eq. (3.5), Z turns out to be 0.275 ms. Since Z is an upper bound on the initial delay that indicates the largest delay before packet transmission, the resulting service curve is just a lower bound implying how much data should have at least been transmitted, at any time. The required queue size, Q_1 in Switch B can now be calculated as $\left\lceil \frac{X-Y}{1500} \right\rceil$ which results to be 155 MTU-sized packets long. The required queue size Q_1 in Switch A to support all 5 cameras can be computed in a similar manner. For Switch A only the initial delay Z changes to be 0.55 ms. The required Q_1 size is then equal to 234657 bytes or approximately 157 MTU-sized packets.
According to Figure 3.3, the **DVD and audio CD players** are both connected to Switch A. Thus, Switch B does not need any capacity for queue 2. The service rate R_2 can be computed as follows

$$R_2 = (100\,\text{Mbit/s} - R_0) \cdot \left(\frac{W_2}{W_1 + W_2} \right). \tag{3.7}$$

to be 9.52 Mbit/s. With an average I-frame size of 57191 bytes (see Table 2.5), an initial delay of 1.395 ms, and weights W_1 and W_2 equal to 9 and 1, the queue size Q_2 in Switch A is computed to be 535496 bytes or 36 MTU-sized packets.
All computed queue sizes for the double star network are listed in Table 3.3.

Unidirectional Ring Network

All computations that have been carried out for **CAN applications** in the double star network are also valid for the unidirectional ring network, except that there are 6 switches between the CAN application server and the wired client in the ring (see Figure 3.4). By using Eq. (3.1) and Figure 3.6, the maximum jitter of 122.88 μs results also for CAN packets in the ring network. Table 3.4 shows the required buffer size for CAN applications computed by Eq. (3.2).

[3] By considering that W_3 defines the weight of the lowest priority (best effort) queue, i.e., queue 3, it is assumed that the ratio between W_2 and W_3 is always 10 and it is necessary that $W_1 \geq W_2 \geq W_3$ in this work.

Table 3.3: Queue size requirements given as the number of MTU-sized packets for both switches in the double-star network from Figure 3.3. Q stands for Queue and $W_1 : W_2$ has been set to 9:1.

Switch	Q0	Q1	Q2
A	3	157	36
B	2	155	0

For **driver assistance cameras**, the computations are a bit more complicated, since different switches receive a different number of streams. For example, the Switch S5 in Figure 3.4 only receives one camera stream. Its arrival traffic curve is relatively simple as shown in Figure 3.8. The service rate R_1 is computed by Eq. (3.3) and weights W_1 = 13 and W_2 = 1. The resulting rate R_1 is 88.4 Mbit/s. The initial delay Z is calculated through Eq. (3.5) with a D that does not include any blocking time from higher priority CAN packets. This is because there is no CAN server upstream to Switch S5. The initial delay Z per switch is accordingly 0.255 ms and the required queue size in Switch S5 is 10516 bytes, i.e., approximately 7 MTU-sized packets. The required queue size in Switch S4 can be calculated in a similar manner as for Switch A in the double star network by adding 2 arrival traffic curves from Camera4 and Camera5 and including Z_{S4} in the total initial delay Z. The largest vertical distance between the combined arrival traffic curve and the service curve of Switch S4 is the required queue size. As shown in Figure 3.9, the combined arrival traffic curve starts at 3000 bytes, because there are 2 input ports at each switch in the ring, i.e., there are maximum 2 packets competing for the output queue in the beginning of data transmission. For other camera switches down to Switch S1, the situation is a bit more complicated. For example, Switch S1 receives two input streams, one from Camera1 and one from the output port of Switch S2, which contains data streams of four upstream cameras. Hence, it

Table 3.4: Queue size requirements given as the number of MTU-sized packets for all switches in the ring network from Figure 3.4. Q stands for Queue and $W_1 : W_2$ has been set to 13:1.

Switch	Q0	Q1	Q2
V1	0	0	37
S5	0	7	38
S4	0	46	39
S3	0	52	39
S2	0	58	39
S1	0	64	39
A2	0	66	39
D4	0	68	39
H1	2	70	39
D3	3	72	39
B1	4	74	39
D2	5	76	39
A1	6	77	39
C1	8	79	39

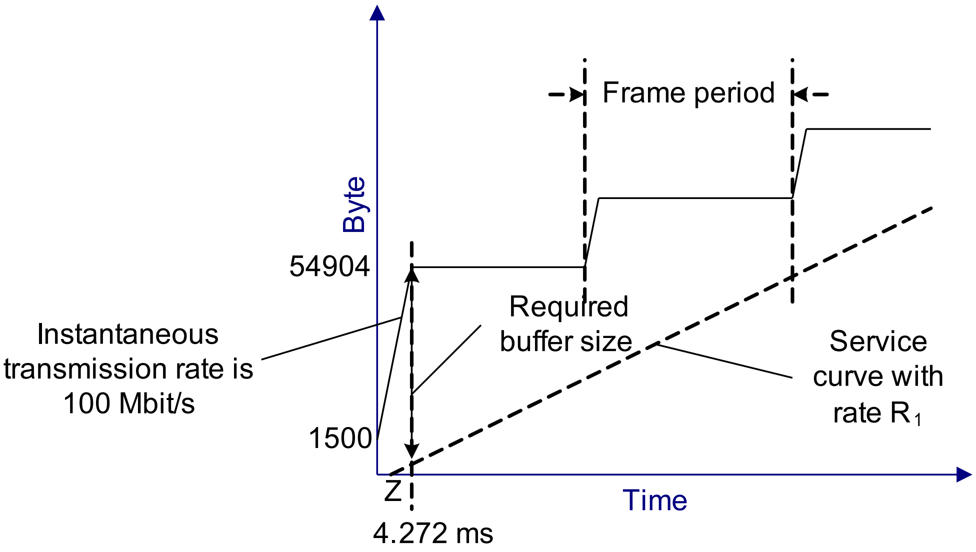

Frame period
Byte
54904
Instantaneous transmission rate is 100 Mbit/s
Required buffer size
Service curve with rate R1
1500
Z
4.272 ms
Time

Figure 3.8: Traffic curve of driver assistance cameras in Switch S5 and its service curve.

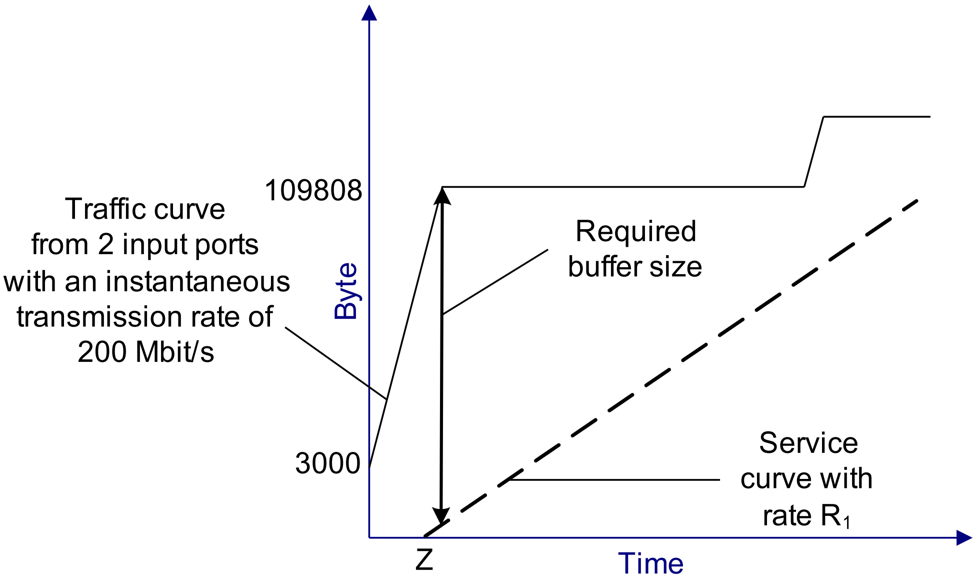

109808
Traffic curve from 2 input ports with an instantaneous transmission rate of 200 Mbit/s
Byte
Required buffer size
3000
Service curve with rate R1
Z
Time

Figure 3.9: Traffic curve of driver assistance cameras in Switch S4 and its service curve.

takes longer for the input port from switch S2 to send the packets from 4 picture frames than for the port connected to Camera1 that sends only its own frame. Therefore, the combined incoming instantaneous transmission rate at the beginning of each frame period has two slopes, 200 Mbit/s when both input ports are active and 100 Mbit/s when only the input port from Switch S2 is active. Figure 3.10 shows the traffic curve of Switch S1. The required queue sizes of all remaining switches down to Switch C1 can be calculated similar to Switch S1, since the combined arrival traffic curve seen by those switches is similar to Switch S1. Only the initial delay Z has to be increased with each additional switch along the transmission path to consider the growing jitter. The queue size requirement of Switch V1 connected to the **DVD player** is computed similar to the mentioned double star network, except for R_2 that is recalculated using new weights. In all other switches, the situation of queue 2 is similar to Switch V1, but the initial delay Z must be

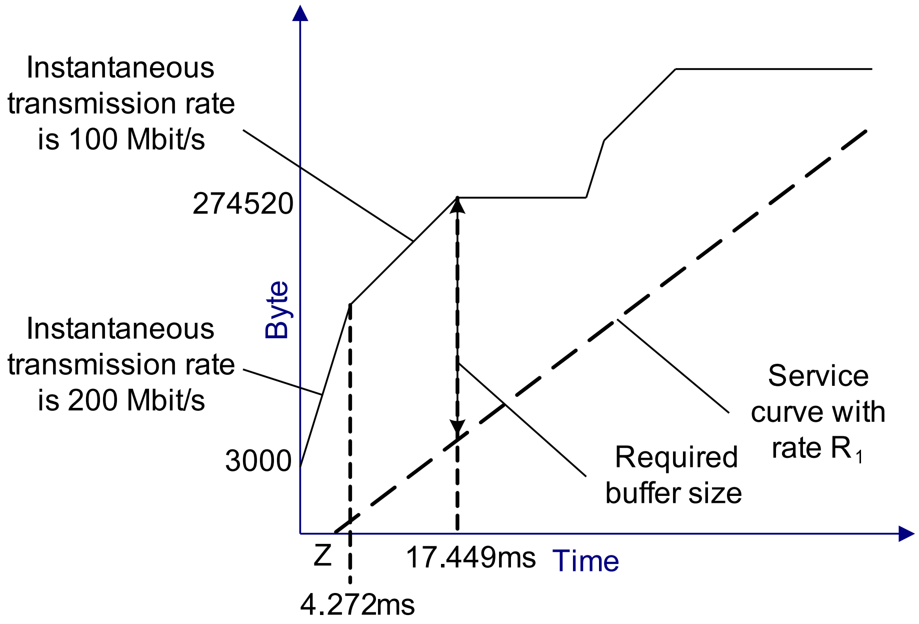


Figure 3.10: Traffic curve of driver assistance cameras in Switch S1 and its service curve.

increased for each additional switch along the transmission path to include the growing jitter. However, in Switch S4, the initial delay Z exceeds the transmission time of an I-frame. The DVD traffic curve is shown in Figure 3.11. Even though Z increases towards the downstream switches, the required queue size does not grow as long as Z remains within a frame period.

All required queue sizes in the ring are listed in Table 3.4. Generally speaking, switches that

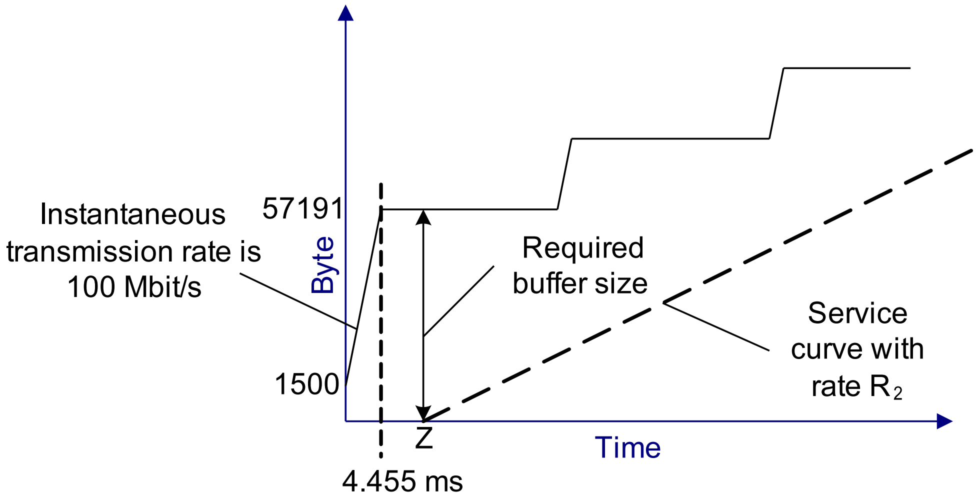


Figure 3.11: Traffic and service curves in all switches after Switch S3 for the DVD player.

are closer to the end of the transmission path require larger queues, because they suffer from a larger jitter.

3.2.2 Simulation Model

Network performance analysis by simulations is a widely used method in the research community. Simulation as an evaluation method is well understood and has been discussed extensively in the literature [122], [103]. While analytical evaluations deliver fast but not always realistic results due to several simplifications and assumptions, simulations offer more realistic results for larger and more complex networks. In this work, the INET framework from the OMNeT++ (Objective Modular Network Testbed in C++) network simulation tool [96] has been used to simulate the in-vehicle communication network. OMNeT++ is a discrete event simulator based on C++. It is highly modular, very well structured, scalable, and free of charge for the research community. The INET framework provides all components of a communication network such as switches, cables, servers, hosts, etc.

For each simulation 10 iterations have been performed with a duration of 600 s each. Full-duplex Ethernet cables with 100 Mbit/s and a negligible propagation delay have been used to interconnect devices in the network. Full-duplex Ethernet switches without input queues, but with QoS-aware output queues (4 priority queues per output port) have been used. The relay unit in the switches forwards received packets to the destination output ports. It contains 2 CPUs that work in parallel with a processing time of 3 μs[4] and an address table size of 100 entries. All ECUs from Figures 3.3 and 3.4 transmit via UDP except for the best effort FTP ECU that uses TCP to transfer a 250 Mbyte file. The applications are configured according to the parameters from Section 2.4.1.

In this work, network simulation is used to evaluate the pessimistic analytical results. As will be discussed in Section 4.2.3, simulation models are configured according to the analytical results, e.g., with analytically computed queue sizes and carried out to exploit the QoS performance such as late and lost packet rates. Simulation results are presented and explained in Section 4.2.3.

The analytically computed network resources in terms of service rate and buffer size (Tables 3.3 and 3.4) indicate a large resource consumption to achieve the required QoS, especially for VBR video applications. The reason is the rate variability of video applications that results in large traffic bursts. For example, while the CBR CAN application requires a buffer size of only 3 MTU-sized packets (roughly 4.5 kbytes), the VBR camera application requires a queue size of 157 MTU-sized packets (roughly 235.5 kbytes) in the double star network (Tables 3.3). This entails high costs and is not satisfactory for the automotive sector where a large number of samples is produced for a model range of cars. Accordingly, the data rate variability should be bounded to obtain a lower resource usage.

3.3 Wireless Peripheral Network

As shown in Figure 3.1, the wired core network in the car can be extended by an IP-based wireless peripheral network to improve flexibility, and reduce cable harness and the risk of cable break. Thus, the two rear seat entertainment (RSE) sinks can, for example, be connected via a wireless network to the IP/Ethernet-based in-vehicle network as shown in Figure 3.5. RSE sinks

[4] The processing time of 3 μs has been taken from some switch manufacturer data sheets.

are provided by car manufacturers. The required QoS is therefore guaranteed. The connection of further wireless hosts such as consumer electronic (CE) devices to the in-vehicle IP/Ethernet-based network is also possible. However, in this work, we assume that providing QoS for CE devices is less important than for the RSE sinks from car manufacturers' point of view. Accordingly, in the first step, the two RSE sinks are considered for wireless communication analysis in this work.

The IEEE 802.11g standard has been used for wireless communication in the car due to its wide availability in consumer and professional electronic segments, comparably high transmission rate and low cost. However, for future use, other wide band wireless technologies such as Ultra Wide Band and IEEE 802.11n can also be taken into consideration. In order to provide QoS for the packets sent from the wired to the wireless network, priority queues are introduced to the output port of the access point from Figure 3.5. The above mentioned four priority queues in the switches are also implemented in the access point and extended with an additional queue with the highest priority and strict priority scheduling for the WLAN management data, i.e., beacons, RTS/CTS messages etc. The output port queues in the access point are configured as follows: WLAN management data ⟶ queue 0; K-/PT-CAN data ⟶ queue 1; Camera, VoIP ⟶ queue 2; DVD, Audio CD ⟶ queue 3; best effort data ⟶ queue 4. As for switches, queue 0 is assigned with the highest priority level, priority 0 while queue 1 is assigned with a lower priority level, priority 1 and so on.

The worst case transmission scenario applied to the wired network (Section 3.2) is adapted for the wireless network. The wireless hosts, WHost1 and WHost2 from Figures 3.3 and 3.4 representing the RSE sinks receive data from the Camera 5, K- and PT-CAN applications, Audio CD and the DVD player with configurations from Table 3.5. The FTP/TCP-based best effort data is only sent to WHost1. In this transmission scenario, most of the data streams are sent to more than one receiver. This indicates the need for multicast transmission in order to reduce the required transmission resources. Our analysis has shown that multicast transmission over the lossy wireless medium does not perform as well as the unicast transmission due to the lack of an ARQ mechanism in layer-2.

Table 3.5: Traffic Sources sending data to the wireless hosts with packet sizes at the application layer and data rates at network layer.

Source	Data Rate [Mbit/s]	Packet Size [Byte]	Priority	max. allowed delay [ms]
PT-CAN	4.0	8	1	10
K-CAN	0.8	8	1	
Camera 5	≈ 7.0	1472	2	33
DVD	≈ 4.7	1472	3	200
Audio CD	1.4	1472	3	100
Best Effort	-	-	4	-

Channel measurements in the car with no passengers and two wireless receivers placed in the front and back of the car have shown that more than 90% of all losses are single packet losses, about 5% consist of two consecutive packets and the rest defines larger bursts. Independent of packet size and data rate, a maximum bit error rate of 2% has been observed in the car [128].

Several error recovery schemes based on Forward Error Correction (FEC) codes have been proposed in the literature to support multicast transmissions over lossy wireless channels. [73] proposes an adaptive FEC algorithm that outperforms other FEC-based mechanisms in terms of error recovery, overhead load and end-to-end delay. The proposed adaptive FEC module polls the wireless hosts periodically with multicast loss queries. The hosts answer with loss reply messages, which contain the information about packet loss rates. Thus, the FEC module is implicitly informed about the wireless channel status without measuring the Signal to Noise Ratio (SNR) etc. The hosts compute packet loss rates by using the sequence numbers of the inbound packets. The FEC module disposes of several FEC levels, each defining a certain number of FEC packets. It sets the sojourn time in each FEC level based on the success rate of prior transmissions. Control messages are sent from the FEC module to inform the wireless hosts about the current FEC level before the FEC data is sent via multicast to the receivers. It has been shown in [73] that the adaptive FEC algorithm performs better than static FEC mechanisms by adapting the amount of FEC data and reducing the overhead in less noisy channels. [73] also indicates the superiority of the adaptive FEC over hybrid ARQ mechanisms in channels with high bit error rates and several wireless hosts in terms of error correction and produced overhead. Thus, the adaptive FEC mechanism has been selected to protect multicast transmissions in the car enhanced with the priority assignment mechanism. The FEC module assigns priority levels to the FEC packets before transmission according to their related data packets. Thus, FEC packets are queued in the same output queue in the access point as their related data.

The IEEE 802.11g standard has a nominal throughput of 54 Mbit/s which cannot be achieved in a real system due to protocol overhead and channel access limitations. The overhead is the result of headers that are added to the packets in the access point. WLAN is a shared medium. Consequently, each host has to check whether the channel is free before transmitting data in order not to interfere with other data flows. A significant parameter that influences the throughput over wireless channels is the packet size. Smaller packets access the channel more frequently and produce a higher overhead than larger packets for the same transmission rate. Hence, they limit the throughput. Thus, the smaller the packet is, the lower is the achieved throughput. An interesting solution to this issue is the frame bursting mechanism proposed in the IEEE 802.11e standard which provides a dedicated transmission time, called transmission opportunity (TXOP), to each sender. During TXOP the sender is allowed to transmit as many frames as the time slot allows without competing for the channel access. By using the formulas from [132] and [111] and the parameters from Table 3.6, throughput values have been computed analytically for different packet sizes with and without frame bursting. Also, simulations have been conducted that confirm the analytical results as shown in Figure 3.12. According to the IEEE 802.11e standard [140], the TXOP limit, i.e., the slot duration is defined by an 8 bit data field in each data frame. It can therefore have 255 different values. The duration can vary from 32 μs up to 8160 μs in 32 μs steps. A large TXOP limit evolves an unfairness to other wireless hosts which are not capable of frame bursting. In the above mentioned transmission scenario, all wireless hosts are

Table 3.6: IEEE 802.11g parameters.

SlotTime	9 μs
DIFSTime	28 μs
SIFSTime	10 μs
CW$_{min}$	31 μs
tPLCPHeader	4 μs
tPLCPPreamble	16 μs
tSymbol	4 μs

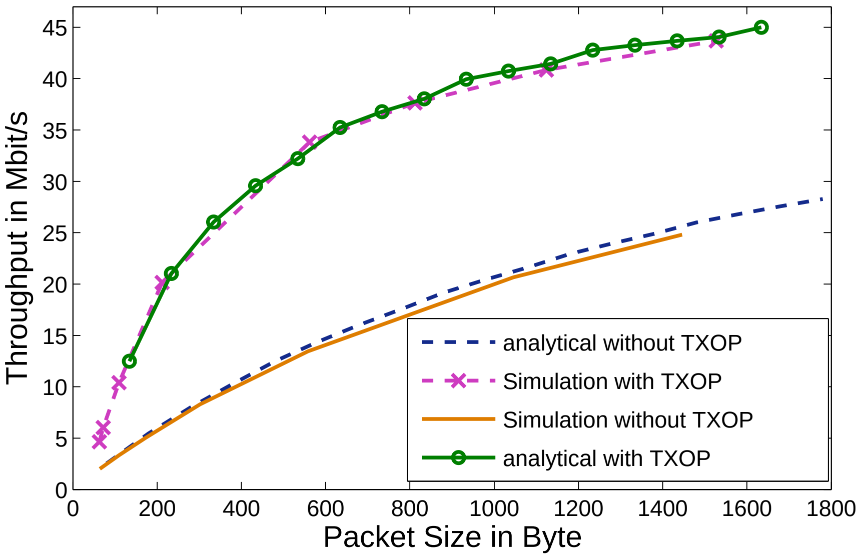


Figure 3.12: IEEE 802.11g throughput values computed analytically and via simulations. TXOP indicates the application of frame bursting.

capable of frame bursting and there are only very few data packets such as acknowledgments that are transmitted from the wireless hosts to the wired network. Therefore, the unfairness can be neglected and for the best throughput performance the TXOP limit is set to its maximum values of 8160 μs. As shown in Figure 3.12, this TXOP value is high enough to accommodate larger packets and improve their throughput performance as well.

As for the wired network, it is also possible to guarantee QoS by an accurate resource planning in the wireless network before data transmission is initiated. The analytical and simulation models are described in the following.

3.3.1 Analytical Model

In order to compute the required service rates in the IEEE 802.11g network, the service rates computed for the IP/Ethernet-based in-vehicle network in Section 3.2 are scaled according to the WLAN throughput values from Figure 3.12 as

$$\frac{R_{100\,Mbit/s}}{\text{Fast Ethernet Throughput} - R_{CAN}} = \frac{R_{54\,Mbit/s}}{\text{WLAN Throughput}}. \tag{3.8}$$

$R_{100\,Mbit/s}$ represents the service rate in the wired Fast Ethernet network (Section 3.2) with 100 Mbit/s throughput while R_{CAN} is the required data rate for K- and PT-CAN packets of 4.8 Mbit/s (Section 2.4.2). $R_{54\,Mbit/s}$ defines the required service rate in the IEEE 802.11g network[5] which depends on the throughput values from Figure 3.12 limited by the packet size. The initial delay Z can be computed according to Eq. (3.5) where C_j is the packetization delay, D_j the head-of-line blocking delay at the j^{th} network element, and R is set to $R_{54\,Mbit/s}$ for the IEEE

[5] The nominal bit rate of the IEEE 802.11g channel is 54 Mbit/s.

802.11g channel. The buffer size in the access point can consequently be calculated according to [83]. Thus, resource requirements can be computed for the IEEE 802.11g access point and channel.

3.3.2 Simulation Model

The simulation environment described in 3.2.2 has been extended for wireless communication with an access point, two wireless hosts and an FEC module. The IEEE 802.11g access point of the INET framework has been enhanced with 5 priority output queues as explained before. The proposed adaptive FEC mechanism is placed in a dedicated FEC module that is connected to the same switch as the access point (see Figures 3.3 and 3.4). Thus, the transmission delay of FEC packets to the wireless receivers is reduced. The in-vehicle sources remain unmodified and are preserved from additional computational complexity. The FEC module is a member in all multicast groups that require redundant information for data reconstruction. Consequently, all multicast data sent to the wireless hosts is also forwarded to the FEC module, which produces FEC information and sends it to the wireless members of that multicast group. The data itself is dumped in the FEC module as soon as the FEC data is transmitted. The INET UDP application is extended with the FEC functionality in the FEC module and in the wireless hosts according to [64]. The frequency of loss queries and control messages of the adaptive FEC mechanism is set to 2 s and 2.1 s, respectively, to achieve a good error recovery performance. The wireless hosts WHost1 and WHost2 have both a distance of 5 m to the access point.
In order to model packet loss in the IEEE 802.11g channel a simplified Gilbert Elliot model, as shown in Figure 3.13, has been applied. It has basically two parameters *meanGood* and *meanBad*. These parameters correspond to the average time spent in each state before the transition to the other state or in other words, to the mean state sojourn times. The timers are exponentially distributed. In the bad state, all packets are marked as corrupted so that upon arrival at the receivers, they are discarded and considered as lost. The main difference compared to the original Gilbert Elliot model is that the transition probabilities are fixed in the sense that the probability to stay in a state after the sojourn time is elapsed is equal to zero. According to the channel measurement results from [128], *meanGood* is set to 1 while *meanBad* is set to 0.02 for the simulations in order to model the maximum bit error rate of 2% in the car.
Simulation results are presented in Section 4.2.3 for analytically configured networks. Analytically computed resources are also compared to those defined by simulation.

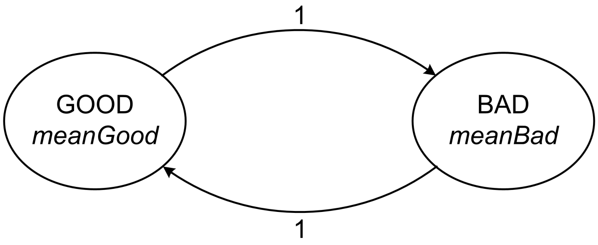


Figure 3.13: Simplified Gilbert Elliot Model

The required network resources for wireless communication in the proposed transmission scenario exceed the WLAN channel capacity due to the large bursts from the wired network. Consequently, the wireless channel is saturated after a short time and packets are continuously buffered in the access point output queues without being transmitted. The wireless peripheral network can thus not function without limiting the incoming traffic bursts.

3.4 Summary

In this chapter, an IP-based network architecture is introduced for the overall in-vehicle communication to cope with growing complexity and comply with future automotive demands, especially in driver assistance and multimedia application domains. Due to the car manufacturers' requirement highly safety critical control messages are not covered by the proposed IP-based network and are assumed to be transmitted via a separate network, i.e., FlexRay. Appropriate network topologies are discussed. The introduced network architecture is a heterogeneous network system implying that it consists of a wired core network and a peripheral wireless network. While Fast Ethernet technology is used to realize the wired network, IEEE 802.11g is applied for wireless communication in the car. Multicast and unicast transmissions have been analyzed. An adaptive FEC-based mechanism is proposed to support wireless multicast transmissions in the car. In order to assure QoS in both networks the concept of static priority assignment is presented. It is extended with static network dimensioning to prevent packet loss due to buffer overflow in overload situations. The Network Calculus fluid model is used to analytically compute the required network resources in a worst case transmission scenario. Network simulations have been performed to prove the analytically computed resources for the required QoS and to define more realistic network resource values. It turned out that a large amount of network resources (i.e., service rate and buffer size) is needed to achieve the required QoS. This implies a high cost for the wired Fast Ethernet network while a simultaneous transmission of all data flows is not possible over the IEEE 802.11g network due to channel saturation. The large traffic bursts from the VBR video sources are the main reason for this issue which is further analyzed in the next chapter.

4 Traffic Shaping for Resource-Efficient In-Vehicle Communication

The analysis of the previous chapter has shown a large buffer size consumption and a saturation of the wireless channel due to large video traffic bursts. As a consequence, video data transmission is costly in the wired in-vehicle network while video streaming is infeasible over the wireless channel in the car. In this chapter, the mechanism of traffic shaping is presented to attenuate traffic bursts. Thus, the required amount of resources is reduced and video streaming over the wireless channel becomes feasible in the car. Traffic shapers are analyzed in video sources, in switches and in the access point. A novel traffic shaping algorithm called Simple Traffic Smoother (STS) is presented that outperforms other traffic shapers in terms of resource usage when applied to VBR sources in the introduced double star network [7]. Additionally, a stream-based traffic reshaper implementation is investigated in the switches and access point of the proposed network topologies from Chapter 3. A detailed discussion is provided on the relationship between QoS performance and resource usage. The best QoS performance and resource usage trade-offs are defined for the analyzed in-vehicle network topologies. Analytical, simulation and experimental results are presented.

4.1 Traffic Shaping Algorithms

Traffic shaping has proven to be very useful to reduce required resources in the networks [84], [83]. Shapers are used to conform traffic streams to certain pre-defined characteristics. The principle of traffic shaping can be mathematically explained by using a fluid model to describe a flow of traffic [83]. A traffic shaper, characterized by the envelope $\bar{A}(\tau)$, is a network entity that sends each packet at time t such that $A[t, t+\tau] \leq \bar{A}(\tau)$ for $0 \leq \tau$ and $0 \leq t$. This means that the traffic shaper regulates its outgoing stream by delaying some packets until they can be sent without violating the desired traffic characteristics specified by $\bar{A}(\tau)$. In the following, appropriate traffic shaping algorithms are investigated for VBR video sources, interconnected switches and the access point. Leaky Bucket and Token Bucket belong to the most popular shapers in the literature [83], [69]. In addition, a new shaping algorithm called Simple Traffic Smoother (STS) is described and analyzed in the following.

Leaky Bucket The Leaky-Bucket shaper emits a strict CBR stream and allows no sudden bursts. Therefore, it is more suitable for CBR applications, such as VoIP. It is difficult to optimize the maximum transmission rate p of the shaper for video streams, because the difference between the frame sizes is quite large according to Section 2.4.2. The traffic curve in Figure 4.1 is assumed to represent the outgoing traffic from a VBR video application. Figure 4.2 shows the

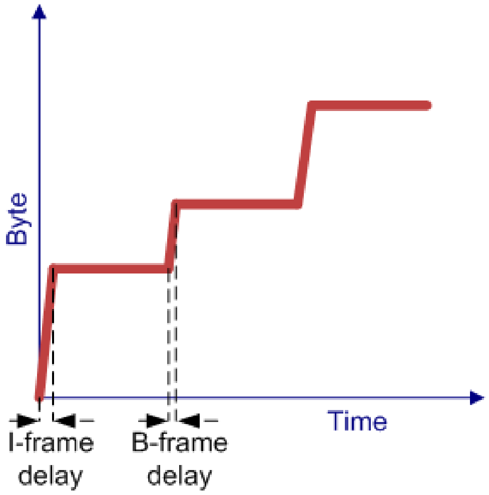


Figure 4.1: The traffic curve from a VBR video source.

Leaky Bucket shaper output with a large *p* (left curve) causing congestion in the network and a small *p* (right curve) causing a large shaper delay.

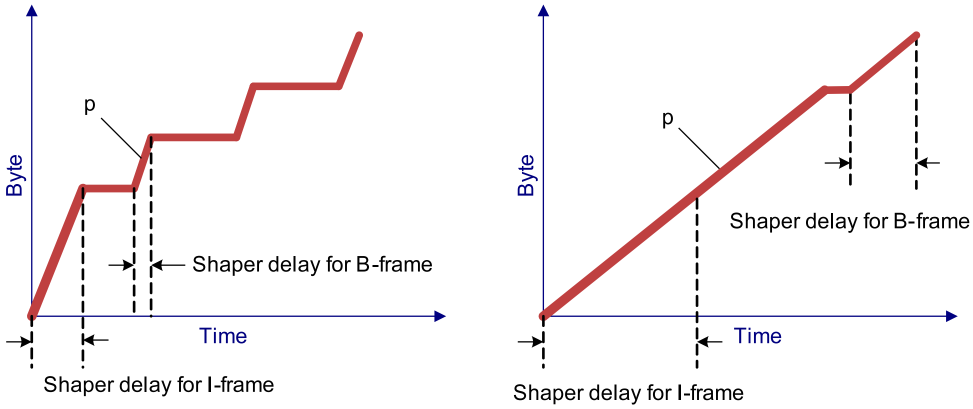


Figure 4.2: Resulting traffic curves from a Leaky Bucket video source shaper with different peak rates p.

Token Bucket The Token Bucket shaper allows a short burst during the peak transmission rate which makes it suitable for video sources. If the peak rate p and token regeneration rate r are well selected, the shaper delay from large frames will not be too high while the overall average transmission rate is low enough not to congest the network. Figure 4.3 shows the output traffic curve from a Token Bucket shaper for the input traffic from Figure 4.1.

Simple Traffic Smoother The introduced Simple Traffic Smoother (STS) in this work differs in its functionality from all other traffic shapers in the literature. It works similar to an isochronous transmission system. STS reduces the peak transmission rate by sending the packets of each picture frame with a certain time distance between them instead of sending the packets

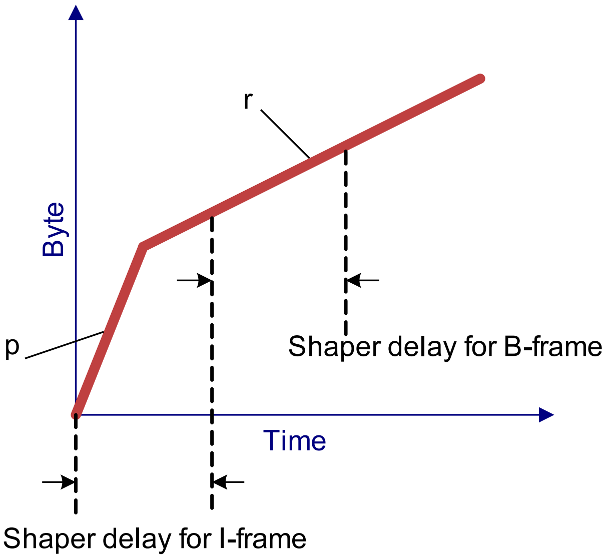


Figure 4.3: Resulting traffic curve from a Token Bucket video source shaper with the token regeneration rate r and the peak rate p.

very close to each other as a burst to the network. As shown in Figure 4.4, STS stores all packets of a picture frame that have been received within a predefined smoothing interval and sends them to the network in the next interval equally spaced in time. The delay introduced by STS

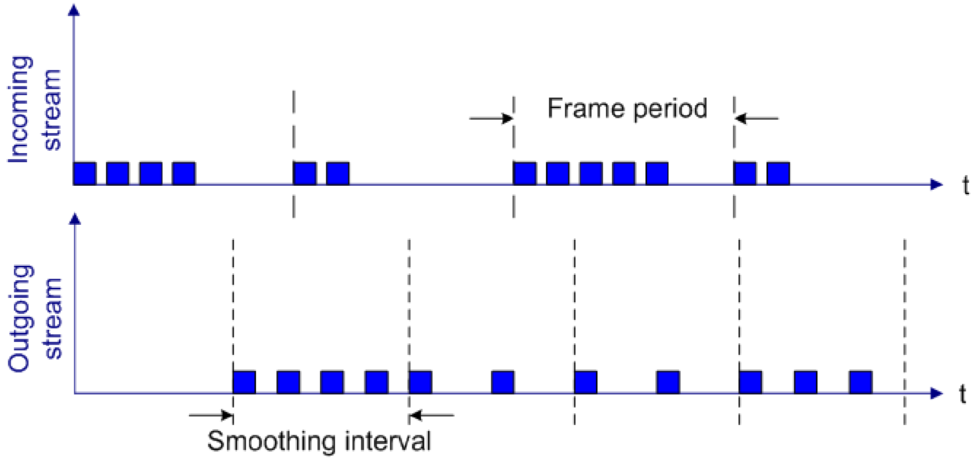


Figure 4.4: The principle of the proposed Simple Traffic Smoother (STS).

is thus limited to only two smoothing intervals. As opposed to STS, in an isochronous network, data packets are sent within specified time intervals, e.g., 125 μs time intervals in a FireWire network, such that the required transmission rate is met. This means that the same amount of bytes is sent within each time interval in order to achieve the desired constant bit rate. The STS, however, shapes the VBR streams such that the outgoing traffic has a VBR to avoid long shaper delays. Accordingly, STS is defined by the so called smoothing interval T_{SI} which regulates the traffic bursts. A larger T_{SI} provides a lower peak rate while the shaper delay and consequently, the end-to-end delay will be larger, as a trade off. Considering that T_{SI} and the frame period T can be different and do not have to be synchronized, Figure 4.5 represents a possible STS output traffic curve for the assumed input traffic from Figure 4.1. The Pseudocode of the STS algorithm

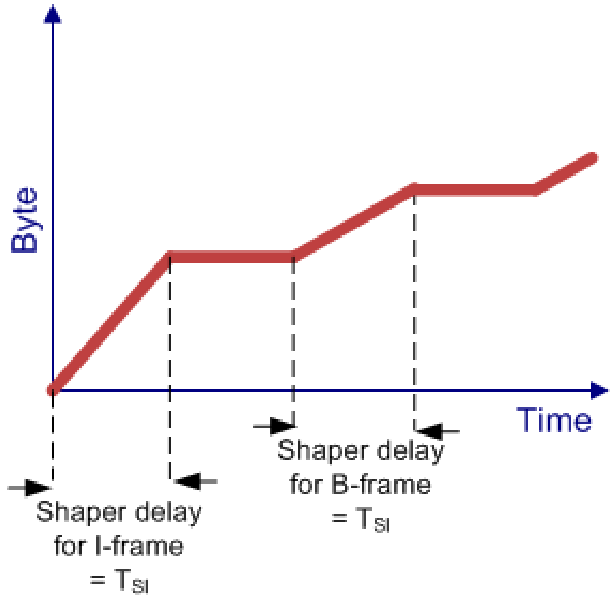


Figure 4.5: Resulting traffic curve from STS with the smoothing interval T_{SI}.

is given as follows. This code is executed at the beginning of each smoothing interval (T_{SI}).

```
vector<packet> queued_pkt
// Storing received packets during the last smoothing interval
int smooth_interval // Smoothing interval
timeval time_now = (obtain current time)
IF(there are packets to be sent in queued_pkt)
  total_pkt = (total packets stored in queued_pkt)
  pkt_interval = smooth_interval/total_pkt
  WHILE(int count = 0; count < total_pkt; count++)
  // Send packets at the beginning of the smoothing interval
    first_packet = (get first packet from queued_pkt)
    SendPacket(first_packet, time_now + (count ∗ pkt_interval))
    (remove first_packet from queued_pkt)
  End WHILE
End IF
```

4.2 Analytical and Simulation Analysis

In this section, analytical and simulation models from Chapter 3 have been enhanced with the traffic shaping functionality. Different network configurations have been investigated. The best settings have been used in a prototypical implementation that confirms the analytical and simulation results.

4.2.1 Traffic Shaping in Video Sources

In the following, relationships between shaper settings in video sources and the network resource usage are derived by applying the existing resource planning methods, e.g., from [89] and [83] to fulfill the QoS requirements, such as end-to-end delay

$$T_{e2e} = T_N + T_S \tag{4.1}$$

(where T_N defines the network delay[1] and T_S the shaper delay), throughput, lost and late packet rates. Thus, a trade-off is found between the needed resources and the provided QoS.
Traffic shapers have been implemented in the MAC layer of video sources in the simulation models from Chapter 3.

Analysis for the Token Bucket Shaper

Due to its more frequent employment for VBR data flows compared to the Leaky Bucket shaper, the Token Bucket with peak rate control and a traffic envelope $\bar{A}(\tau) = min\{p \cdot \tau, b + r \cdot \tau\}$ [108] is investigated here as video source shaper. Parameters p, b and r define the shaper peak rate, bucket size and the token regeneration rate, respectively.
By extending the computations from [89], the maximum normalized Token Bucket shaper delay turns out to be

$$T_{S,max,norm} = \begin{cases} \frac{T_{burst}}{\beta} & \text{for } \beta \geq 1 \\ T_{burst} + \frac{I_{max} \cdot (1-\beta) \cdot f}{r \cdot n} & \text{for } \beta < 1 \end{cases} \tag{4.2}$$

where I_{max} defines the largest I-frame size and GoP_{max} the largest GoP, the burst capacity $B = \frac{p \cdot b}{p-r}$, the token regeneration rate $r = \frac{GoP_{max}}{n/f}$ (with n: the number of frames in a GoP and f: the number of frames per second), the normalized burst capacity $\beta = \frac{B}{I_{max}}$ and the normalized burst period $T_{burst} = \frac{B/p}{n/f}$. Figure 4.6 shows the maximum normalized shaper delay $T_{S,max,norm}$ computed by Eq. (4.2) and via simulations with T_{burst} set to 0.1, 0.2 and 0.3. The statistical video model for the DVD application from Section 2.4.2 has been used in the simulations. The graph shows a perfect match between the analytical and simulation results. Any Token Bucket shaper that has a peak rate p less than or equal to its token regeneration rate r will never deplete its bucket, because the bucket is refilled at a greater rate than it is depleted. Thus, the Token Bucket shaper transmits data at its peak rate all the time, which represents the Leaky Bucket shaper functionality with the constant transmission rate p. A Token Bucket shaper with $T_{burst} = 1.0$ represents also a Leaky Bucket, since the burst period covers the entire GoP, i.e., the whole GoP is transmitted with the peak rate p.
Once a video stream is shaped with the Token Bucket to obtain the (p, b, r) characteristic, these parameters can be used to calculate the service rate and the required buffer size for this stream to obtain the desired QoS. Figure 4.7 shows the shaped arrival traffic curve from a video source and the network service curve. According to Figure 4.7, the actual bucket size b of the source shaper changes to

$$b_{net} = b + \frac{r \cdot M}{p} \tag{4.3}$$

[1] Network delay refers to the delay a packet experiences from the time it enters the network until the time it arrives at the receiver and leaves the network.

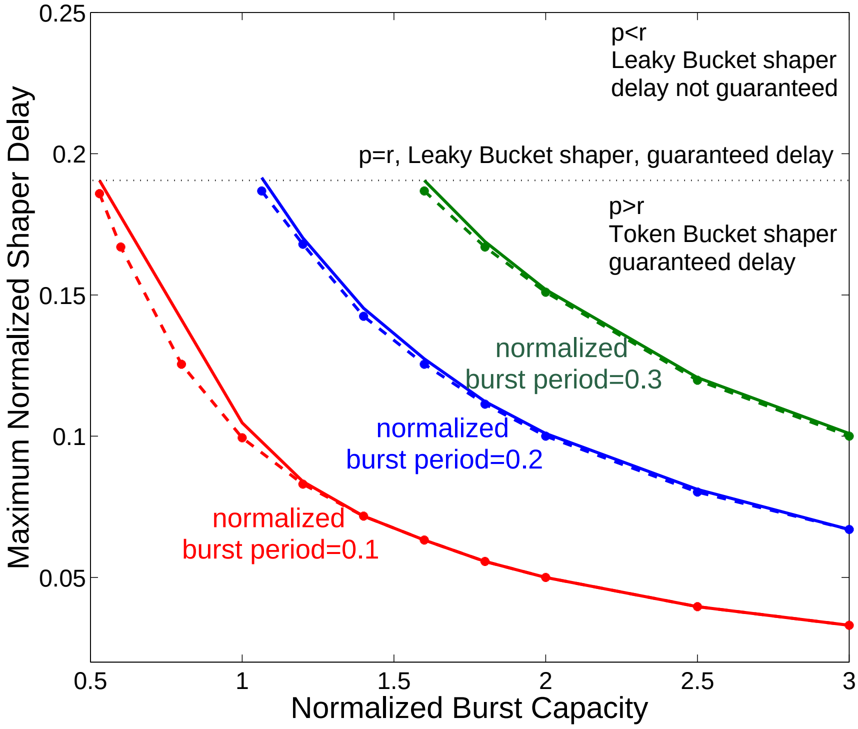


Figure 4.6: Maximum normalized shaper delay $T_{S,max,norm}$ computed analytically (dashed curves) and via simulations (solid curves).

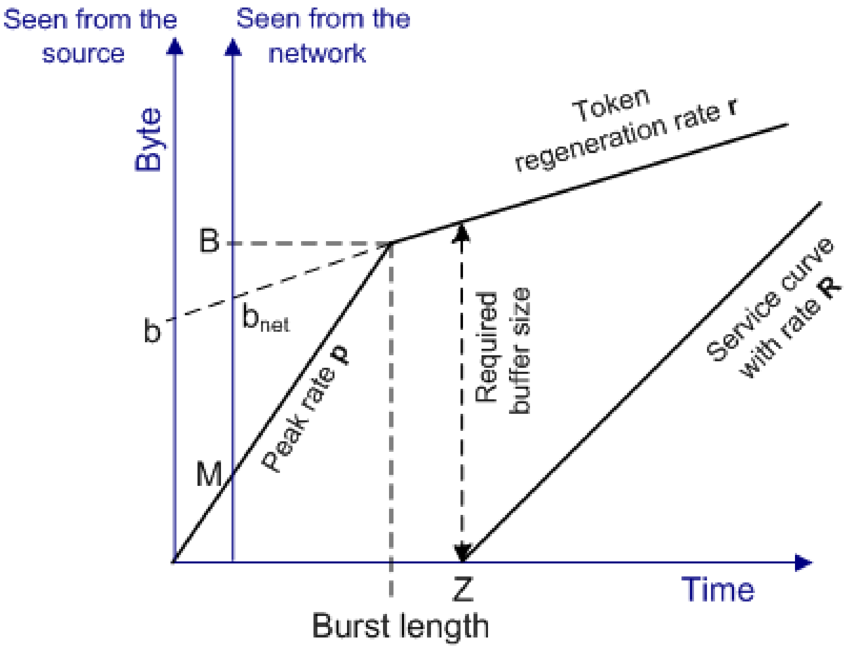


Figure 4.7: Arrival traffic curve from a Token Bucket shaped video source and the network service curve.

as seen from the network, where M denotes the size of the first packet ready to be transmitted and is commonly set to MTU for a video stream. The significance of term $\frac{r \cdot M}{p}$ in Eq. (4.3) depends on the values of r and p. It can be maximum M large when $r = p$ for a Leaky Bucket shaper while it becomes negligible for a very small r and a very large p.
According to [83], the maximum horizontal distance between the arrival curve and the service curve defines the maximum network delay $T_{N,max}$ as

$$T_{N,max} = \begin{cases} \frac{(b_{net}-M)(p-R)}{R(p-r)} + \frac{M}{R} + Z \text{ for } p \geq R \\ \\ \frac{M}{R} + Z \text{ for } p < R \end{cases} \tag{4.4}$$

where R is the service rate and Z the upper bound on the initial delay (see Eq. (3.5)). Given that the shaper parameters and $T_{N,max}$ are known, the required service rate R can be computed as

$$R = \begin{cases} \frac{M+\sum_{j=1}^{i} C_j}{T_{N,max}-\sum_{j=1}^{i} D_j} \text{ for } T_{N,\infty} < T_{N,max} \leq T_{N,p} \\ \\ \frac{p(b_{net}-M)+(p-r)\left(M+\sum_{j=1}^{i} C_j\right)}{b_{net}-M+(p-r)\left(T_{N,max}-\sum_{j=1}^{i} D_j\right)} \text{ for } T_{N,p} < T_{N,max} \leq T_{N,r} \\ \\ \text{undefined} \quad \text{otherwise.} \end{cases} \tag{4.5}$$

where

$$\begin{aligned} T_{N,max} &= T_{e2e,max} - T_{S,max} \\ T_{N,\infty} &= T_{N,R\to\infty} = \sum_{\forall j} D_j \\ T_{N,p} &= T_{N,R=p} = \sum_{\forall j} \left(\frac{C_j}{p} + D_j\right) + \frac{M}{p} \\ T_{N,r} &= T_{N,R=r} = \frac{b_{net}}{r} + \sum_{\forall j} \left(\frac{C_j}{r} + D_j\right). \end{aligned} \tag{4.6}$$

When T_N gets too close to the lower bound $T_{N,\infty}$, R becomes more unstable and increases rapidly to a very high value, i.e., $R \to \infty$. Hence, the required network delay should not be very small.
Given that R is known, the upper bound on the required buffer size at the k^{th} node along the transmission path, i.e., the maximum vertical distance between the arrival traffic curve and the service curve of that NE in Figure 4.7, can be determined as

$$\text{buffer}_k = M + \left(\frac{p-\rho}{p-r}\right)(b_{net} - M) + \sum_{j=1}^{k} \left(\frac{C_j}{R} + D_j\right) \cdot \rho \tag{4.7}$$

with

$$\rho = \begin{cases} r \text{ for } \left(\frac{b_{net}-M}{p-r}\right) \leq \sum_{j=1}^{k} \left(\frac{C_j}{R} + D_j\right) \\ \\ R \text{ for } \left(\frac{b_{net}-M}{p-r}\right) > \sum_{j=1}^{k} \left(\frac{C_j}{R} + D_j\right), p > R \\ \\ p \quad \text{otherwise.} \end{cases} \tag{4.8}$$

From the triplet (p, b, r) specifying the Token Bucket shaper, r has already been defined. In the following, constraints are set for shaper parameters $T_{S,max,norm}$, p and b and consequently, for the required network resources, R and buffer$_k$ based on the normalized burst capacity β and burst

period T_{burst} of the video source. Thus, a feasible area is defined for the (β, T_{burst}) pairs that can be used to adjust the Token Bucket shaper parameters to meet all set constraints.
The first constraint is defined as

$$T_{S,max,norm} \leq min\left(\frac{T_{e2e,max}}{n/f}, \frac{I_{max} \cdot f}{n \cdot r}\right). \tag{4.9}$$

The inequality (4.9) declares that $T_{S,max,norm}$ should neither exceed the required maximum end-to-end delay $T_{e2e,max}$ nor T_{burst} when $p = r$. The limitation for the peak rate p is defined as

$$r \leq p < R_A. \tag{4.10}$$

The upper bound R_A defines the available link capacity. In our transmission scenario with Fast Ethernet links, CAN applications require 4.8 Mbit/s and leave 95.2 Mbit/s of the link capacity for all other applications (R_A). The bucket size b cannot be less than the maximum packet size, M. Otherwise, the shaper will never have enough tokens left in its bucket to send any packet of size M bytes. Consequently,

$$b \geq M. \tag{4.11}$$

The constraint for the required service rate R is that it should not be lower than the token regeneration rate r. Accordingly,

$$R \geq r. \tag{4.12}$$

For the required buffer size

$$\text{buffer}_k \leq \text{Buffer Limit} \tag{4.13}$$

should hold.

Analysis for the Simple Traffic Smoother (STS)

According to the STS definition from Section 4.1, Figure 4.8 represents an example of all possible cases for the STS traffic curve. It shows that STS does not transmit the second frame at time B but instead at time C which is the beginning of the next smoothing interval. The traffic envelope is accordingly defined as

$$\bar{A}(\tau) = \begin{cases} \sum_{i=1}^{\lfloor \tau/T \rfloor} B_i + \frac{B_{\lceil \tau/T \rceil}}{T_{SI}} \cdot \left(\tau - \left\lfloor \frac{\tau}{T_{SI}} \right\rfloor \cdot T_{SI}\right) \text{ for} \\ \left\lfloor \frac{\tau}{T_{SI}} \right\rfloor \cdot T_{SI} \geq \left\lfloor \frac{\tau}{T} \right\rfloor \cdot T > \left(\left\lfloor \frac{\tau}{T_{SI}} \right\rfloor - 1\right) \cdot T_{SI} \\ \\ \bar{A}(\tau') \text{ otherwise.} \end{cases} \tag{4.14}$$

where

$$\tau' = max_t \left\{t < \tau, \left\lfloor \frac{t}{T_{SI}} \right\rfloor \cdot T_{SI} \geq \left\lfloor \frac{t}{T} \right\rfloor \cdot T > \left(\left\lfloor \frac{t}{T_{SI}} \right\rfloor - 1\right) \cdot T_{SI}\right\}. \tag{4.15}$$

and B_i defines the amount of bytes shaped by the STS at time instance i. Thus, $B_{\lfloor \tau/T \rfloor}$ defines the size of the last frame while $B_{\lceil \tau/T \rceil}$ is the size of the current frame. The first condition of $\bar{A}(\tau)$ is to check whether τ is in the increasing part of the traffic curve or not. Given any τ, the interval $\left(\left\lfloor \frac{\tau}{T_{SI}} \right\rfloor \cdot T_{SI}\right)$ represents the beginning of the smoothing interval that contains this τ. The interval $\left(\left\lfloor \frac{\tau}{T_{SI}} - 1 \right\rfloor \cdot T_{SI}\right)$ is the beginning of the previous smoothing interval and $\left(\left\lfloor \frac{\tau}{T} \right\rfloor \cdot T\right)$

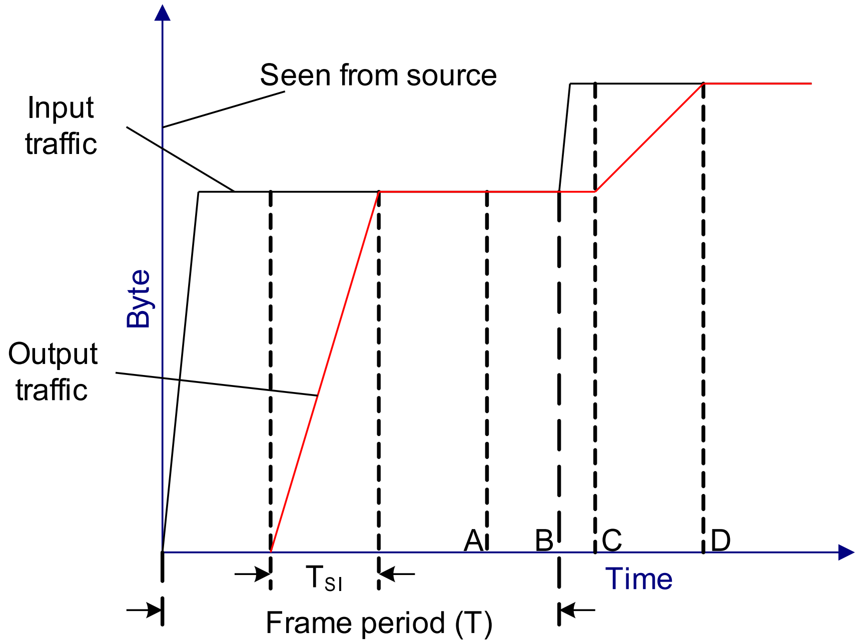


Figure 4.8: Input/output traffic curves of a video application to/from the Simple Traffic Smoother (STS).

defines the beginning of the current frame. For all other values of τ, the traffic curve is just equal to the previous $\overline{A}(\tau')$ where τ' is the maximum time t that is less than τ and satisfies the first condition. In other words, $\overline{A}(\tau')$ is the largest value from the last increasing smoothing interval. For example, in Figure 4.8, if τ is in [C, D], then $\left(\left\lfloor \frac{\tau}{T_{SI}} \right\rfloor \cdot T_{SI}\right)$ and $\left(\left\lfloor \frac{\tau}{T_{SI}} - 1 \right\rfloor \cdot T_{SI}\right)$ represent points C and A, respectively. In addition, $\left(\left\lfloor \frac{\tau}{T} \right\rfloor \cdot T\right)$ is the beginning of the second frame which is point B in Figure 4.8. This τ falls into the first condition, which implies that if the beginning of the current frame (point B) is between the beginning of the current smoothing interval (point C) and the beginning of the previous smoothing interval (point A), then this τ is in the increasing part of the traffic curve. Otherwise, the traffic curve is just equal to the previous maximum value of $\overline{A}$ that is defined in the second condition of Eq. (4.14). The minimum service rate

$$R_{min} = \frac{GoP_{max}}{n/f} \tag{4.16}$$

is defined to guarantee no accumulation of delay between the GoPs as done for the token regeneration rate r of the Token Bucket shaper with n: the number of frames in a GoP and f: the number of frames per second. The maximum STS transmission rate $r_{x,max}$ can be defined as

$$r_{x,max} = \frac{I_{max}}{T_{SI}}. \tag{4.17}$$

There are two cases for the service rate R as shown in Figure 4.9.

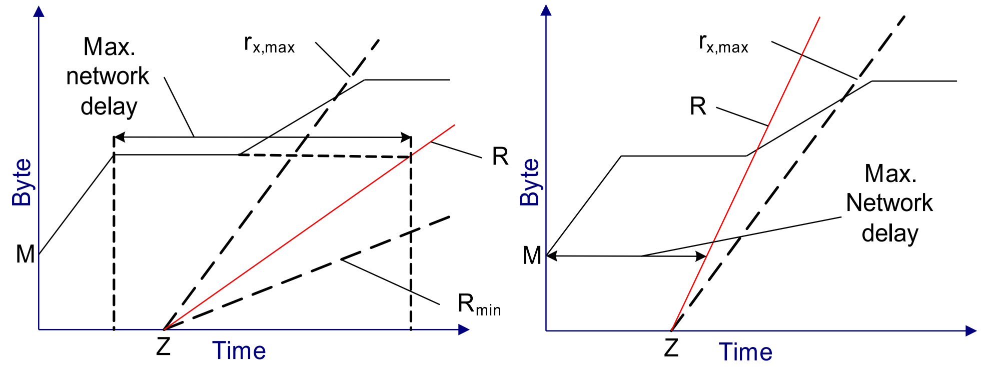


Figure 4.9: Arrival traffic curve from a smoothed video source and the max. network delay for two different service rate cases.

Accordingly, the maximum network delay $T_{N,max}$ can be computed as

$$T_{N,max} = \begin{cases} \left(\frac{I_{max}}{R} + Z\right) - T_{SI}\left(1 - \frac{M}{I_{max}}\right) & \text{for } r_{x,max} > R > R_{min} \\ \frac{M}{R} + Z \ \text{ for } R > r_{x,max}. \end{cases} \tag{4.18}$$

Deriving R from Eq. (4.18) results in

$$R = \begin{cases} \frac{I_{max} + \Sigma_{j=1}^{i} C_j}{T_{e2e,max} - T_{SI}\left(1 + \frac{M}{I_{max}}\right) - \Sigma_{j=1}^{i} D_j} & \text{for } T_{N,r_{x,max}} < T_{N,max} \leq T_{N,R_{min}} \\ \frac{M + \Sigma_{j=1}^{i} C_j}{T_{e2e,max} - 2 \cdot T_{SI} - \Sigma_{j=1}^{i} D_j} & \text{for } T_{N,\infty} < T_{N,max} \leq T_{N,r_{x,max}} \\ \text{undefined} & \text{otherwise.} \end{cases} \tag{4.19}$$

where

$$\begin{aligned} T_{N,max} &= T_{e2e,max} - T_{S,max} = T_{e2e,max} - 2 \cdot T_{SI} \\ T_{N,r_{x,max}} &= T_{N,R=r_{x,max}} = \Sigma_{\forall j}\left(\frac{C_j}{r_{x,max}} + D_j\right) + \frac{M}{r_{x,max}} \\ T_{N,R_{min}} &= T_{N,R=R_{min}} = \Sigma_{\forall j}\left(\frac{C_j}{R_{min}} + D_j\right) + \frac{I_{max}}{R_{min}} \\ T_{N,\infty} &= T_{N,R\to\infty} = \Sigma_{\forall j} D_j. \end{aligned} \tag{4.20}$$

The buffer size depends only on the network initial delay Z. Generally, it can be written as

$$\text{buffer} = \bar{A}\left(Z + \frac{M}{r_{x,max}}\right) \tag{4.21}$$

to reuse Eq. (4.14). Note that Z is measured from the network side while the time τ in Eq. (4.14) is seen from the source. Therefore, Z is extended by $\frac{M}{r_{x,max}}$ in Eq. (4.21). However, for the considered networks the buffer calculation can be reduced to two cases as follows

$$\text{buffer} = \begin{cases} M + Z \cdot r_{x,max} \ \text{ for } Z \leq \left(T_{SI} - \frac{M}{r_{x,max}}\right) \\ \\ I_{max} \ \text{ for } Z > \left(T_{SI} - \frac{M}{r_{x,max}}\right), 2 \cdot T_{SI} < T. \end{cases} \tag{4.22}$$

The first condition $Z \leq \left(T_{SI} - \frac{M}{r_{x,max}}\right)$ implies that the initial delay Z is still in the first increasing period of the traffic curve. When Z has passed the first increasing period of the traffic curve, or $Z > \left(T_{SI} - \frac{M}{r_{x,max}}\right)$, it should be guaranteed that the next T_{SI} does not transmit anything, because there is no information available about the size of the next P- or B-frame. This is equivalent to impose another constraint, i.e., $2 \cdot T_{SI} < T$ (see Figure 4.8). By this condition, if at least 2 smoothing intervals T_{SI} fit into a frame period T (there are 3 intervals in Figure 4.8.), then the next T_{SI} after the first increasing interval that transmits the I-frame will not be sending anything. Accordingly, the required buffer size has to be large enough to store the I_{max}.

4.2.2 Reshaping

The effects of traffic reshaping on the average and maximum end-to-end delays are studied in [69] using Network Calculus and in [83] using a simple fluid model. Placing traffic shapers in the interconnected switches decreases jitter and burstiness of the traffic flow. The average end-to-end delay is increased while the maximum delay is not affected.

In the following, the fluid model from Figure 4.10 is discussed to show that the reshaper decreases jitter and does not increase the maximum end-to-end delay if its envelope is selected properly. Let S_1 and S_2 be identical ideal delay elements with a service rate R. The input stream entering S_1 is shaped by a source shaper with the shaping envelope A. All packets of this stream are delayed by a constant value and sent with the rate R when they exit S_1. A resulting traffic envelope after the stream has passed through S_1 is A' which also includes the delay introduced by S_1's delay element. The same happens when the stream traverses S_2. The resulting traffic en-

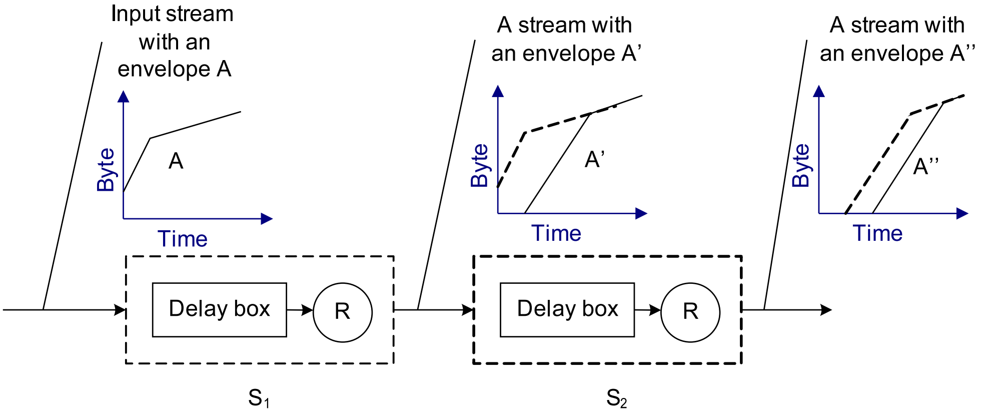

Input stream with an envelope A
Byte
A
Time
A stream with an envelope A'
Byte
A'
Time
A stream with an envelope A''
Byte
A''
Time
Delay box
R
Delay box
R
S₁
S₂

Figure 4.10: Reshaper motivation by a traffic curve passing through two network elements presented as ideal fluid systems.

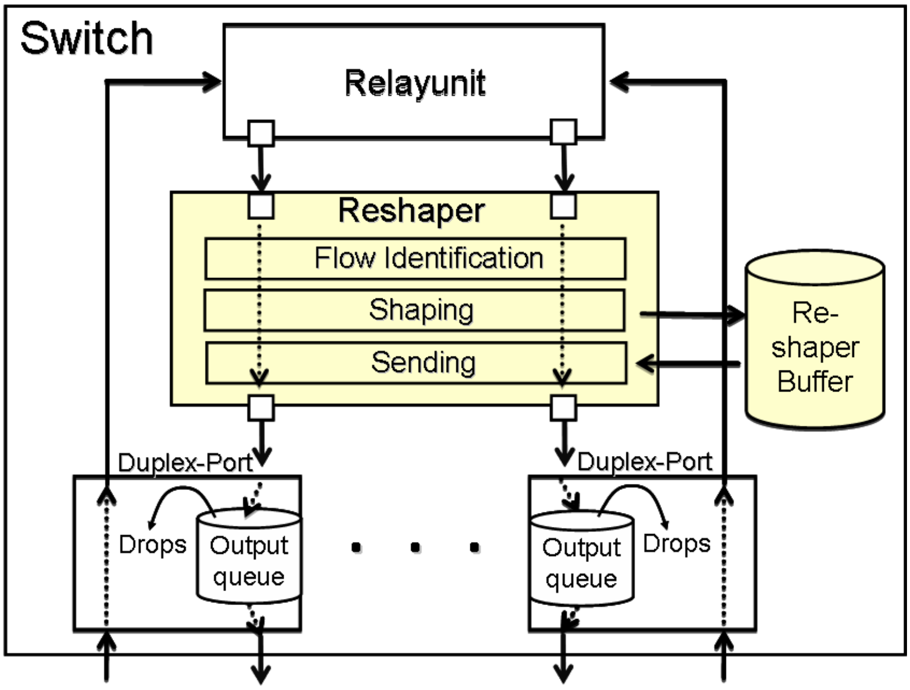


Figure 4.11: The designed reshaper architecture in a switch. The reshaper is slightly colored.

velope A'' shows an additional delay introduced by S_2's delay element. It is easy to conclude that by using a traffic shaper with a shaping envelope A' inserted between S_1 and S_2, the maximum end-to-end delay will not increase. In fact, any shaping envelope A''' that is more ”relaxed” than A' such that

$$A'''(\tau) \geq A'(\tau) \text{ for } \tau \geq 0 \tag{4.23}$$

can be used without introducing additional delay to the maximum end-to-end delay. In a more realistic system where S_1 and S_2 are replaced by network elements NE_1 and NE_2, representing for example two switches, the conclusion still holds. Realistic network elements do not necessarily delay all packets with a constant maximum delay like the delay elements from Figure 4.10 do. Instead, some packets are forwarded faster than the others causing jitter as previously discussed. A traffic curve after NE_1 can therefore be larger than A' which means A' is not an envelope at this point but is rather a lower bound. This also means that if a shaper with an envelope A' is placed at this position, it would delay early packets to conform with A' before entering NE_2, which results in an increased average end-to-end delay. However, the maximum end-to-end delay does not grow. In conclusion, placing traffic shapers in the cascaded NEs such as switches helps in reshaping the stream that has been distorted by the jitter of the previous switch back to its original envelope. Thus, the required buffer size at the client can be reduced at the cost of a higher computational load in the interconnected switches. Consequently, a stream-based reshaper has been designed and implemented in the switches of the simulation tool as shown in Figure 4.11. It identifies incoming traffic flows from their sender and receiver MAC and UDP/TCP port addresses and loads accordingly the shaping parameters, i.e., shaper type and shaper settings to reshape the traffic flow if necessary. After reshaping, data packets are forwarded to the output ports of the switch to be scheduled and sent to their destinations. As shown in Figure 4.11, a common buffer space is defined for reshaping of all flows in each switch. The size of this buffer

space is determined by separately computing the maximum required reshaper buffer size for each flow and summing them up.
The maximum reshaper buffer size Q_{RS} has been computed in [83] for a data flow as the maximum vertical distance between the incoming traffic from the source shaper $\overline{A}(\tau)$ and the outgoing traffic from the reshaper after some time $\overline{A}'(\tau - Z_{prev})$ where $Z_{prev} = \sum_{j=prev(i)}^{i-1} \frac{M}{R} + D_j$ and $prev(i)$ defines the last switch with a reshaper[2]. Q_{RS} in each switch is accordingly defined as

$$Q_{RS} = \overline{A}(\tau) - \overline{A}'(\tau - \frac{M}{R} - D). \tag{4.24}$$

In this work, $\overline{A}(\tau)$ and $\overline{A}'(\tau)$ are considered to be identical. The required reshaper buffer size is thus computed as the amount of data that the source shaper sends within the time $\frac{M}{R} + D$, as shown in Figure 4.12(a). However, since [83] assumes an instantaneous transmission of data between the switches, the above mentioned reshaper buffer computation is very pessimistic. In this work, the reshaper buffer model is enhanced by considering the limited link capacity R_L between the switches. Thus, data packets from the previous switch NE_{i-1} arrive with a maximum rate of R_L at the reshaper of the current switch NE_i. Figure 4.12(b) shows a shorter vertical distance between the input and output curves, i.e., a smaller reshaper buffer size compared to Figure 4.12(a). In the following, an analytical model based on the assumptions from Figure 4.12(b) is derived for reshaper buffer size computation with Token Bucket as source shaper and reshaper.

1. For $I_{max} > B \wedge Z_{prev} >$ Burst period $\wedge\ r < p < R_L$, $Q_{RS} = P_1 - P_2$ in Figure 4.12(b), thus

$$Q_{RS} = \left(b + \left(\frac{M}{p} + \frac{M}{R_L} + D\right) \cdot r\right) \cdot \frac{1 - \frac{p}{R_L}}{1 - \frac{r}{R_L}} + \frac{Mp}{R_L}. \tag{4.25}$$

2. For $I_{max} < B \wedge Z_{prev} <$ Burst period $\wedge\ r < p < R_L$

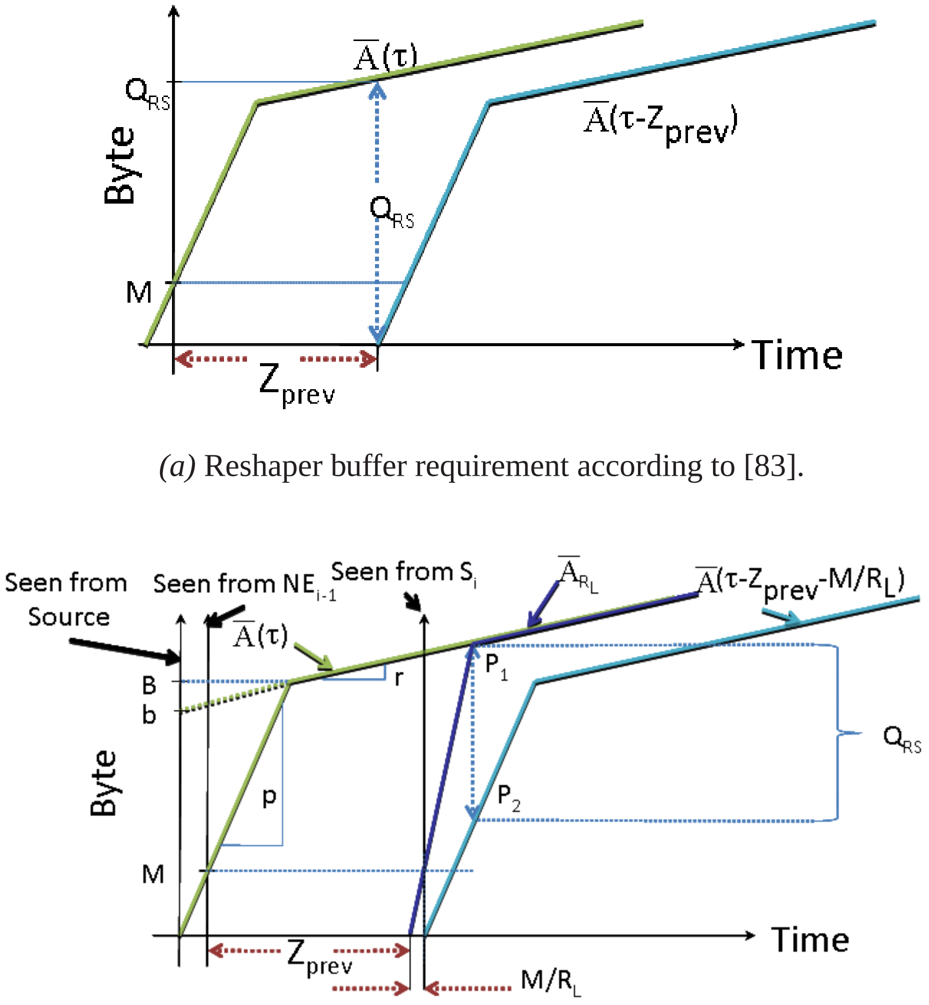
$$Q_{RS} = M + \left(\frac{2M}{R_L} + D\right) \cdot p. \tag{4.26}$$

3. For $R_L < p$

$$Q_{RS} = M. \tag{4.27}$$

A reshaper has also been integrated into the access point. In this context, packets coming from the wired IP/Ethernet-based network are forwarded to the relay unit which decides to either send the packets to the WLAN interface or back to the Ethernet interface or to drop them if the indicated destination addresses do not exist. On the way toward the WLAN interface, the packets pass through the reshaper module as shown in Figure 4.13. Because of layer-3 multicast addresses, the reshaper checks the source IP addresses of the incoming packets instead of their source MAC addresses as it is done in the switches. By also considering the source and destination TCP/UDP ports, all incoming streams can be identified. No reshaping is applied to the packets coming from the WLAN network.

[2] In the present work, $prev(i)$ is the previous switch, because except for the first switch, all other switches are equipped with a reshaper.



(a) Reshaper buffer requirement according to [83].

(b) Reshaper buffer requirement by considering the limited input link capacity R_L. NE_{i-1} defines the previous switch in the network while S_i determines the reshaper in the current switch NE_i.

Figure 4.12: Reshaper buffer requirement.

4.2.3 Results

The introduced analytical and simulation models from Chapter 3 enhanced with traffic shaping have been applied for the analysis in this section.

Shaper Configuration

Token Bucket Based on all established constraints for the Token Bucket shaper in Section 4.2.1, a feasible area is defined in Figure 4.14 to set the shaper parameters for the DVD application in the double star network, as an example. The I_{max}, GoP size and frame rate are obtained from Table 2.5. The token regeneration rate r is set to 1995081 bytes/s while the Buffer Limit is set to 30 MTU-sized packets. All (β, T_{burst}) pairs in the feasible area fulfill the QoS requirements of the DVD application. The coordinate (3.5, 0.61) from the feasible area in Fig-

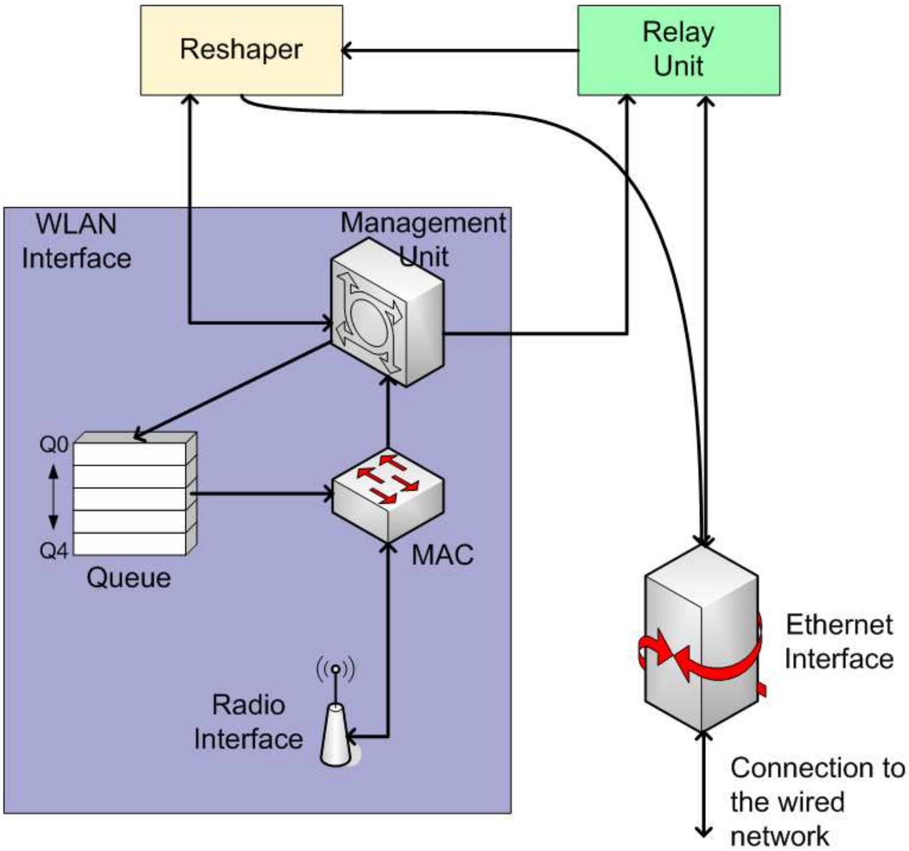


Figure 4.13: Access Point architecture with the integrated reshaper.

Table 4.1: Shaper settings and the required service rate (per stream) for both networks from Figures 3.3 and 3.4. All rates are given in Mbit/s. The bucket size b is given in bytes.

		(β, T_{burst})	p	b	r	R
Double Star	Cam	(2.25,0.13)	39.01	182330	16.58	37.36
	DVD	(3.5,0.61)	17.03	39263	15.96	16.22
Ring	Cam	(2.25,0.11)	46.11	203050	16.58	44.03
	DVD	(3.5,0.61)	17.03	39260	15.96	16.82

ure 4.14 is selected and used to calculate the p and b parameters of the Token Bucket shaper. Table 4.1 lists all resulting (p, b, r) triplets and the maximum R values for the DVD application and driver assistance cameras in both networks.

Simple Traffic Smoother (STS) Among all allowed T_{SI} values from the analytical model (Section 4.2.1), the best buffer size and service rate values for the DVD application and cameras are obtained with the configurations from Table 4.2 for both network networks.

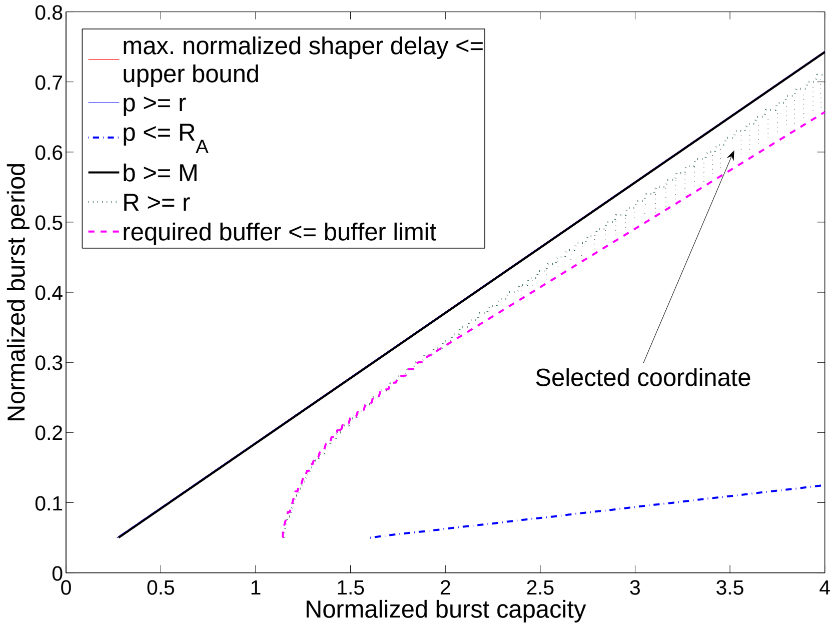


Figure 4.14: Feasible area (shaded area in the graph) to define β and T_{burst} values for the DVD application in the double star network and an example coordinate.

Table 4.2: STS settings, the required service rate (per stream) and buffer size for both networks from Figures 3.3 and 3.4. All rates are given in Mbit/s. T_{SI} is given in milliseconds and the queue size as the number of MTU-sized packets.

		T_{SI}	R	Queue
Double Star	Cam	16	63.6	25
	DVD	35	22.3	4
Ring	Cam	11	56.92	156
	DVD	35	25.4	30

Traffic Reshaper Leaky Bucket and Token Bucket have been identified as appropriate reshaping algorithms. They have been configured with the parameters from Table 4.1. STS, on the other hand is inappropriate as a reshaper. First, because of its maximum shaping delay time of $2 \cdot T_{SI}$ that would cause the excess of $T_{e2e,max}$ for nearly late packets. The second reason is that the source shaper and reshaper are not synchronized. Thus, data packets that are sent within one T_{SI} by the source shaper, can be transmitted by the reshaper within $2 \cdot T_{SI}$ which causes jitter and does not recover the original traffic curve.

As mentioned in the previous section, the reshaper compensates the negative effects of preceding switches in the network. However, according to [34], the application of reshapers is useful only for more than two cascaded switches. Therefore, reshaping has not been applied in the double star network. Instead, reshaping in the ring, especially, for the multimedia traffic that suffers from a larger jitter due to weighted fair queuing, has proven to be very useful in reducing bursts as shown in Figure 4.15 via simulations. It shows the cumulative frequency of occurrence of DVD packet arrival rates in all switches. Values with and without reshaper application are

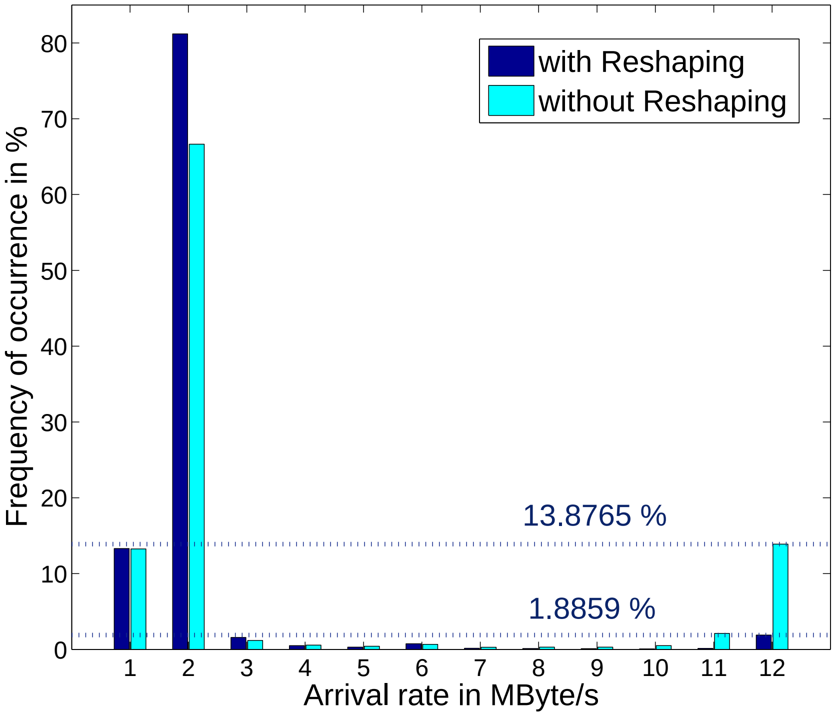


Figure 4.15: The effect of reshaping for the DVD application in the ring network when Token Bucket is used as source shaper and reshaper.

shown. With reshaper application higher arrival rates occur much less frequent (about 13 times less) than without reshapers. The reduction of traffic bursts obviously impacts the buffer size requirement in the switches which is discussed in the following.

Resource Usage versus QoS Performance

Wired Core Network The results from analytical models define the required resources in terms of buffer size and service rate for a QoS-aware data transmission. The networks in the simulation model have been configured according to the analytical results. Simulation results indicate the achieved QoS performance in terms of lost and late (later than 33 ms and 100 ms for cameras and DVD, respectively) packet rates in the investigated networks from Figures 3.3 and 3.4 with the analytically computed buffer sizes but a limited link capacity of 100 Mbit/s according to [6]. The occurrence of late and lost packets is a result of the limited transmission rates in the simulation model that are smaller than the analytically computed service rates. All indicated results are obtained with the same random generator seed to be comparable among each other and lie within certain confidence intervals with the probability of 95%. In order to determine the confidence intervals the student-t distribution has been applied, since the streams from video sources follow a Normal distribution.

The first simulation results show that the Token Bucket (TB) shaper with the selected settings from Table 4.1 behaves like a Leaky Bucket (LB) with the same peak rate.

Table 4.3 shows the amount of lost and late packets in the double star network (Figure 3.3) with different source shapers. The indicated loss rate values for camera and DVD lie within the 95%

Table 4.3: Analytically computed buffer sizes for queues 1 and 2 and the related simulation results in terms of late and lost packets for cameras and DVD in the double star network. For queue 0, the values from Table 3.3 have been used. LB, TB, STS and Desync. stand for Leaky Bucket, Token Bucket, Simple Traffic Smoother and camera desychronization, respectively.

	No Shaper	LB/TB Shaper	STS Shaper	STS Shaper+Desync.
$W_1 : W_2$	9:1	11:1	10:1	9:1
$Q_1/Q_2/\sum Q$	157/36/193	65/22/87	25/4/29	27/7/34
Cam/DVD Loss (%)	0.129/0.427	1.139/0.012	3.02/0.495	0.118/0.408
Cam/DVD Late (%)	0.007/0	0.009/0	0.213/0	0.049/0

confidence intervals of [0.113, 0.186] and [0.201, 0.433] when no source shaper is used, [1.137, 1.254] and [0.005, 0.013] when TB or LB is used, and in [2.343, 2.952] and [0.273, 0.497] when STS is used as source shaper. The camera late packet rates from Table 4.3 lie within [0.006, 0.01] when no source shaper is used, in [0.006, 0.01] when TB or LB is used, and within [0.167, 0.197] when STS is used as source shaper while there are no late DVD packets. As previously mentioned, reshaping does not affect the performance for two cascaded switches and is therefore not analyzed in the double star network. The throughput bottleneck is the connection between the two switches which is overloaded by the five simultaneously transmitting cameras. Camera desynchronization by $t_{Gap} = \left\lfloor \frac{\text{Frame period}}{\text{\# Cameras}} \right\rfloor = \left\lfloor \frac{33\,\text{ms}}{5} \right\rfloor = 6\,\text{ms}$ reduces bursts and consequently packet losses significantly by avoiding burst overlaps. However, simulations have shown that desynchronization alone is not enough to meet the QoS requirements. The last column of Table 4.3 shows the simulation results when all five cameras are desynchronized by 6 ms each and STS is used in video sources. Indeed, the best transmission performance with the lowest resource usage is achieved by this configuration.

Table 4.4 shows the simulation results of the ring network (Figure 3.4). Due to longer transmission times and intermediate buffering in the ring, camera desynchronization is not required. As explained before, the STS is not adapted as a reshaper and is therefore not included in the reshaping results of Table 4.4. The listed queue sizes define the sum of switch output queue and reshaper buffer requirements. According to Table 4.4, the employment of reshapers increases the amount of camera late packets in the network. On the other hand, the total required buffer size is reduced by using the reshaper. In addition, the DVD application benefits from reshaping in terms of lost packets when comparing TB shaper/No reshaper and TB shaper/LB reshaper DVD loss rate results. The 95% confidence interval of [0.004, 0.022] indicates the statistical significance for the TB shaper/No reshaper setting. No losses were observed during the 10 iterations with the TB shaper/LB reshaper setting. Consequently, by accepting the little increase of camera late packets of approximately 0.1%, the configuration TB shaper/LB reshaper (or TB shaper/TB reshaper) provides the best transmission performance and resource usage trade-off.

As mentioned in Chapter 3, the analytical model delivers pessimistic results due to its generality and simplicity. The required service rates from the analytical model cannot be fully provided in

Table 4.4: Analytically computed buffer sizes for queues 1 and 2 including output queues and reshapers and the related simulation results in terms of late and lost packets for cameras and DVD in the ring network. For queue 0, the values from Table 3.4 have been used. LB, TB and STS stand for Leaky Bucket, Token Bucket and Simple Traffic Smoother, respectively. $W_1 : W_2$ has been set to 13:1 for cases "No Shaper", "LB Shaper" and "TB Shaper". 9:1 has been used for the "STS Shaper".

		No Shaper	LB Shaper	TB Shaper	STS Shaper
No Reshaper	$(Q_1 + Q_{RS})/(Q_2 + Q_{RS})/\sum Q$	79/39/118	156/23/179	156/23/179	156/30/186
	Cam/DVD Loss (%)	0.025/0.032	0.001/0.018	0.001/0.018	0.003/0.007
	Cam/DVD Late (%)	0.061/0	0.079/0	0.079/0	1.121/0
LB Reshaper	$(Q_1 + Q_{RS})/(Q_2 + Q_{RS})/\sum Q$	205/62/267	125/13/138	125/13/138	55/11/66
	Cam/DVD Loss (%)	0.001/0	0.001/0	0.001/0	0.001/0.001
	Cam/DVD Late (%)	0.195/0	0.195/0	0.195/0	1.333/0
TB Reshaper	$(Q_1 + Q_{RS})/(Q_2 + Q_{RS})/\sum Q$	205/62/267	125/13/138	125/13/138	55/11/66
	Cam/DVD Loss (%)	0.001/0	0.001/0	0.001/0	0.001/0.001
	Cam/DVD Late (%)	0.195/0	0.195/0	0.195/0	1.333/0

Table 4.5: Effective buffer sizes for queues 1 and 2 computed by simulations for the specific transmission scenario from Chapter 3 in order to maximally reduce lost and late packet rates. TB reshaping results are equal to LB reshaping results and are therefore not listed here. $W_1 : W_2$ has been set to 13:1 for cases "No Shaper", "LB Shaper" and "TB Shaper". 9:1 has been used for the "STS Shaper".

		No Shaper	LB Shaper	TB Shaper	STS Shaper
No Reshaper	$Q_1/Q_2/\sum Q$	99/60/159	85/23/108	85/23/108	98/30/128
	1% limit for $Q_1/Q_2/\sum Q$	55/23/78	38/11/49	38/11/49	43/5/48
LB Reshaper	$Q_1/Q_2/\sum Q$	80/20/100	80/20/100	80/20/100	91/14/105
	$(Q_1 + Q_{RS,1})/(Q_2 + Q_{RS,2})/\sum Q$	205/62/267	120/29/149	120/29/149	106/18/124
	1% limit for $(Q_1 + Q_{RS,1})/(Q_2 + Q_{RS,2})/\sum Q$	161/48/209	66/12/78	66/12/78	52/8/60

the simulations due to the limited link capacity of 100 Mbit/s. Table 4.5 shows the effectively required buffer sizes in the ring computed by simulations for the specific transmission scenario from Chapter 3 to obtain the lowest lost and late packet rates. The results of Table 4.5 can be explained as follows. First, in the considered ring network, the best performance and resource usage trade-off is not achieved by the TB shaper/LB reshaper as concluded from the analytical model. But it is obtained with a LB or TB shaper/No reshaper configuration when the reshaper buffer requirement is taken into account. Second, the application of reshapers reduces the switch output queue size also for the VBR traffic ([83] and [34] proved the buffer size reduction for the CBR traffic.). An example is shown in Figure 4.16 by computing the relative reduction of buffer places per allocation. -100% in Figure 4.16 indicates that the marked buffer place has not been allocated at all with reshaper application, e.g., the buffer places 19, 21 to 23 have not been assigned to any DVD packets with reshaper application. However, by considering the reshaper buffer (Q_{RS}), the total buffer size in each switch is increased according to Table 4.5. Third, the 1% limit shows a significant buffer size reduction down to approximately half of the required buffer size in both cases, with and without reshaping. This means that by allowing a small amount of packet loss, i.e., $\leq$ 1% for camera and DVD applications in the considered worst case transmission scenario, almost 50% less buffer size is needed to assure QoS which shows a significant cost saving potential.

Wireless Peripheral Network In the following, the Token Bucket shaper configured according to Table 4.1, is assumed to be integrated in all video sources of the double star and ring networks from Chapter 3. Thus, high traffic bursts are attenuated and the transmission scenario from Section 3.3 can be realized. Also, Token Bucket reshaping in the access point has been investigated with the settings from Table 4.1. It turned out not to be as efficient as in the switches

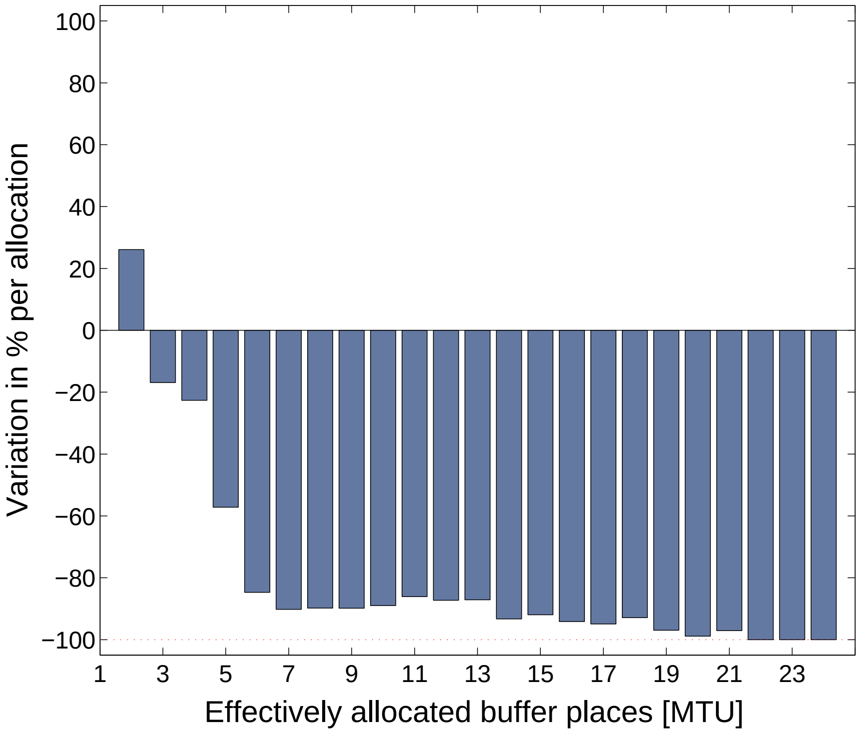


Figure 4.16: The relative reduction of buffer places per allocation with and without reshapers for the DVD application in the ring when Token Bucket is used as source shaper and reshaper.

mainly because of the FEC bursts that cannot be shaped. Traffic shaping in the FEC module is not feasible, since the FEC traffic is dynamically adapted to the channel condition and the different data streams. Therefore, reshaping in the access point is not further considered in this work. The applied adaptive FEC mechanism reduces the introduced 2% packet loss rate down to 0.48% by imposing some overhead as shown in Table 4.6. The comparably low overhead underlines the advantage of multicast over unicast which would require a separate transmission of each stream for each of the two wireless receivers in the car. In the multicast scenario, the total amount of traffic is $\text{Traffic}_{multicast}$ = Traffic[sources] + Traffic[FEC] with a maximum overhead of 38% according to Table 4.6. This can be denoted as $\text{Traffic}_{multicast}$=1.38 · Traffic[sources] while for the unicast transmission the total traffic amounts to $\text{Traffic}_{unicast}$= 2 · Traffic[sources]. Therefore, for multicast transmission only $\frac{\text{Traffic}_{multicast}}{\text{Traffic}_{unicast}} \cdot 100\% = 69\%$ of the unicast resources are needed. This entails a reduction of traffic load by 31%. Accordingly, multicast with the adaptive FEC is the preferred approach and is therefore further discussed in the following.

After computing the service rates in the WLAN network via Eq. (3.8), buffer size requirements in the access point can be calculated by using Eq. (4.5). Table 4.7 shows the analytically computed resource requirements for video applications and compares them with the simulation results. The analytical model does not include the FEC queue usage. A comparison of the analytically computed queue sizes with those obtained via simulations in Table 4.7 without FEC queue size consideration shows that, as expected, the analytical model defines an upper bound for the required network resources. The unacceptably large Q3 sizes, 583 and 1007 MTU-sized packets

Table 4.6: FEC Overhead with and without TXOP for the ring [R] and the double star [DS] networks. The influence of K-CAN data is shown for both networks when TXOP is enabled.

	Scenario	FEC Overhead in %
w/o TXOP	[R]	21.654
	[DS]	23.16
TXOP	[R]	23.006
	[DS]	22.002
	[R], K-CAN	38.414
	[DS], K-CAN	38.675

Table 4.7: Maximum Queue Usage: analytical results versus simulation results.

		analytical		Simulation				
		w/o TXOP	TXOP	w/o TXOP		TXOP		
				w/o FEC	FEC	FEC	w/o FEC + K-CAN	FEC + K-CAN
Ring	Cam (Q2)	164	131	63	123	67	39	86
	DVD (Q3)	314	243	62	583	44	26	93
Double-Star	Cam (Q2)	159	122	56	351	44	22	75
	DVD (Q3)	311	240	55	1007	35	20	96

in ring and double star networks, indicate the large number of FEC packets needed for the two applications DVD and Audio CD that are both assigned to Q3 and scheduled via weighted fair queuing with a lower priority than the camera packets in Q2. Finally, the application of TXOP reduces the required buffer size significantly in both networks which is a very promising result for in-vehicle wireless communication. Figure 4.17 shows a histogram representing the frequency of occupation of the queue usage for different queues in the ring network from Figure 3.4. These values have been stored over a simulation time of 600 s. As in the wired Ethernet network, the large queue sizes are only very rarely occupied. Consequently, if a small amount of packet loss is tolerated, much smaller queue sizes can be used than those shown in Figure 4.17.

Analyses have shown that PT- and K-CAN data cannot be transmitted together with all other data flows in the considered transmission scenario from Section 3.3. The reason is their small packet size of 64 bytes at the network layer that leads to a low throughput on the communication channel as explained in Section 3.3. CAN packets are assigned with the highest priority level after WLAN management data and are scheduled with the strict priority queuing. Therefore, they block the whole communication channel to transmit a total bit rate of 4.8 Mbit/s (Section 2.4.2) and prevent the transmission of packets from lower priority queues. While the transmission of K-CAN data together with other data flows becomes possible when frame bursting is applied, PT-CAN can only be transmitted when no other data flows are sent. PT-CAN sends 64 byte data packets with a high constant data rate of 4 Mbit/s which quickly saturates the WLAN channel. Simulation results with PT-CAN in the ring network are shown in Table 4.8. The large queue usage in queues 2 and 3 indicates the channel saturation, because those queues cannot send any

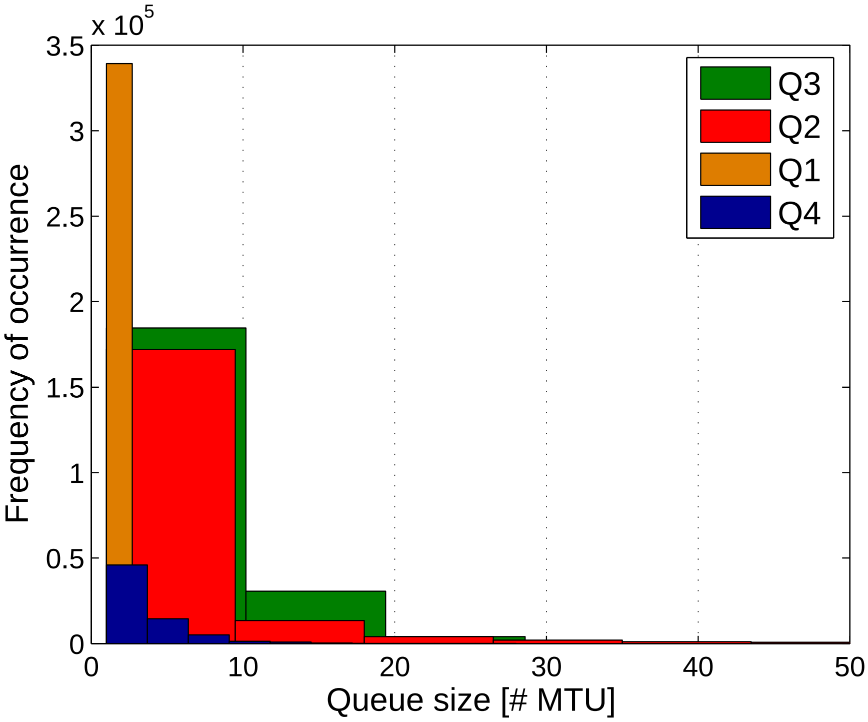


Figure 4.17: Frequency of occurrence of queue usage for different queues in the ring network with active TXOP and K-CAN.

Table 4.8: Queue Usage (maximum / mean / standard deviation σ) with enabled TXOP and PT-CAN in the ring network.

Queue	max	mean	σ
Q0	248	6	27
Q1	22	5	3
Q2	1E5	69854	33524
Q3	1E5	84151	28456
Q4	23	4	3

packets. Queue 1 has a maximum queue usage of 22 MTU-sized packets, which confirms the feasibility of an exclusive transmission of PT-CAN data with frame bursting.

Table 4.9 indicates the amount of lost and late packet rates for the two wireless hosts (WHost1/ WHost2) in the considered in-vehicle networks from Figures 3.3 and 3.4 with the FEC overhead. According to the channel measurement results mentioned in Section 3.3, lost and late packet rates do not differ much between the two receivers that are positioned at equal distances to the access point. Since all buffers are dimensioned carefully with the introduced analytical model, packet drops can be excluded. All loss rates in Table 4.9 are due to the channel condition. As discussed in Section 3.3, the Gilbert Elliot model introduces 2% packet loss rate. Consequently, all loss rates in Table 4.9 are around 2%. The application of TXOP improves the QoS performance for all data flows in terms of late packet rates. Table 4.10 enhances the results of Table 4.9 by considering the K-CAN data when TXOP is applied. While the loss rates remain around 2%,

Table 4.9: Late and lost packet rates given for WHost1/ WHost2 considering that all applications send data to the wireless hosts with and without frame bursting (TXOP). The results are obtained via simulations. Due to large throughput requirements CAN applications are disabled.

Appl.	Ring				Double Star			
	w/o TXOP		TXOP		w/o TXOP		TXOP	
	loss%	late%	loss%	late%	loss%	late%	loss%	late%
Camera5	2.11/ 2.12	2.51/ 2.48	1.90/ 2.00	0.34/ 0.34	2.10/ 2.18	3.44/ 3.39	1.99/ 1.91	0.07/ 0.07
DVD	2.04/ 2.09	0.92/ 0.90	1.18/ 1.91	0/ 0	2.16/ 2.18	1.59/ 1.60	2.14/ 1.91	0/0
Audio CD	2.15/ 2.11	0.84/ 0.82	2.10/ 2.22	0/ 0	2.16/ 2.26	1.18/ 1.18	1.21/ 1.04	0/0

the Camera5 late packet rates increase when K-CAN data is transmitted due to the output queue scheduling mechanisms of the access point. A comparison between the TXOP performance results of ring and double star networks from Tables 4.9 and 4.10 shows a higher late packet rate in the ring network that corresponds to [9] and is due to the higher number of cascaded switches in the ring. 10 simulation runs have been carried out, each with a different initial seed of the random number generator. However, the indicated results are obtained with the same random generator seed to be comparable among each other. They lie within certain confidence intervals with the probability of 95%. In order to determine the confidence intervals the student-t distribution has been applied, since the video streams follow a Normal distribution. For example, the confidence intervals for Camera5 and DVD late packet rates in the ring are computed to be [0.59891, 1.2019] and [-0.015, 0.038], respectively. [1.753, 2.096] and [1.922, 2.118] define the confidence intervals for the Camera5 and DVD lost packet rates in the ring network.

Table 4.10: Late and lost packet rates given for WHost1/ WHost2 considering that all applications (including K-CAN) send data to the wireless hosts with frame bursting (TXOP). Results are obtained via simulations.

Appl.	Ring		Double Star	
	loss%	late%	loss%	late%
Camera5	2.08/ 2.17	0.70/ 0.71	2.09/ 1.91	0.30/ 0.30
DVD	2.20/ 2.06	0/ 0	2.13/ 1.82	0/ 0
Audio CD	2.25/ 2.18	0/ 0	2.03/ 1.94	0/ 0
K-CAN	2.11/ 2.10	0/ 0	2.04/ 1.92	0.001/ 0.001

4.3 Prototypical Implementation

Traffic shapers can basically be realized in any of the OSI layers. However, the lower the layer in the protocol stack, the less jitter is introduced to the shaped data flows before entering the network. As a candidate for all observed traffic shapers, the Token Bucket has been implemented and analyzed for its performance in an end-system, i.e., a video server.

The temporal granularity at which the kernel of a Linux operating system is invoked, is determined by the timer tick rate, also referred to as the Hz value. Amongst others, the Hz value determines the scheduling granularity, e.g., how often bus interfaces are polled. This is particularly important for real-time detection of shaper events, e.g., in order to signal the availability of a new packet on time. The highest possible granularity in a standard Linux operating system is 1000 Hz, i.e., events are detected every 1 ms. Therefore, as also mentioned in [74], the theoretically analyzed smooth traffic curves represent only an approximation for a real system interrupted by system cycles as shown in Figure 4.18. The Token Bucket is available as a module (Token Bucket Filter (TBF)) in the Linux kernel space. It has been activated to function in our system. The applied testbed consists of a sender station (Dell 1.7 GHz Laptop, OS: Linux kernel 2.6.14, Ubuntu version 7.6, recompiled Kernel to 1 kHz) that transmits an MPEG-2 encoded video sequence with an average data rate of 7.02 Mbit/s for 42 seconds over an IP/Ethernet network to a receiver station. The TBF is configured to have a bucket size of 10 Mbyte, a burst size of 40930 byte, a token regeneration rate *r* of 5 Mbit/s and a peak rate *p* of 15 Mbit/s. Wireshark has been applied to measure the delay values of the TBF. It delivers the so called Delta values *D* (see Figure 4.18) for two consequent packets *i* and *j* computed as shown in Figure 4.19 and Eq. (4.28)

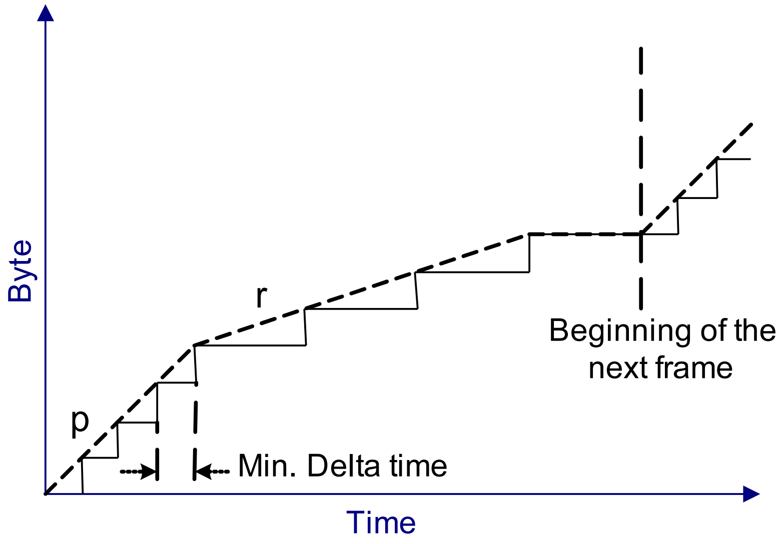


Figure 4.18: Theoretical Token Bucket traffic envelope curve versus real Token Bucket traffic curve.

$$D(i,j) = \left|(R_j - S_j) - (R_i - S_i)\right| \tag{4.28}$$

where *S* and *R* define the time stamps at sender and receiver applications, respectively. Table 4.11 shows the computed delay values between two consecutive packets from the measured Delta values. The results from Table 4.11 confirm [83] and [34] in that traffic shaping increases the mean end-to-end delay but does not exceed the maximum delay.

According to [74], the higher the data rate and system interrupt frequency, the larger is the CPU usage by the traffic shaper. [74] measured the CPU usage of a TBF for different data rates and

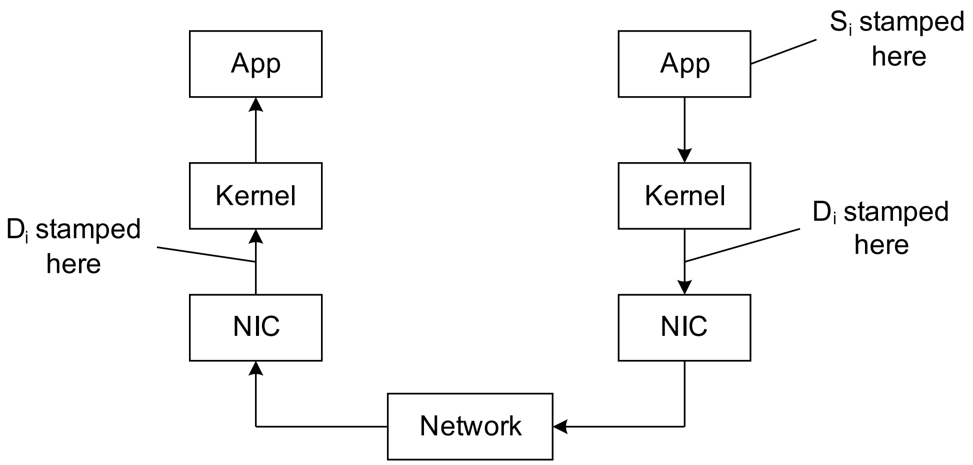


Figure 4.19: Wireshark timestamps packets at S and R for further measurements.

Table 4.11: Inter-packet delays computed from the measured Delta values by Wireshark for an non-shaped and a TBF shaped VBR video sequence.

Delay (ms)	No Shaper	TBF
Min	0.02	0.56
Max	23.02	20
Mean	3.077	4.428
Std	2.85	0.85

Table 4.12: CPU usage of TBF for different data rates and interrupt cycles Δ from [74].

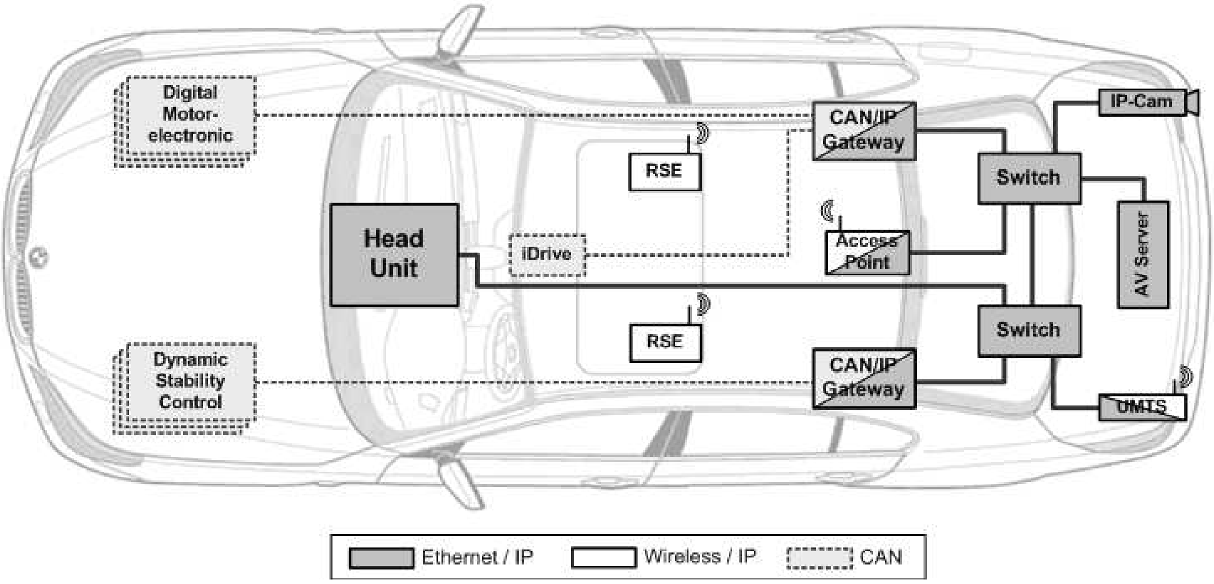
	40 Mbit/s	32 Mbit/s	20 Mbit/s
Δ = 1ms	11%	9%	7.2%
Δ = 100μs	21.2%	17.2%	11.9%

operating system interrupt frequency values expressed in cycle times Δ as shown in Table 4.12. To reduce the CPU load caused by traffic shaping, there are different possibilities such as

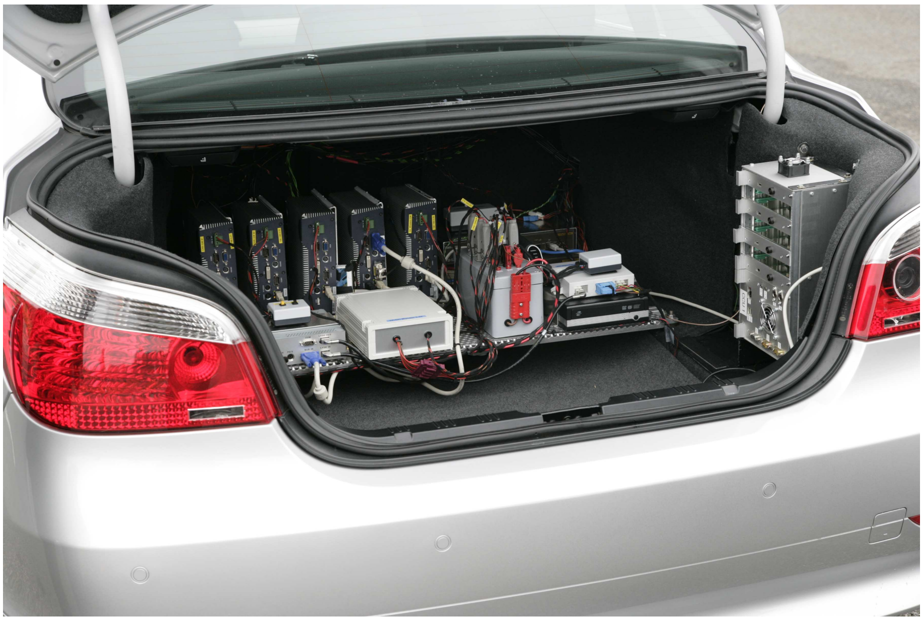
- software solution, i.e., parallel computing with network processors (CPU + Co-Processors): It is already available in most of Intel network processors, for example in Intel IXP420 Network Processor, where peripheral processors adopt computationally intensive network data operations such as IP header inspection, 100 Mbit/s full-duplex Ethernet packet filtering, checksum computation etc. from the CPU,
- hardware solution, i.e., application of off-the-shelf hardware implemented shapers. Several switch manufacturers have already implemented traffic shapers such as Leaky Bucket in hardware at switch output ports. The first hardware traffic shaper generation operates per port, but can be extended to work per stream.

Accordingly, it is possible to realize the proposed network architecture with traffic shapers by using standard components which is one of the most important requirements of the automotive industry.

As explained in [12], a prototypical IP-based network with the double star topology has been implemented in a BMW 530d to show that the proposed network architecture works properly in a real car. Figure 4.20(a) shows the realized prototypical network and Figure 4.20(b) shows the trunk of the prototypical vehicle with some of the network devices. A customized IP-based head-



(a) Overview of the prototypical IP-based in-vehicle network.



(b) The trunk of the prototypical vehicle with the network devices.

Figure 4.20: Prototypical IP-based in-vehicle (BMW 530d) network.

unit software has been implemented to represent the wired sink (headunit) in the IP/Ethernet-based network. The RSE sinks are realized by two Tablet PCs that are connected via WLAN to the in-vehicle network. The Tablet PCs run with the same software as the headunit and have exactly the same functions. The Precision Time Protocol (PTP) [47] implemented through the PTP daemon has been used in the application layer to synchronize sources and sinks. The PTP synchronization cycle has been set to 1 s. Thus, the continuous drift of the internal device clocks apart from each other is prevented and audio and video streams are played out synchronously at different sinks. Data can be transmitted to every sink in the IP-based network without any conversion. For example, a DVD movie can be watched on one of the Tablet PCs while the rear-view camera video runs on the other one. The prototypical vehicle is connected to the Internet through UMTS and WLAN. Thus, Internet applications such as IP-TV, IP-radio or video-based assistance services can be realized in the car. However, as mentioned in [12], security mechanisms such as Firewalls are needed to protect the car from outside malicious access. Furthermore, in order to demonstrate the real-time communication over the IP-based network, the CAN connection between the dynamic stability control system (DSC) and the digital motor electronic has been disconnected. As mentioned in Section 2.4.2, CAN data packets are packed into Ethernet frames in the CAN/IP gateways and transmitted through the two commercial manageable D-Link Ethernet switches over the IP-based network to the sinks. Thanks to the priority assignment mechanism, network dimensioning and traffic shaping in video sources, all QoS requirements could be met in all tested transmission scenarios as shown in [12].

Some measurement results are presented in the following for different transmission scenarios. In the first test scenario, only a DVD stream (average bit rate: 2.5 Mbit/s) is sent from the AV Server shown in Figure 4.20(a) over the two cascaded switches to the headunit. In the next scenarios, CAN (constant bit rate: 4.8 Mbit/s) and best effort (Transmission bit rate: 100 Mbit/s to fill the available link capacity) data have been additionally sent to the headunit. As mentioned in Chapter 3, CAN is assigned with the highest priority, DVD with a lower priority while the best effort data has the lowest priority. By activating and deactivating the QoS mechanisms different values have been measured as shown in Tables 4.13 and 4.14. Two different scheduling scenarios have been considered for switch output port queues. In the weighted fair queuing scenario (WFQ), DVD and best effort queues are assigned with weight values while the CAN queue supports the strict priority queuing. In the strict priority queuing scenario (SPQ), CAN and DVD queues apply the strict priority mechanism while the best effort queue is not supported by any scheduling mechanism. The transmission scenarios of Tables 4.13 and 4.14 indicated with 1. to 5. are summarized as follows.

1. QoS mechanisms are deactivated. DVD stream is transmitted.

2. QoS mechanisms are deactivated. DVD and CAN data are transmitted.

3. QoS mechanisms are deactivated. DVD, CAN and best effort data are transmitted.

4. QoS mechanisms are activated applying WFQ with weight values 55:1 for DVD and best effort data, respectively. DVD, CAN and best effort data are transmitted.

5. QoS mechanisms are activated applying SPQ for DVD and CAN data. DVD, CAN and best effort data are transmitted.

Table 4.13: Loss rate and delay values measured for CAN, DVD and best effort applications in different transmission scenarios.

Scen.	Loss (%)			Delay (ms)								
	CAN	DVD	BE	CAN			DVD			BE		
				min.	mean	max.	min.	mean	max.	min.	mean	max.
1.	-	0	-	-	-	-	0.37	0.4	0.59	-	-	-
2.	0	0	-	0.03	0.19	0.66	0.32	0.37	0.72	-	-	-
3.	4.24	17.04	12	0.11	0.38	0.85	0	0.75	1.77	0.08	0.25	0.6
4.	0	8.168	11	0.01	0.22	2.39	0.17	0.29	0.78	0.05	0.27	1.12
5.	0	0	12	0.08	0.48	1.85	0.18	0.3	1.35	0.02	0.24	1.16

Table 4.14: Jitter values measured for the DVD application in different transmission scenarios.

Scenario	Jitter (ms)		
	min.	mean	max.
1.	0	0.03	0.21
2.	0	0.04	0.36
3.	0	0.07	0.94
4.	0	0.07	0.47
5.	0	0.07	1.09

Tables 4.13 and 4.14 show acceptable delay and jitter values for the different applications in all transmission scenarios. The loss rate results are significantly improved by the activation of the QoS mechanisms in the network. The large loss rate of 8% for the DVD data in the WFQ scenario from Table 4.13 shows an incorrect operation of the switch schedulers in network overload situation. This issue should be carefully addressed in the automotive qualified switch design. However, it should be mentioned that the loss rate value of 8% is still much smaller than the loss rate of 17% when no QoS mechanism is applied. Thus, the importance of QoS mechanisms for the in-vehicle communication is confirmed in a real system.

4.4 Summary

In this chapter, traffic shaping is introduced as a mechanism to reduce the required resources in the in-vehicle network. It assigns desired characteristics, such as the peak rate upper bound, to the traffic flows when applied to sources and compensates delay jitter thus reducing buffer size requirements when applied to the intermediate network elements. Different traffic shaping algorithms have been presented and analyzed for an application in in-vehicle video sources, switches and the access point. In addition to the well-known traffic shaping algorithms, Leaky Bucket and Token Bucket, a novel traffic shaper called Simple Traffic Smoother (STS) has been introduced. Reshapers (i.e., traffic shapers in the switches and in the access point) have been implemented to operate per video stream. The analytical results show that in the considered worst case transmission scenario, the best QoS performance and resource usage trade-off is obtained by applying the STS as video source shaper and by desynchronizing camera streams in the double star network

while in the ring network the Token Bucket source shaper and Leaky Bucket reshaper lead to the best trade-off. However, simulation results indicate that in our specific transmission scenario, the best performance and resource usage trade-off in the ring is obtained by only applying Leaky Bucket or Token Bucket source shapers without reshaping. It turned out that reshaping reduces the switch output buffer requirements, but the total buffer size consisting of the output buffer and the reshaper buffer is increased. The introduced 1% limit shows that by allowing a packet loss rate less than 1%, the buffer consumption can be reduced by 50%. Traffic shaping in video sources is essential for wireless transmission while reshaping in the access point turned out to be inefficient. The reason is the FEC burst traffic that overloads the access point output queues and cannot be shaped, because it is dynamically adapted to the channel condition and the different data flows. By video source shaping and frame bursting, the considered wireless transmission scenario becomes feasible. A prototypical implementation of the traffic shaper confirmed the results from the analytical and simulation models. An experimental IP-based heterogeneous network has been implemented in a BMW 530d to show the feasibility of the proposed network architecture in a real car. All different in-vehicle traffic classes have been realized and evaluated in the prototypical network. The results have shown that the concept of an IP-based network architecture is feasible for in-vehicle communication.
To conclude, a general guideline for an appropriate network dimensioning with VBR source shapers is given in the following.

1. Define the statistics of the VBR sources. Important statistics, for example for compressed videos are GoP_{max}, I_{max}, n and f, which are explained in Table 2.5.

2. Use the traffic curves and the related formulas from Section 3.2.1 to compute the initial delay Z including the packetization delay C and the head-of-line blocking delay D.

3. Use the formulas from Section 4.2.1 that are derived from traffic curves to compute the required buffer size and service rate per switch for each traffic shaper. If more than two VBR sources with large data rates are connected to one switch, desynchronize them as it has been done for the double star network in Section 4.2.3. Compute the required resources accordingly.

4. Compare the analytical results for different traffic shapers and select the traffic shaper that leads to the lowest resource requirements. Note that the analytical model produces pessimistic results that should be interpreted as upper bounds.

5. Configure the network simulation model by the analytically computed shaper parameters and network resources.

6. If all required resources are provided in the simulation model, no lost and late packets should be observed. Try lower resources and check the lost and late packet rates. By trial and error, find the best settings for the lowest lost and late packet rates in the considered network.

7. Otherwise, if some resource limitations, for example on the link capacity are given, go through the following steps.

8. Simulate the configured network with a modified relationship between the switch buffer size and service rate. For example, for a limited link capacity use unlimited buffer sizes in the switches and register the effective buffer allocation as done in Section 4.2.3. Thus, you find the required network resources for the lowest loss and late packet rates.

9. Simulate several iterations to get reasonable confidence intervals for the results in order to define their statistical significance.

10. Configure a prototypical network system with the computed traffic shaper and resource settings. Note that the required network resources might be lower in the real system due to hardware restrictions.

5 Video Compression and Image Processing for Driver Assistance Systems

The current in-vehicle driver assistance network system is realized via point-to-point connections as mentioned in Section 2.1. Due to the growing number of cameras, the in-vehicle network is becoming increasingly complex, inflexible and costly. In order to reduce the complexity and cost, a heterogeneous IP-based network architecture has been proposed in Chapter 3 to interconnect ECUs including cameras in the car. However, in order not to exceed the link capacity while transmitting several video streams simultaneously over one Fast Ethernet link, video compression is required. Video streams from driver assistance camera systems have strict delay and quality requirements. In this chapter, applicable video codecs are analyzed for driver assistance services in the car. Metrics are defined to evaluate the performance of the algorithms primarily in the wired in-vehicle network. It is furthermore investigated how video compression algorithms should be parametrized in order to avoid negative effects on driver assistance image processing algorithms. Concepts for hardware realization of IP cameras with integrated video codecs are briefly described as an outlook to further improve the coding performance in the car.

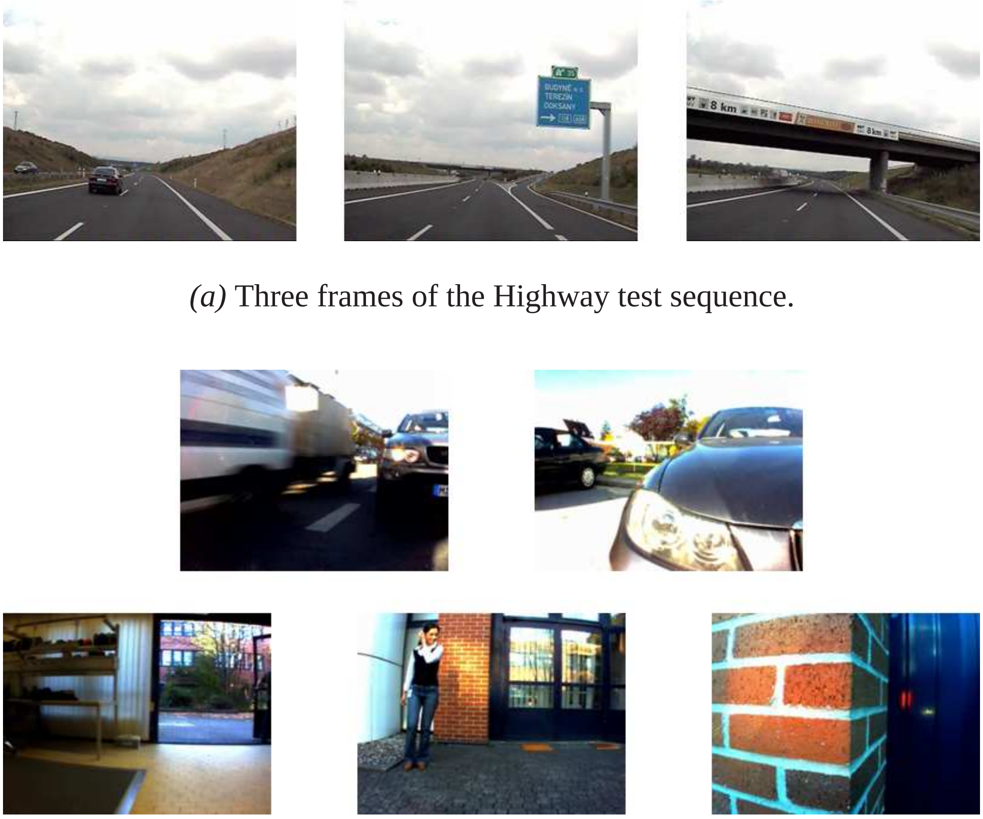
5.1 Analysis of Applicable Video Codecs for Driver Assistance Camera Systems

This section is concerned with the real-time compression of video streams from direct image-based driver assistance cameras introduced in Section 2.3.2. Examples are rear- and side-view cameras for the parking use case. In a real-time system, the correctness of an operation does not only depend on the logical correctness, but it also has to be performed within a given time period (deadline). As mentioned in Chapter 1, the real-time criterion for a camera application in this work is defined as an end-to-end delay of one frame interval, which is the strongest delay requirement for camera systems in the car. Looser end-to-end delay requirements of 100 ms can also be accepted for camera systems under certain circumstances. In this work, however, the lowest delay requirement of one frame interval has been chosen for an efficient and low delay video compression in future IP-based in-vehicle communication networks. Applicable video codecs realized in software are evaluated for their performance by using specific metrics. The most appropriate compression schemes with the best settings are determined and implemented in a prototypical testbed. Evaluation results are presented for wired communication. Similar investigations have been carried out in the literature such as in [119]. However, previous studies have not considered the in-vehicle communication requirements.

5.1.1 System Description

In the considered in-vehicle transmission system, each video frame is pre-processed after being captured at the source and then compressed. Afterwards, the bit stream is packetized and transmitted. Upon reception, the stream is depacketized, decoded, and possibly post-processed before being displayed at the sink. The wired communication network introduced in Chapter 3 is considered for the analysis in this section. The Ethernet network is assumed to be error-free due to its very low bit error rates according to [4]. It can therefore better illustrate the compression effects than error-prone wireless networks.
Several software codecs have been implemented in a prototypical testbed. They include Motion JPEG[1], MPEG-2 and MPEG-4 (part 2) realized by the libavcodec library [41] and the Xvid library [116]. The x264 library [127] is used for H.264/AVC (MPEG-4 part 10). For the decoding of H.264/AVC bit streams, the libavcodec decoder is used. The basic components of the prototype are as follows.

Test Sequences:



(a) Three frames of the Highway test sequence.

(b) Five frames of the VGA test sequence.

Figure 5.1: Test sequences.

[1] Since Motion JPEG is an established format in surveillance applications, it has been applied for performance analysis in this section.

1. Highway sequence [20]: The test sequence Highway shown in Figure 5.1(a) is subsampled to the CIF resolution and 4:2:0 format. Even though the scene and the driving speed in the Highway sequence are different than expected for a direct image-based camera, it still represents some of the typical properties. The sequence shows other cars that are passing by and the drive under a bridge with a change from a bright to a dark surrounding and vice versa. This represents some of the expected variations in the video motion and the lightning conditions. The contrast between the road with its dominating gray areas and the bright sky are also typical. Among the 2000 frames in the original video, frames 550 to 1349 are used, since the content of the frames before and after the selected subsequence is very similar. Therefore, the test sequence length is set to 800 frames.

2. VGA sequence: The second video sequence is produced from typical scenes captured by a rear-view camera. Different scenes were recorded with the camera that was used in the prototype design and combined into a sequence of 715 frames of VGA resolution. The test sequence shown in Figure 5.1(b) contains a scene in which another car drives by close to the camera, a scene in which another car is approached while reversing (typical parking situation), a scene in which a person walks in front of the camera, a reversing scene at low speed in a darkened environment to reflect dimness, as well as reversing towards a wall with high spatial details in which the characteristic distortions of a wide-angle lens become visible.

FireWire Camera: For frame acquisition, the camera DFK 21AF04 from The Imaging Source [117] is used. It captures YUV 4:2:2 frames and is connected to a PC via a FireWire cable. In the implementation, it is controlled via the unicap library [21].

Server and Client: The server and the client are implemented upon two computers connected via Fast Ethernet using a layer-2 switch. Test computer 1 (server) is a Dell Precision notebook with an Intel Dual Core CPU running at 2 GHz. Test computer 2 (client) is a desktop PC with an Intel 3.4 GHz CPU. On both test systems, the Linux distribution open SUSE 10.1 with kernel 2.6.16 working with 1000 Hz Kernel cycle frequency is used.

Data Transmission and Session Control: For the transmission, the UDP protocol without any additional higher layer protocols is employed. Session control is implemented through the SNMP framework.

5.1.2 Applied Metrics

Complexity: Video standards only define the bit stream syntax and the decoding process. Thus, developers have degrees of freedom in their choices when implementing an encoder. The algorithmic complexity of an encoder is then affected by the specific implementation architecture, its data and memory structures, and optimizations. Further, when looking at the number of features and options that can be combined for a standard like H.264/AVC, it becomes clear that the complexity of both encoder and decoder is affected by the feature choices. However, the algorithmic or computational complexity is often based on the processing time for a given sequence and on a particular platform. Despite the implementation uncertainties, the processing time still provides useful and valuable indications and is defined as the metric of the computational complexity here.

End-to-End Delay: Low delay has been identified as one of the major performance criteria for in-vehicle camera systems. In this context, the end-to-end delay, i.e., the time from image acquisition to the reception and presentation at the client, is the main metric. It is an accumulation of the individual delays that occur in a video acquisition and transmission system. The end-to-end delay requirements are defined in Section 2.4.1.

Video Quality: The video quality at the sink is another important metric for the comparison of video compression systems. Besides being a function of the data rate and the video sequence properties, the perceived video quality also depends on environmental conditions, such as lighting, the sampling density in the form of display size versus resolution, the viewing distance, etc. For further analysis in this work, the environmental conditions have been considered to be static for all compression systems. Two well-known objective metrics, the Peak Signal-to-Noise Ratio (PSNR)[2] and the Structural Similarity Index (SSIM) [134], [135] have been applied for the analysis of the compression algorithms. The PSNR cannot be directly translated into perceived image quality when comparing different kinds of distortion, as it is a mere mathematical function [23]. The perceptual approach SSIM, on the other hand is able to predict the perceived quality of an image or a video automatically based on properties of the human visual system. SSIM takes the structures of an image or video sequence into account. Thus, it interprets a scene based on its structures and the changes in the structures. A higher SSIM value corresponds to higher similarities between two compared video sequences with a maximum value of 1 representing two identical sequences.

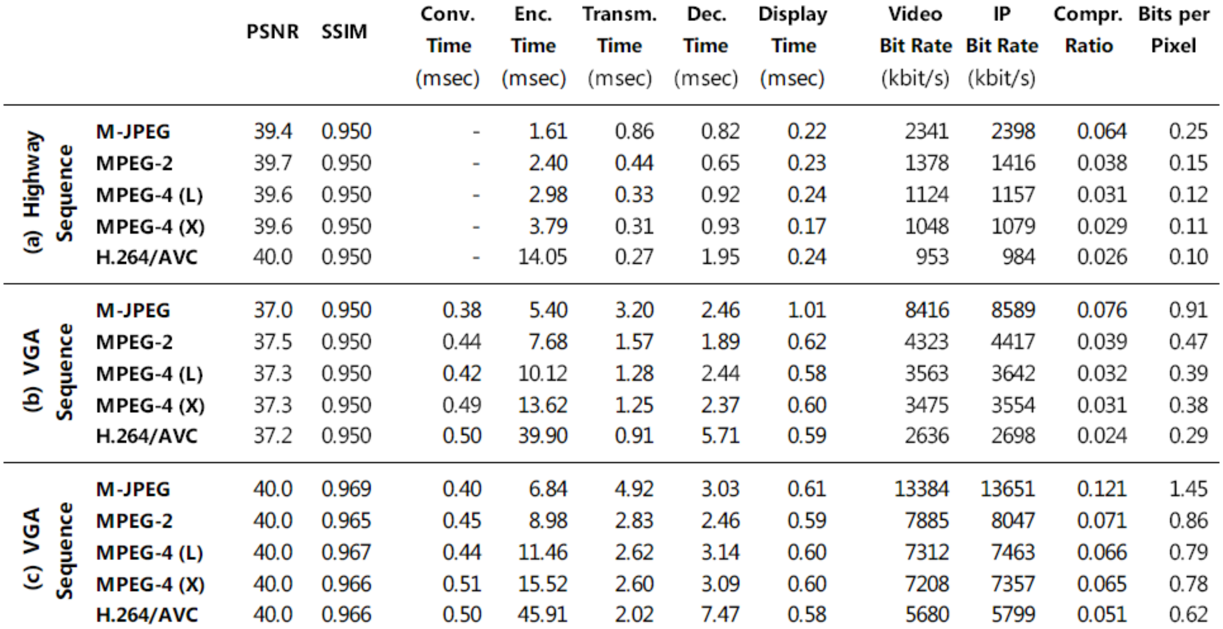
5.1.3 Comparison Results

Computational Complexity

In the following, the compression ratio as well as the processing times of Motion JPEG (libavcodec), MPEG-2 (libavcodec), MPEG-4 (libavcodec), MPEG-4 (Xvid), and H.264/AVC (x264) are presented. For the sake of simplicity, the codecs are denoted by M-JPEG, MPEG-2, MPEG-4 (L), MPEG-4 (X), and H.264/AVC, respectively. They are parametrized such that the resulting video quality is the same for all codecs. The SSIM value for the Highway sequence is 0.95 (condition a). For the VGA sequence, two conditions are defined: an SSIM value of also 0.95 (b) and a PSNR value of 40 dB (c). Table 5.1 summarizes the results of the three conditions. The table includes the measurements of the video quality, processing times, and the compression ratio. The quality is given in terms of the PSNR and the SSIM. As expected, H.264/AVC requires the highest complexity in terms of encoding and decoding times in all measurements.

[2] $PSNR = 10\ \log_{10} \frac{N^2}{MSE}$, where MSE defines the mean squared error and $N = (2^n - 1)$ is the dynamic range of the pixel values with n being the number of bits used to represent the value of a pixel per component, is a mathematical function that evaluates the effects of distortion introduced during compression and transmission.

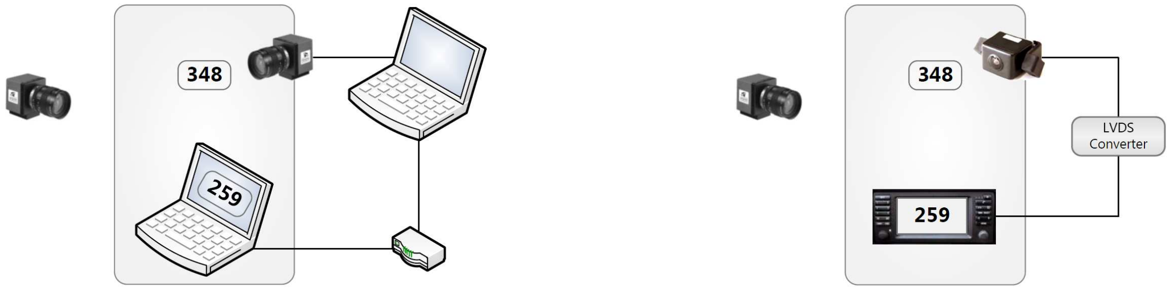
Table 5.1: Comparison of processing times and compression ratios for M-JPEG, MPEG-2, MPEG-4, and H.264/AVC. The abbreviations Conv., Enc., Transm., Dec., Compr. define Color Conversion, Encoding, Transmission, Decoding and Compression, respectively.



		PSNR	SSIM	Conv. Time (msec)	Enc. Time (msec)	Transm. Time (msec)	Dec. Time (msec)	Display Time (msec)	Video Bit Rate (kbit/s)	IP Bit Rate (kbit/s)	Compr. Ratio	Bits per Pixel
(a) Highway Sequence	M-JPEG	39.4	0.950	-	1.61	0.86	0.82	0.22	2341	2398	0.064	0.25
	MPEG-2	39.7	0.950	-	2.40	0.44	0.65	0.23	1378	1416	0.038	0.15
	MPEG-4 (L)	39.6	0.950	-	2.98	0.33	0.92	0.24	1124	1157	0.031	0.12
	MPEG-4 (X)	39.6	0.950	-	3.79	0.31	0.93	0.17	1048	1079	0.029	0.11
	H.264/AVC	40.0	0.950	-	14.05	0.27	1.95	0.24	953	984	0.026	0.10
(b) VGA Sequence	M-JPEG	37.0	0.950	0.38	5.40	3.20	2.46	1.01	8416	8589	0.076	0.91
	MPEG-2	37.5	0.950	0.44	7.68	1.57	1.89	0.62	4323	4417	0.039	0.47
	MPEG-4 (L)	37.3	0.950	0.42	10.12	1.28	2.44	0.58	3563	3642	0.032	0.39
	MPEG-4 (X)	37.3	0.950	0.49	13.62	1.25	2.37	0.60	3475	3554	0.031	0.38
	H.264/AVC	37.2	0.950	0.50	39.90	0.91	5.71	0.59	2636	2698	0.024	0.29
(c) VGA Sequence	M-JPEG	40.0	0.969	0.40	6.84	4.92	3.03	0.61	13384	13651	0.121	1.45
	MPEG-2	40.0	0.965	0.45	8.98	2.83	2.46	0.59	7885	8047	0.071	0.86
	MPEG-4 (L)	40.0	0.967	0.44	11.46	2.62	3.14	0.60	7312	7463	0.066	0.79
	MPEG-4 (X)	40.0	0.966	0.51	15.52	2.60	3.09	0.60	7208	7357	0.065	0.78
	H.264/AVC	40.0	0.966	0.50	45.91	2.02	7.47	0.58	5680	5799	0.051	0.62

End-to-end Delay

The delay of a camera transmission system including compression ("digital system") is measured with the testbed shown in Figure 5.2(a). It is compared with the delay of the current analogue rear-view camera system ("reference system" in Figure 5.2(b)). In each system, a counter showing the current time in milliseconds is captured by the respective camera. In the digital system, the data is compressed with Motion JPEG (M-JPEG), MPEG-2, MPEG-4 (X), and H.264/AVC. The compression parameters for the codecs are set such that a real-time compression was possible (even though, for x264 this still results in some skipped frames). The resulting video quality is not important in this context and has not been assessed. After compression, the data is transmitted via UDP/IP to the client computer over a 100 Mbit/s Ethernet link and a switch. At the client, as soon as a frame is completely received (no additional buffering), it was decompressed and displayed on the screen. The counter in the digital system is displayed on the same screen. In the reference system, the video signal is converted into a digital signal before being displayed on the screen of the headunit. In both systems, the current time on the counter and the time on the display are captured with a second camera and stored as individual images. The difference between the counter and display readings represents the end-to-end delay of the system. Even though not all of the readings were clear (overlapping of the counter values due to the display response times and the shutter opening time of the camera) around 50 "clear" values were evaluated for each system and codec with a standard deviation of about 8 ms. The outliers (set to 25% of the values for each test) were discarded. The results of the measurements with the most influencing values are shown in Figure 5.3. The average end-to-end delays are: 37.2 ms for the reference system, 84.4 ms for M-JPEG, 83.8 ms for MPEG-2, 92.5 ms for MPEG-4, and



(a) Digital system (using compression)

(b) Reference system

Figure 5.2: Testbed for the measurement of the end-to-end delay.

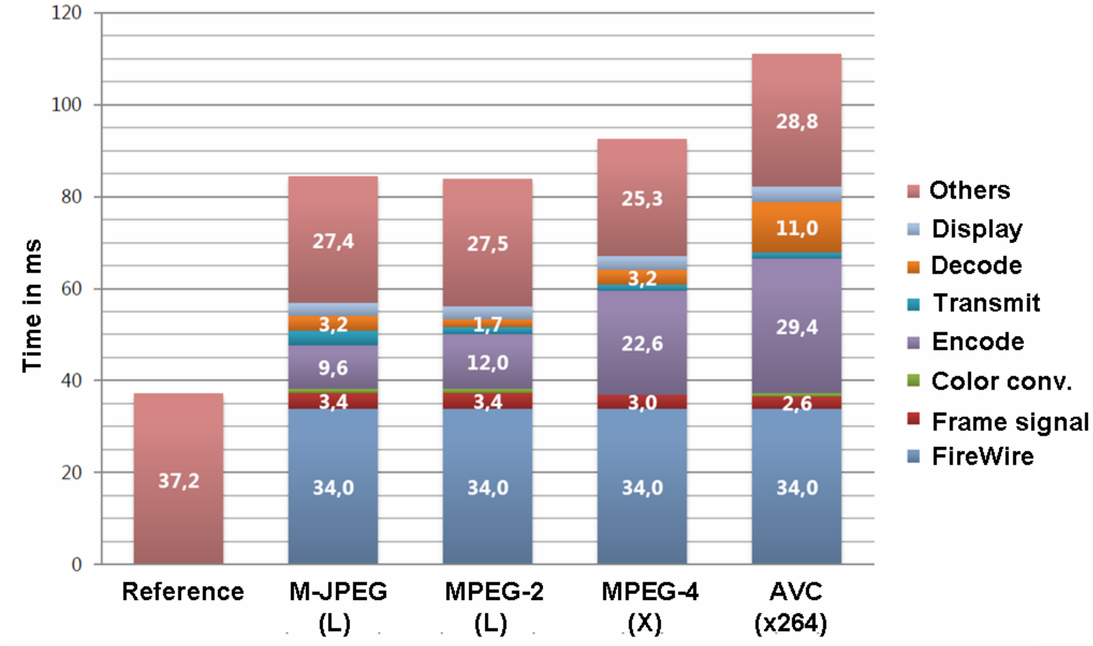


Figure 5.3: Comparison of the end-to-end delay values including delay components for all considered codecs and the analogue reference system.

110.7 ms for H.264/AVC. A large amount of the delay in the digital system is the result of the FireWire camera. In a FireWire system, the data of a frame is spread over the whole frame period for transmission (isochronous transfer). With an assumed short processing time in the camera and the frame time of 33.3 ms, the image transfer from the camera to the computer accounts for about 34 ms of the total delay.

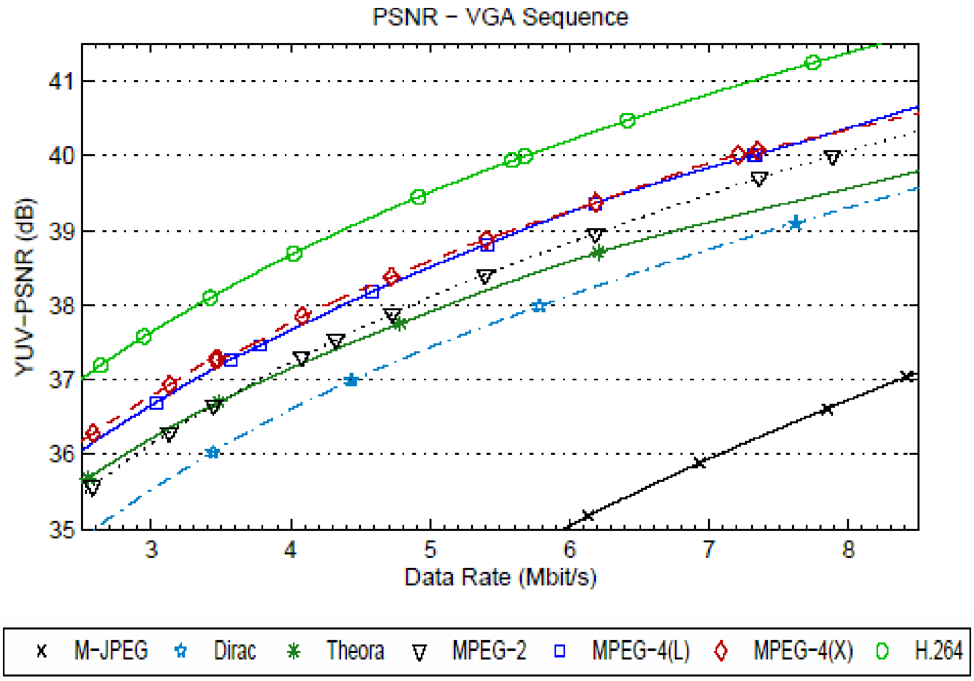
Delay components that could be measured are the time until the Linux kernel signals the availability of a new frame and the processing started (frame signal), the time for the YUV subsampling (color conversion), the encoding, transmission, and decoding times, as well as the time that elapses while copying the frame to the graphics buffer and issuing the command to update the display. These values are measured through software scripts that output the different times such as the encoding and decoding times from the codec libraries, e.g., from the libavcodec. The resulting delay contributions (assigned with ”Others” in Figure 5.3) that could not be explicitly measured mainly comprise the time from displaying the actual counter until it is captured by the camera and the time from issuing the display update command for the delay counter until the frame is actually displayed (depending on the display refresh rate, amongst others).

Consequently, in a fully IP-based in-vehicle network, the major delay component ”FireWire” will be eliminated when integrating the camera sensor and codec into one chip as will be explained in Section 5.3. By further adaptations to the in-vehicle requirements, the delay component ”Others” can also be significantly reduced. Thus, the end-to-end delay requirement from Section 2.4.1 can be fulfilled with the software-based codecs.

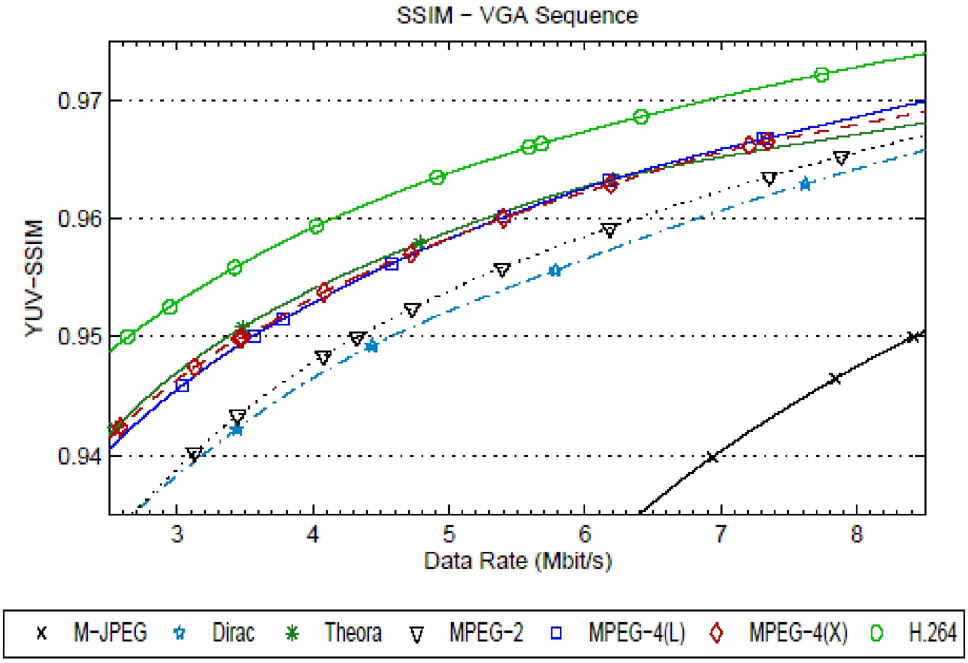
Video Quality

In the following, the video quality obtained from the considered compression algorithms is assessed for different bit rates using the test sequences from Section 5.1.1. For analysis, the two open source codecs Theora and Dirac introduced in Section 2.3 are also considered besides the standard codecs. The PSNR and SSIM values of the VGA sequence are shown in Figures 5.4(a) and 5.4(b) for bit rates of 2.5 to 8.5 Mbit/s. The quality results of the Highway sequence are similar to the results of the VGA sequence and are therefore not presented here. To obtain an SSIM index of 0.95 for the VGA sequence, the data rate of an M-JPEG coded sequence has to be more than three times higher than with H.264/AVC and more than twice in comparison to the MPEG-4 coded sequence. The difference is higher in the lower bit rate range. Yet, in our analysis the superiority of H.264/AVC in terms of quality is not as clear as it might have been expected. This is due to several reasons. First, B-frames were not used for prediction due to the stringent delay requirements. Also, a rather short I-frame interval of 15 frames (see Section 2.4.1) has been used. Second, due to the low complexity requirements, the motion estimation process was carried out with algorithms that offer a trade-off between complexity and efficiency. Third, some advanced coding tools, such as CABAC for H.264/AVC, were not used due to restrictions of the Baseline profile. Also, parameters such as the content of video sequences and artifacts impact the motion estimation efficiency.

Although H.264/AVC is superior to all other compression systems, its significantly higher processing requirements for encoding and decoding do not justify its employment as software codec in the discussed real-time scenario. As expected, the performance of MPEG-4 lies in between that of MPEG-2 and H.264/AVC. The two MPEG-4 implementations show very similar results. While the wavelet-based Dirac codec performs worse than the other compression schemes and its encoding and decoding times are about a factor ten higher than DCT-based compression algorithms, the performance of Theora is almost equal to MPEG-4 in terms of perceived quality.



(a) PSNR values



(b) SSIM values

Figure 5.4: Video Quality Evaluation: PSNR and SSIM values for the VGA sequence and data rates from 2.5 to 8.5 Mbit/s.

Its encoding time is in the range of the x264 implementation of H.264/AVC. Since the current release is only an alpha version, Theora shows a high potential as a free alternative to MPEG-4 in the future. However, due to their insufficient timing performance, Theora and Dirac have not been analyzed further in the present work.

5.2 Influence of Video Compression on Driver Assistance Image Processing Algorithms

In the following, the influence of video compression on image processing algorithms is analyzed for indirect image-based driver assistance services introduced in Section 2.3.2. A similar analysis has been performed in [99] to investigate the influence of video compression on face detection algorithms in distributed surveillance camera systems. Their analysis method to identify the spot up to which the video quality can be reduced, without affecting the accuracy of the surveillance task has been used for the analysis in this work. Image processing in the automotive domain typically refers to the task of object recognition. Objects are identified and their types within a picture or a video sequence is determined [94]. To achieve this goal, the so called Regions Of Interest (ROI) are identified by globally searching images for objects or by evaluating data from external sources. In the car, the ROI are commonly selected by external sources such as radar or 3D-cameras. Accordingly, the next processing steps, feature extraction and classification are considered in this work.

5.2.1 System Description

Also in this section, the wired network is considered for the analysis. Thus, the effect of lossy compression on image processing algorithms is determined without any external influences. Two commonly used object extractors, the Edge Histogram Descriptor (EHD) of MPEG-7 [109] and Histogram of Oriented Gradients (HOG) [92] have been chosen for the following analysis. They are used in the spacial domain of images, i.e., they are applied to every single video frame. The Optical Flow object extractor [66], [24] has been additionally used to take the temporal domain of video sequences into account. The Support Vector Machine (SVM), a supervised learning method, has been used to classify objects after feature extraction as this method achieves good classification results [33]. SVM is trained with features of well-known objects before the classification process. The two video codecs, MPEG-4 (DCT-based) and M-JPEG 2000 (DWT-based) realized by the libavcodec [41] and the OpenJPEG library [29] have been used as candidates for DCT-based and DWT-based compression algorithms.

Test Sequences: Four different video sequences have been recorded with the camera (VRMagic VRmC-12/BW PRO) of an experimental vehicle. They have a resolution of 752×480 pixels and a frame rate of 25 frames/s. Video frame resolutions 752×480 pixels are defined as full size while 376×240 pixel frames are called quarter size. The frame size of the original video is reduced by using a simple linear filtering and subsampling approach. All images are gray-scaled.

Figure 5.5 shows the considered test sequences. The amount of motion in sequences c) and d) is higher than in a) and b).

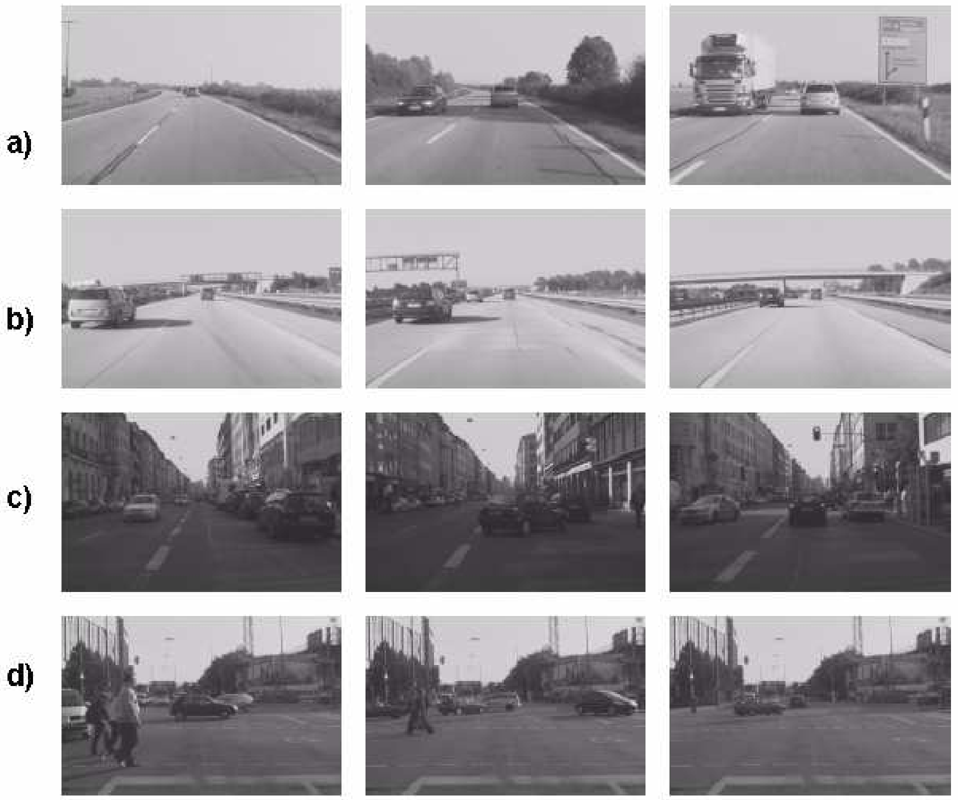


Figure 5.5: Example frames of the a) Country Road b) Highway c) City and d) Intersection video sequences.

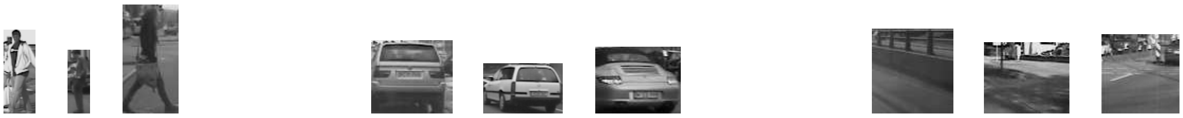
1. Country Road: It is a typical driving scene on a country road (Figure 5.5(a)). A car is followed. Once in a while, oncoming traffic (cars, trucks, vans), traffic signs and bushes appear. Much of the sky and free countryside can be seen. The video sequence consists of 874 frames and a total of 1067 objects have been labeled.

2. Highway: This sequence has been taken on a two lane highway (Figure 5.5(b)). A car is followed and overtaken several times by other cars. Several bridges (with traffic) and traffic signs are passed. Much of the sky (but less than in the Country Road sequence) can be seen. The sides are heavily planted with bushes and other obstacles such as signs. The video sequence consists of 475 frames and a total of 998 objects have been labeled.

3. City: The city sequence shows a typical scenario in a darker street (Figure 5.5(c)). A lot of oncoming traffic and parking cars, trucks, buses, vans, (motor-) bikes and pedestrians appear. The experimental vehicle slows down to wait for a car pulling out of a parking space. It then follows the car up to a traffic light. There is very little sky, a lot of buildings and houses can be seen on both sides of the street. The video sequence consists of 640 frames and a total of 564 objects have been labeled.

4. Intersection: The intersection sequence has been taken at a red traffic light (Figure 5.5(d)). A lot of crossing traffic (cars, bicycles) and pedestrians appear. The background of this video is static, as the experimental vehicle is not moving. Some sky as well as several walls and a building can be seen. The video sequence consists of 801 frames and a total of 2543 objects have been labeled.

Training Data: The training of the SVM has been performed with 1748 images of three different object classes: car, pedestrian and non-object. The images are completely independent from the video sequences that have been analyzed as they were extracted from other sequences. The car class contains cars from the rear and the rear-side while the pedestrian class contains people walking or standing. The non-object class contains objects that are not defined in the object class data set like trees. Figure 5.6 shows some typical examples of the training data set.



a) pedestrians b) cars c) non-objects

Figure 5.6: Typical examples for training data.

Ground Truth: In order to evaluate and compare the object classification results, the so called ground truth should be specified for the test sequences. To generate the ground truth, a person identifies all objects within a video sequence and labels them according to their type. In this work, an existing labeling tool has been applied. Figure 5.7 shows the tool while labeling the highway sequence. As only training data for cars, pedestrians and non-objects were available, only those three objects are classified in the test sequences. Other objects like trucks and signs have not been taken into account for analysis.

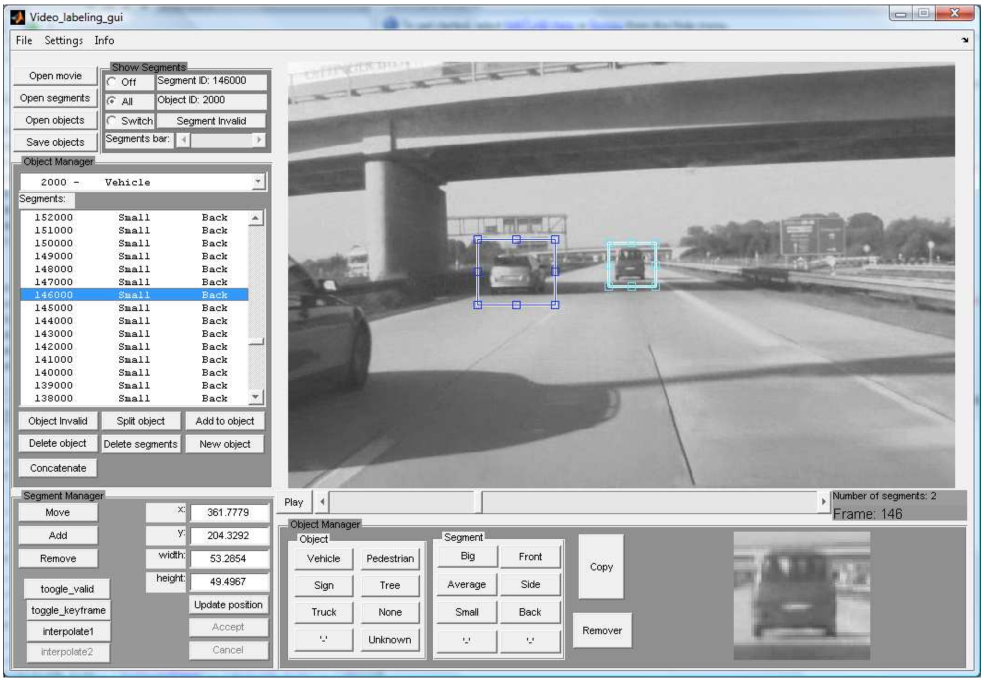
Applied Codec Parameters:
MPEG-4: Experiments have been carried out with the libavcodec sofware codec mentioned in Section 5.1 and the preset target bit rates b

$$b_i = \left\{ \begin{array}{c} 10, 50, 100, 200, 300, 400, 500, 600, \\ 700, 750, 800, 850, 900, 950, \ldots \\ \ldots, 1000, 1100, 1200, 1300, 1400, 1600, \\ 1800, 2000, 2500, 3000, 3500, 4000, 5000, 6000 \end{array} \right\}$$

given in kbit/s, where $i = 1, \ldots, 28$ is the number of the actual test run. Moreover, two main parameter sets have been defined for the MPEG-4 compression as follows.

- Parameter set 1: Constant bit rate
  To emulate an almost constant bit rate, the maximum allowed bit rate tolerance has been set to 10 % of the target bit rate of the actual simulation run $b_{target,i}$ so that $0.9b_i \leq b_{target,i} \leq 1.1b_i$.

- Parameter set 2: Variable bit rate
  This parameter set configures the allowed bit rate tolerance to infinity, so that the rate control mechanism generates a VBR video sequence.



Video_labeling_gui

Show Segments

○ Off Segment ID: 146000

● All Object ID: 2000

○ Switch Segment Invalid

Object Manager

2000 - Vehicle

Segments:

152000	Small	Back
151000	Small	Back
150000	Small	Back
149000	Small	Back
148000	Small	Back
147000	Small	Back
146000	Small	Back
145000	Small	Back
144000	Small	Back
143000	Small	Back
142000	Small	Back
141000	Small	Back
140000	Small	Back
139000	Small	Back
138000	Small	Back

Segment Manager

x: 361.7779

y: 204.3292

width: 53.2854

height: 49.4967

Number of segments: 2

Frame: 146

Figure 5.7: GUI of the video labeling tool.

Both parameter sets include the MPEG-4 parameters Four-Motion-Vector (enabled), Data Partitioning (enabled), Quarter Pixel Motion Estimation (enabled), MPEG Quantizer Set: 1 and a minimum key frame interval of 15 [5]. The constant compression ratio is related to the quantizer values $Q = 31$ for lower target bit rates and $Q = 2$ for higher target bit rates. Thus, the rate control algorithm of the MPEG-4 encoder chooses $Q = 2$ as maximum and therefore does not produce video sequences with the highest possible quality. The MPEG-4 standard, however, allows quantizer values between 1 and 31. While the difference between $Q = 1$ and $Q = 2$ can not be distinguished by human eye, it is recognizable by algorithms and has therefore been considered for the analysis with image processing systems.

M-JPEG 2000: The openJPEG2000 encoder has been configured with the DWT decompositions: 6, irreversible compression using Irreversible Color Transformation (ICT) and Daubechie 9-7 filters [55].

5.2.2 Applied Metrics

Besides the video quality metrics PSNR and SSIM from Section 5.1, the following metrics have been used [10].

Object Classification Rate: Similar to [99], the object classification rate R has been defined here as the ratio between the correctly detected objects O and the overall recognizable objects within a video sequence L:

$$R = \frac{O}{L} \tag{5.1}$$

The classification rate R has been evaluated to find the minimum and maximum allowed compression ratios for each video sequence.

Average Mean Deviation: In order to be able to evaluate the changes of features that are extracted, the Average Mean Deviation (AMD) of a feature vector x of length l is defined for an encoded video sequence with k frames as

$$AMD = \frac{1}{k}\sum_{i=1}^{k}\bar{D_i} \tag{5.2}$$

where the mean deviation $\bar{D_i}$ of each frame $i = 1, \ldots, k$

$$\bar{D_i} = \frac{1}{l}\sum_{n=1}^{l}\frac{x_n - y_n}{y_n} \tag{5.3}$$

and y denotes the feature vector within a frame in the uncompressed video sequence as x defines it in the compressed video sequence.

Number of Vectors: Due to the nature of the optical flow determining the ground truth is rather difficult. Thus, in this work, only the computed optical flow vectors for each frame in the original video sequence have been compared with those in the compressed video sequence. Accordingly, vectors existing in both video sequences have been defined and counted to evaluate the influence of compression on the optical flow.

5.2.3 Comparison Results

A framework has been implemented in MATLAB to perform the following processing steps for each video sequence and to evaluate the results.

- Apply feature extraction (EHD and HOG) and object classification (SVM) to the original (uncompressed) data.
- Calculate optical flow for the original (uncompressed) data.
- Compare the classification results with the ground truth.
- Carry out MPEG-4 encoding and decoding by using the target bit rates from Section 5.2.1.
- Compute the compression ratio.
- Apply M-JPEG 2000 encoding and decoding to the original video sequence configured by the MPEG-4 compression ratios[3].

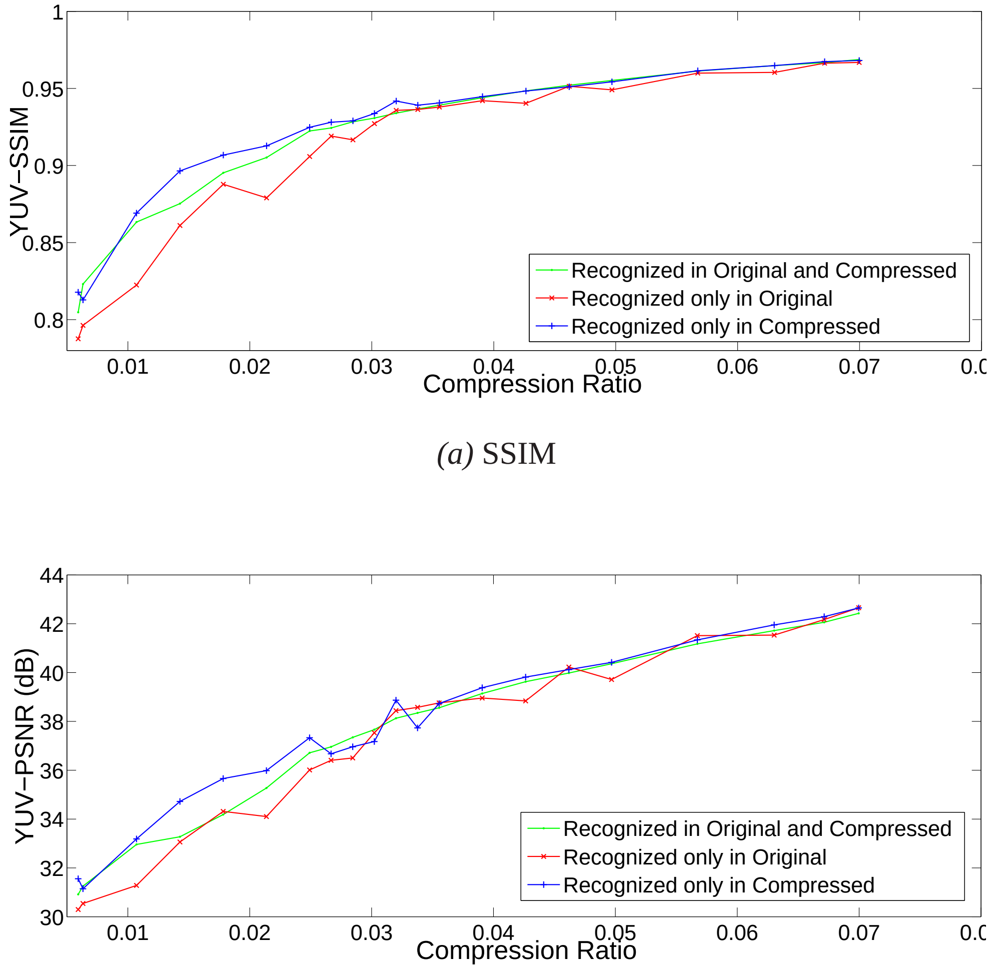
[3] In order to compare the two compression algorithms, MPEG-4 and M-JPEG 2000, each video sequence has been first compressed by the MPEG-4 codec to reach a certain bit rate. The resulting MPEG-4 compression ratio has been determined and then used as an input parameter for the M-JPEG 2000 codec.

- Apply feature extraction (EHD and HOG) and object classification (SVM) to the decoded video sequences.
- Calculate optical flow for the decoded data.
- Compare the classification results with the ground truth and the original video classification results.

The most significant results have been selected for discussion. Due to similar implications of HOG and EHD on compressed video sequences, the results of HOG are exemplarily explained in the following.

The gray-scaled video sequences in this work lead to lower bit rates than the color video sequences from Section 5.1 due to the lack of color components. For example, Figure 5.8 shows a compression ratio upper bound of 0.07 for PSNR = 42 dB and SSIM = 0.96 in the City video sequence (Figure 5.5(c)) corresponding to a data rate of 5 Mbit/s. The PSNR and SSIM values shown in Figure 5.8 are computed for each recognized object in the original and in the compressed video sequences. This means that instead of computing the PSNR and SSIM for a complete video frame as done in Section 5.1, they have been computed for objects of a frame such as those shown in Figure 5.6. However, the results from Figure 5.8 do not indicate any dependency between the PSNR and SSIM quality metrics and the classification results, i.e., recognized objects in the original and compressed sequences. The exact behavior of the codec is shown through the compression ratio. For example, as shown in Figures 5.9(a) and 5.9(b), the codec does not function for compression ratios higher than 0.03 in the case of the Country Road video sequence (Figure 5.5(a)). Figures 5.10(a) and 5.10(b) depict a more detailed view on the classified objects based on the test sequence ”City” from Figure 5.5(c). As mentioned before, no relationship between the object classification rate, the segment size, the PSNR, and the SSIM can be observed. Figures 5.10(a) and 5.10(b) show that MPEG-4 compression negatively influences the classification rate for compression ratios ≤ 0.03, while M-JPEG 2000 compression leads to higher classification rates. For compression ratios very close to zero, the quantizer values are out of the codec acceptance range. Consequently, the related object classification rates for nearly zero compression ratios are invalid. The object classification rate results can be explained as follows. Since the HOG feature extraction heavily relies on the existence of edges, the extensive spatial filtering by quantizing the DCT coefficients in MPEG-4 leads to less features and thus, less objects are classified correctly. Additionally, by introducing image distortions more objects are classified as non-objects and the object classification rate is decreased. In the case of M-JPEG 2000, the DWT smooths (blurs) images. By using a simple gradient filter $[-1, 0, 1]$, uniformly colored edges are not detected. Yet, by smoothing the edges (and thus varying the gray scale values), the uniform distribution is lost and more gradients are generated leading to a higher weighted direction binning. Thus, smoothed edges support the classification of objects with HOG. However, smoothing only effects lower bit rates, i.e., lower compression ratios, as for higher bit rates a certain amount of spatial details needs to be preserved and is not smoothed by the DWT.

The analysis of optical flow vectors have, in some cases shown an increase of the number of vectors in the compressed video sequences. The reason is the lower number of uniform planes within the frames of the compressed video sequence that leads to a higher number of vectors by the Census transform. Moreover, the additional vectors are concentrated within the ROI as shown in Figure 5.11. The number of common vectors in the original and compressed videos

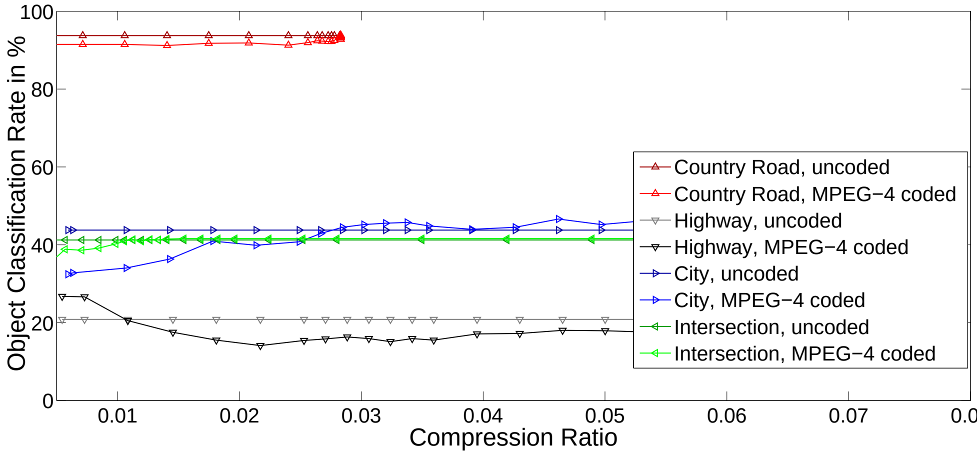


(a) SSIM

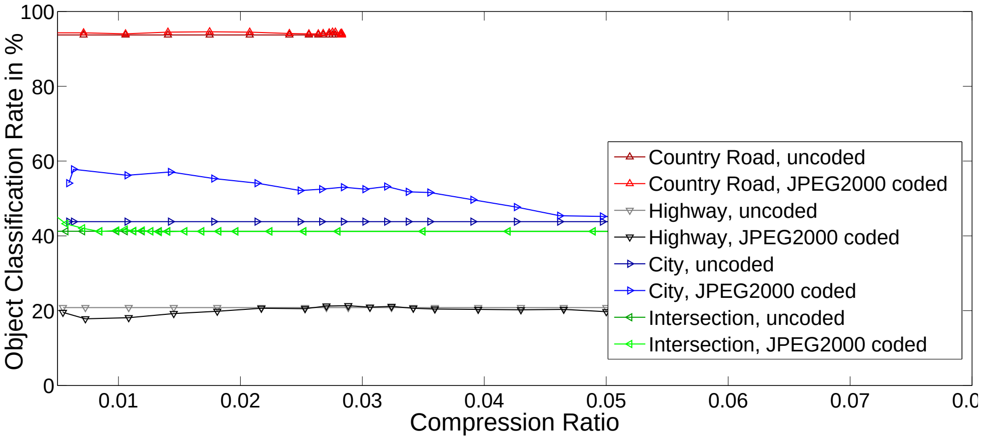
(b) PSNR

Figure 5.8: SSIM and PSNR values of the recognized objects by HOG for different compression ratios (Test sequence "City", parameter set 1, segment-based, full size, MPEG-4 compression.).

is low because of their definition to be absolutely equal. Consequently, vectors that are shifted only by a few pixels in the compressed video are not considered to be equal. Visual validation and vector length histograms have been applied for the optical flow analysis.
To conclude, the results have shown that lossy video compression can be applied besides driver assistance image processing algorithms without any significant quality degradation if a proper codec and feature extraction pair is selected. The influence of video compression on driver assistance functions strongly depends on the applied compression and image processing algorithms, but also on the video sequence itself. Accordingly, no conclusion can be generalized for different codecs and image processing algorithms.



(a) MPEG-4 video compression

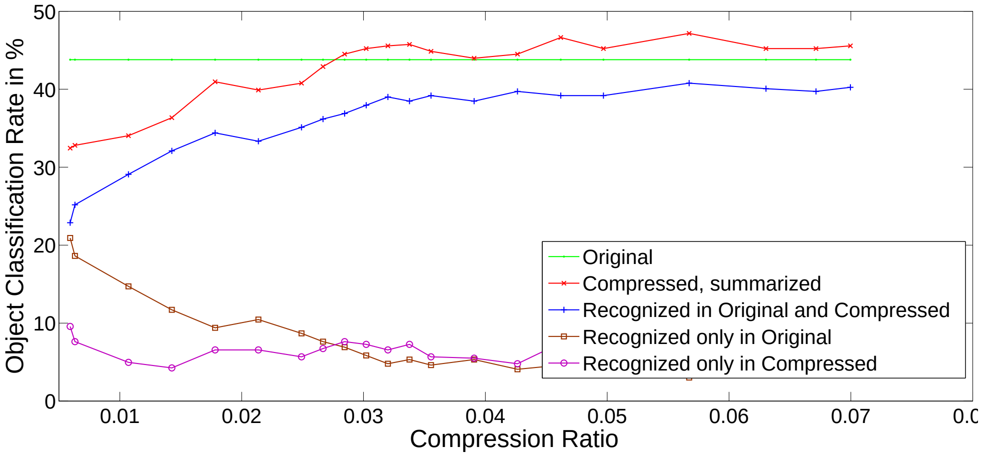


(b) M-JPEG 2000 video compression

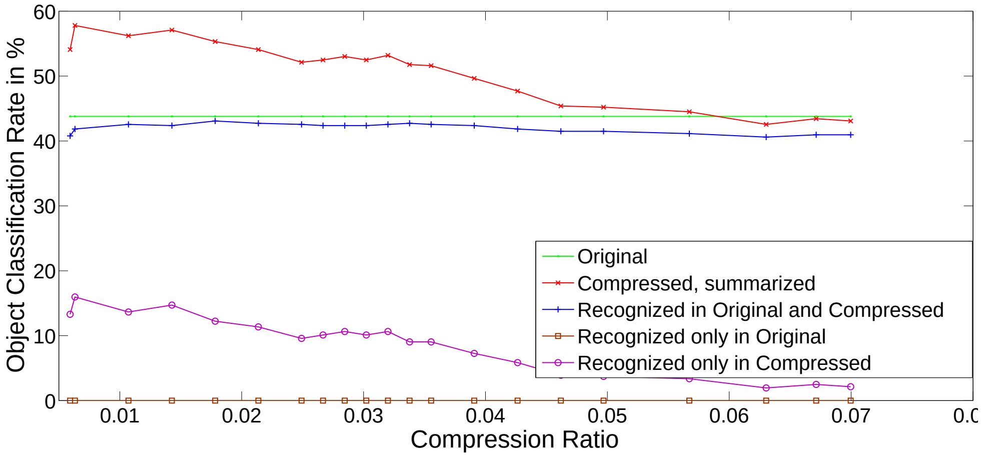
Figure 5.9: Influence of video compression on object classification using HOG with parameter set 1, segment-based and full sized frames.

5.3 Hardware Implementation Concepts for IP cameras with Video Codecs in the Car

According to [5], designing an IP camera with video codecs for automotive applications can basically follow two design methods. The first one would be a highly customized solution to fulfill the requirements of the automotive sector while the second one is based on the utilization of hardware and software toolboxes of the consumer electronic industry. The latter one should be set accordingly in order to meet the automotive requirements.



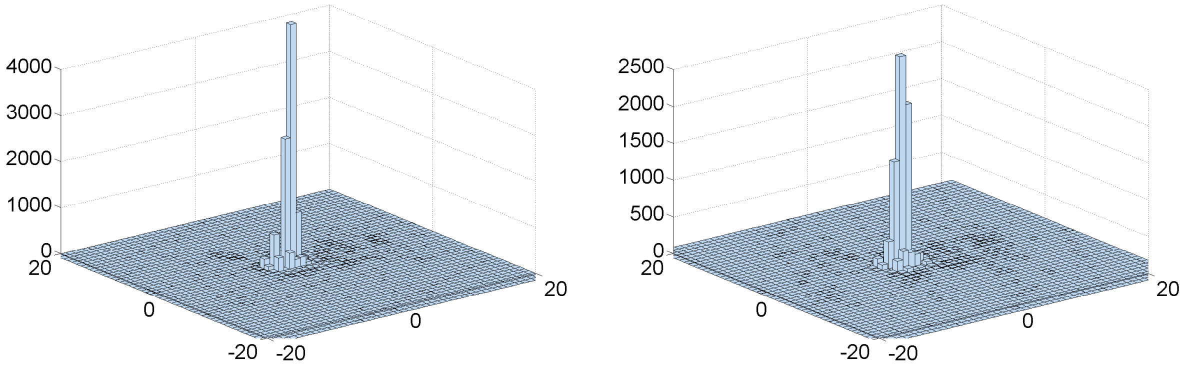
(a) MPEG-4 video compression



(b) M-JPEG 2000 video compression

Figure 5.10: Influence of video compression on object classification using HOG with parameter set 1, segment-based and full sized frames for compressed and uncompressed test sequence "City".

The customized solution would meet the automotive requirements, but can be more cost intensive and inflexible while the utilization of consumer electronic solutions would be more promising in terms of flexibility and low cost thanks to the economy of scale caused by the consumer electronic market. However, meeting the automotive requirements with the consumer electronic tools is challenging. Both realization concepts with some examples are briefly described in the following.



(a) Original video

(b) MPEG-4 coded video

Figure 5.11: Histogram of optical flow vectors for a selected frame in original 5.11(a) and MPEG-4 coded 5.11(b) "City" video sequences with parameter set 1, full size and compression ratio 0.01 (corresponding to 750 kbit/s target bit rate).

5.3.1 The Customized Solution - FPGA/ASIC Implementation

In the case of customized solution, the compression algorithm, the communication protocols such as RTP/UDP/IP, the Ethernet MAC, and if required the image processing algorithms should be integrated in the FPGA/ASIC design. Most of the mentioned components which have to be integrated in the FPGA/ASIC are already available as IP cores. Important electronic components which are left out of the FPGA/ASIC are the power supply, voltage control, the Ethernet PHY, and the camera sensor. An example for the CMOS camera sensor which is able to deliver a video sequence with VGA resolution and 30 frames/s is the LM9628 sensor from National Semiconductor [91]. Generally, the FPGA/ASIC is connected via an 8 to 12 bit data bus and is controlled via an I^2C bus.

5.3.2 Solutions from the Consumer Electronic Industry

The application of consumer electronic hardware and software toolboxes leads to two different solutions.

Multimedia Processor Solution

A multimedia processor-based solution provides a higher flexibility than an FPGA/ASIC-based solution. The current multimedia processors are equipped with all necessary features for building an IP-based camera system. They include MPEG-4 and H.264 Baseline and main profile codecs for video compression with a VGA resolution and 30 frames/s transmission rate, an integrated ARM processor for applying the TCP/IP or UDP/IP stacks and implementing application layer protocols such as RTP and RTSP, and an Ethernet MAC for the network connection.
The integrated system on chip design of the multimedia processors offers the possibility to reach

very small dimensions of the PCB (printed circuit board) layout which is also a very important requirement for an in-vehicle camera system. Similar to the ASIC solution, electronic components that are left out of the processor are the camera sensor, power supply, voltage control and the Ethernet PHY. Several well-known multimedia processor manufacturers have already presented promising solutions to the market.

CMOS On-Chip Compression Solution

A further interesting solution is the utilization of a CMOS camera sensor with an integrated video codec. So far, it has only been possible to integrate the M-JPEG compression algorithm because of its low complexity compared to the MPEG family compression algorithms. An example is the VS6724 single-chip camera module from ST microelectronics [112]. This kind of sensor provides a M-JPEG video compression up to SVGA (800x600 pixel) resolution. It also integrates digital image processing functions like lens shading correction, sharpening etc. The sensor can be controlled by an I^2C interface and provides an 8-bit parallel video interface for the video data. For a complete IP camera implementation, a processor with sufficient processing power such as the ARM9 processor should be added to the sensor in order to packetize and send the compressed images captured by the sensor via an Ethernet interface. Assuming that the Ethernet MAC is also provided by the applied processor, the remaining electronic components to create an IP camera will be similar to the multimedia processor solution mentioned before.

5.4 Summary

The performance of software video codecs has been analyzed to introduce video compression in driver assistance camera systems of future IP-based cars. The MPEG-4 (Part 2) compression algorithm provides the best trade-off between the achieved quality and the computational complexity. It fulfills the strict delay, complexity and quality requirements of in-vehicle camera systems while at the same time, it provides an adequate compression ratio of 0.066 corresponding to a mean bit rate of 7.4 Mbit/s for the considered color VGA sequence with a PSNR value of 40 dB and an SSIM value of 0.967. A compression ratio of 0.032 delivers a lower bit rate of 3.6 Mbit/s on the transmission link for PSNR and SSIM values of 37.3 dB and 0.95, respectively. The influence of video compression on typical driver assistance image processing algorithms has also been analyzed. Commonly used object extraction and classification algorithms from the driver assistance domain have been taken into account for analysis. Compression ratios have been identified that do not affect the driver assistance functions, but rather improve their performance. Among the DCT-based MPEG-4 and the DWT-based M-JPEG 2000 algorithms, M-JPEG 2000 leads to higher object classification rates for HOG in most of the cases. The discrete Wavelet transform smooths the uniform planes in video frames and varies the gray scale values. Thus, more gradients are generated leading to a higher classification rate with the HOG object extraction. MPEG-4, on the other hand, supports EHD binning by filtering details and sharpening edges in images of a video sequence. However, a general conclusion cannot be made, since the classification rate strongly depends on the codecs, image processing algorithms and also the amount of motion in video sequences. In order to further improve the compression performance and be able to apply more complex codecs such as the widespread H.264/AVC in the car, several hardware implementation concepts for IP cameras with integrated video codecs have been presented and discussed.

6 Conclusion and Outlook

The number of electronic control units and automotive networks for in-vehicle communication is continuously growing in today's premium cars. Thus, the in-vehicle network architecture has become inflexible, complex and expensive. It will be even more complex and costly in the near future due to the growing number of applications, especially in the driver assistance and multimedia domains. Accordingly, traditional automotive network technologies with limited transmission capacity will no longer fulfill the in-vehicle communication requirements. In order to cope with the growing demands, a new network system for in-vehicle communication is required.

In this work, a heterogeneous IP-based network architecture has been proposed for in-vehicle communication. According to car manufacturers' requirement, all in-vehicle applications are covered by the proposed network architecture except for FlexRay applications. A static network dimensioning method based on the Network Calculus theory has been presented to compute the required network resources in the car. Simulation models have been developed to verify the worst case analytical results for two selected network scenarios. Traffic shaping as a method to reduce the required network resources has been investigated in variable bit rate video sources and in the interconnected network elements. A novel traffic shaping algorithm has been introduced that outperforms the existing traffic shapers in terms of resource usage under certain constraints. Appropriate video compression algorithms have been identified for simultaneous transmission of several video streams over one Fast-Ethernet link and investigated for their influence on image processing algorithms from the driver assistance domain. In the following, the major results of this dissertation are summarized.

Heterogeneous IP-based Network Architecture for In-Vehicle Traffic

By considering the QoS requirements of communicating devices, the in-vehicle traffic has been subdivided into four classes of real-time control data, real-time audio and video streams, multimedia and best effort data. Mathematical models from the literature [108] have been adapted to represent the in-vehicle traffic (Section 2.4.1). A novel network architecture based on IP has been designed for in-vehicle communication to enable unicast, multicast and broadcast transmissions in Chapter 3. Full-duplex switched Fast-Ethernet and WLAN have been identified as appropriate network technologies beneath of IP to realize the wired core and wireless peripheral networks in the car. Double star and unidirectional ring network topologies have been selected as candidate parts of the future overall in-vehicle network. A QoS-API has been introduced that statically assigns priority levels to different applications and maps them to IP and Ethernet packets. QoS-aware switches and access points thus forward packets according to their importance. QoS is guaranteed when the network load is below the link capacity.

Static Network Dimensioning

In overload situations, packets are delayed due to head-of-line blocking, scheduling and queuing strategies in the network elements before being transmitted. Thus, QoS cannot be guaranteed.

The car is a closed network system where all applications and their transmission scenarios are known a priori. Accordingly, all required resources for a QoS-aware data transmission can be computed before the network startup. In Chapter 3, a static network dimensioning method based on the Network Calculus theory has been presented to compute the required network resources in the car before data transmission is initiated. Hence, lost and late packets due to buffer overflow and large queues can be prevented in overload situations by providing the required resources throughout the network. Simulation models have been implemented to verify the worst case analytical results for a realistic transmission scenario. The results have shown a large resource consumption in the wired in-vehicle network while the transmission over the wireless link cannot be established due to channel saturation. The large instantaneous bursts from variable bit rate video sources are the cause of the afore mentioned high traffic load in the considered transmission scenario.

Resource Reduction by Means of Traffic Shaping

Traffic shapers conform data streams to pre-defined characteristics and regulate their burstiness. By lowering traffic bursts, required resources in the network are reduced. In order to determine the effectiveness of traffic shaping, its application in variable bit rate video sources and in the interconnected switches and access point has been analyzed. Beside the well-known Leaky Bucket and Token Bucket shaping algorithms, a novel traffic shaping algorithm called Simple Traffic Smoother (STS) has been introduced and analyzed. STS reduces the peak transmission rate by simply sending the packets of each picture frame with a certain time span between them instead of sending them very close to each other as a burst to the network. For this purpose, it stores all packets of a picture frame received within a predefined smoothing interval and sends them to the network in the next interval equally spaced in time. Analyses have shown that STS provides the best QoS performance with the lowest resource usage in the described double star topology when applied to slightly desynchronized video sources. The application of STS is not limited to the in-vehicle network. It can be applied to any other video transmitting network system which underlines its large application field. Reshaping in the interconnected network elements turned out to be useful for more than two cascaded switches and performs the best when realized on a per stream basis. By slightly relaxing the QoS requirements for video sources the amount of needed resources can be further reduced. Thus, a resource-efficient and QoS-aware network architecture has been defined for future in-vehicle communication. The network design and configuration instructions explained for the two selected topologies, double star and unidirectional ring, have been generalized for any arbitrary in-vehicle network topology and transmission scenario as a user guideline in Chapter 4.

A prototypical IP-based network with the double star topology has been implemented in a BMW 530d to show that the proposed network architecture works properly in a real car. Experimental results presented in Chapter 4 confirm the flawless performance of the resource-efficient IP-based network system in any network load situation.

Video Compression in Driver Assistance Camera Systems

The number of camera systems is continuously growing in the driver assistance domain. Camera images are either directly shown to the driver to enlarge his surrounding view (direct image-based services) or are first investigated by certain image processing algorithms to obtain specific information for driver assistance (indirect image-based services). In both cases, the quality of images plays a crucial role in the success of the driver assistance service. In the proposed IP-based network architecture covering all video-based applications in the car, the simultaneous transmission of several video streams over one Fast-Ethernet or WLAN link is not feasible without video compression. Accordingly, in Chapter 5, video compression algorithms have been investigated for both, direct and indirect image-based driver assistance services. The MPEG-4 software codec has been identified to fulfill the strict QoS requirements of driver assistance camera systems. In order to improve the compression performance, hardware implementation concepts for IP cameras with integrated video codecs have been presented and discussed. In the context of indirect image-based services, typical object extraction and classification algorithms from the automotive sector have been applied to the compressed video sequences. Results have shown that for each compression algorithm, there is a compression ratio range where video compression does not affect the object extraction and classification, but rather improves them. However, the extraction and classification performance of driver assistance algorithms strongly depend on the codecs, image processing algorithms and also the amount of motion in video sequences. Accordingly, no relationships could be generalized.

Outlook

During this work, several interesting subjects for further research have been identified, among which the following are of particular interest.

In Chapter 3, we proposed an IP-based network architecture and selected the full-duplex switched Fast-Ethernet (IEEE 802.3) and WLAN (IEEE 802.11g) as physical transmission media due to their wide availability in the market. However, the demand for new applications in the car will grow and more transmission resources will be needed. Broadband network technologies with higher transmission rates such the Ultra Wide Band or IEEE 802.11n technology for wireless communication that were not ready at the time this work was done, should be investigated for their adaptability for in-vehicle communication as soon as they are ready. Future shielded or unshielded Gigabit-Ethernet links should be investigated for their electromagnetic radiation and cost for wired communication. Also, the Audio Video Bridging technology mentioned in Chapter 2 represents a very interesting solution by integrating all required QoS mechanisms in the hardware and should be considered as an alternative network technology for in-vehicle communication as soon as it is available in the market. The in-vehicle wireless channel has been measured for an empty car. In order to have more realistic values, different scenarios, e.g., the car with several passengers or with other active wireless devices such as Bluetooth mobile phones should be analyzed.

The network dimensioning method designed for in-vehicle communication in this work is a deterministic method computing resource upper bounds for a worst case transmission scenario. By relaxing the constraints through simulations, more realistic resource usage values could be obtained. Interesting research work can still be done on statistical resource planning in terms of analytical models. Thus, more realistic results can be obtained directly from the analytical model which is particularly interesting for variable bit rate video streams.

The traffic shaping mechanism has been applied in this work mainly to reduce large data bursts from the variable bit rate video sources and thus save network resources. However, if larger link capacities are provided in the network, e.g., by using Gigabit-Ethernet, a constant bit rate compression can be applied to the video sources, which leads to a lower compression efficiency but prevents the traffic bursts caused by the rate variability. Thus, the focus of traffic shaping will change to, for example, an isochronous transmission as done in the Audio Video Bridging standard.

The presented video compression analyses were all based on software codecs. Accordingly, complex and powerful compression algorithms like H.264/AVC did not function well in delay analysis and failed the performance tests. Therefore, hardware implementations are necessary to optimize the codec behavior in the car and to meet the strict delay requirements. An adequate approach should be investigated for the future in-vehicle camera systems based on the mentioned hardware implementation concepts from Chapter 5.

7 Abbreviations and Acronyms

AMD Average Mean Deviation

AP Access Point

API Application Programming Interface

ARTS Advanced Real-Time Simple Profile

ASO Arbitrary Slice Ordering

ASP Advanced Simple Profile

AVB Audio/Video Bridging

CABAC Context-Adaptive Binary Arithmetic Coding

CAN Controller Area Network

CAVLC Context-Adaptive Variable Length Coding

CBR Constant Bit Rate

CE Consumer Electronic

CPU Central Processing Unit

CVBS Color Video Blanking Signal

CW Contention Window

DCT Discrete Cosine Transform

DIFS Distributed InterFrame Space

DP Data Partitioning

DWT Discrete Wavelet Transform

ECU Electronic Control Unit

EHD Edge Histogram Descriptor

EMC Electro Magnetic Compatibility

ET Event Triggered

F-ARIMA Fractional-AutoRegressive Integrated Moving Average

FEC Forward Error Correction

FMO Flexible Macroblock Ordering

FTP File Transfer Protocol

GMC Global Motion Compensation

GoP Group of Pictures

HMI Human Machine Interface

HOG Histogram of Oriented Gradients

IP Internet Protocol

ISO International Organization for Standardization

LAN Local Area Network

LB Leaky Bucket

LIN Local Interconnect Network

LLDP Logical Link Discovery Protocol

LRD Long Range Dependence

LVDS Low Voltage Differential Signaling

MAC Medium Access Control

MOST Media Oriented System Transport

MP Main Profile

MPEG Moving Pictures Experts Group

MSE Mean Squared Error

MTU Maximum Transmission Unit

NAL Network Abstraction Layer

NE Network Element

OSI Open Systems Interconnection

PCB (Printed Circuit Board)

PDC Parking Distance Controller

PAL Phase Alternating Line

PLCP Physical Layer Convergence Procedure

PSNR Peak Signal to Noise Ratio

PTP Precision Time Protocol

PTT Protected Time Triggered

QoS Quality of Service

RS Redundant Slices

RSE Rear Seat Entertainment

RTP Real-time Transport Protocol

SIFS Short InterFrame Space

SNMP Simple Network Management Protocol

SP Simple Profile

SPQ Strict Priority Queuing

SSIM Structural SIMilarity index

STS Simple Traffic Smoother

TB Token Bucket

TBF Token Bucket Filter

TCP Transmission Control Protocol

TDMA Time Devision Multiple Access

TT Time Triggered

UDP User Datagram Protocol

UPnP Universal Plug and Play

UTT Unprotected Time Triggered

VBR Variable Bit Rate

VCL Video Coding Layer

VLAN Virtual Local Area Network

VLC Variable Length Code

VoIP Voice over Internet Protocol

VOP Video Object Plane

WFQ Weighted Fair Queuing

WLAN Wireless Local Area Network

8 Notation

Only notation changes are repeated in the following sections. Otherwise, the last definition of each symbol is valid.

F-ARIMA Video Model

x_i Normally distributed random variable

X A sequence of normally distributed random variables

ν Number of GoPs in a video sequence

n Number of frames in a GoP

$N(m, \sigma^2)$ Normal distribution with the mean value m and the variance σ^2

H Hurst parameter

B Rate of decay of the autocorrelation coefficients

r Autocorrelation function at GoP level

ω Frequency

K_t Knee: Transition point of the autocorrelation function from an exponentially decreasing function to a gradually decreasing function

g_i A GoP in a video sequence

G A sequence of GoPs

α Pareto shape parameter

β The lower bound of x

F Size of all frames in a video sequence

f Frame rate

a Number of P-frames in the GoP

b Number of B-frames in the GoP

T_{e2e} End-to-end delay

Network Dimensioning

a_k Arrival time of the k_{th} packet

f_k Departure time of the k_{th} packet

R Service rate

W Weight value of the weighted fair queuing mechanism

ε Precision factor in the weight function

Z Upper bound on the initial delay

C Store and forward delay

D Head-of-line blocking delay

Q Queue size

Traffic Shaping

$\bar{A}(\tau)$ Traffic envelope

p Peak rate

r Token regeneration rate

b Bucket size seen from the source shaper

b_{net} Bucket size seen from the network

T_{SI} Smoothing interval

T Frame period

T_N Network delay

T_S Shaper delay

B Burst capacity

β Normalized burst capacity

T_{burst} Normalized burst length

M MTU sized packet

R_A Available link capacity

B_i The amount of bytes shaped by STS at time instance i

$r_{x,max}$ The maximum STS transmission rate

Q_{RS} Reshaper queue size

R_L Limited input link capacity

Δ Interrupt cycle times

Video Compression

n Number of bits to represent the value of a pixel

N Dynamic range of the pixel values per component

R Object classification rate

O Correctly detected objects

L Number of recognizable objects within a video sequence

k Number of frames in an encoded video sequence

$\bar{D}_i$ Mean Deviation between the feature vectors in the compressed and uncompressed video sequences of a frame i

b Target bit rate

Q Codec quantizer value

A Appendix

A.1 Color Spaces in Image and Video Compression

There are different ways to represent colors in the digital domain. For computer displays, for instance, pixels represent the color values of the three additive primaries, red, green, and blue (RGB). However, the RGB representation does not directly take into account the human visual system, since the human eye is more sensitive to brightness than to the actual color information. By transforming the $R'G'B'$ (Gamma corrected[1] RGB) with Eq. (A.1), the $Y'C_bC_r$ color space is obtained with Y' representing the brightness, $C_b = B' - Y'$ and $C_r = R' - Y'$ representing the blue and red color difference values of the signal. Brightness is usually referred to as luma and the color difference components as chroma. Another widely used notation is YUV, where Y denotes the luma component and U and V denote the chroma components. The transformation from the Gamma corrected $R'G'B'$ to the $Y'C_bC_r$ space can be written as [63]:

$$\begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \cdot \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}. \tag{A.1}$$

Given sufficient arithmetic precision, this transformation can be reversed by

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix}^{-1} \cdot \begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 1 \\ 1 & -0.19421 & -0.50937 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} Y' \\ C_b \\ C_r \end{bmatrix}. \tag{A.2}$$

For the digital representation of color values, the number of bits preserved for the intensity value of a pixel is referred to as bit depth (bits per pixel). The typical number of bits for the intensity value in either color space representation is eight bits per component.

The information of the chroma components can furthermore be reduced by subsampling [27; 136]. The amount of subsampling is typically denoted by ratios of three integer numbers separated by colons. Four of the most common formats and notations are summarized in Table A.1. The leading 4 is inherited from historical reasons.

[1] Gamma correction is obtained by a non-linear transfer function according to the human perceptual response to brightness.

Notation	Chroma sampling resolutions	Bits/Pixel	Compression
4:4:4	full luma and chroma components sampling resolutions; no subsampling	24	—
4:2:2	chroma components subsampled by factor 2 in horizontal direction	16	1.5 : 1
4:2:0	chroma components subsampled by factor 2 both horizontally and vertically	12	2 : 1
4:1:1	chroma components subsampled by factor 4 in horizontal direction	12	2 : 1

Table A.1: Most common chroma subsampling resolutions and notations.

A.2 Jitter Calculation for CAN Packets

The jitter value of 122.88 μs results from Eq. (3.1). This is the jitter of the k_{th} packet, as defined in [83], where a_k and f_k are the arrival and departure times of the packet. By modifying it as

$$\text{Jitter}_{NE,k} = |(f_k - a_k) - (f_{k-1} - a_{k-1})| \tag{A.3}$$

it becomes clear that the jitter of the k_{th} packet is the difference between the time the k_{th} packet spent in this NE and the time the previous packet spent in this NE. By substituting all values in Eq. (3.1) to calculate the jitter for the last PT-CAN packet, i.e., the gray one in Figure 3.6, the jitter is computed to be

$$\text{Jitter}_{NE,grey} = |(130.24 - 125.12) - (122.88 - (-5.12))|\ \mu\text{s} = 122.88\,\mu\text{s}. \tag{A.4}$$

While the jitter for the black PT-CAN packet from Figure 3.6 is

$$\text{Jitter}_{NE,black} = |(125.12 - (-122.88)) - ((-5.12) - (-133.12))|\ \mu\text{s} = 122.88\,\mu\text{s}. \tag{A.5}$$

The jitter of the gray packet, which is larger has been taken into account for worst case analyses in Chapters 3 and 4.

List of Figures

List of Tables

Bibliography

Author's Publications

[1] Joachim Hillebrand, Mehrnoush Rahmani, Richard Bogenberger, and Eckehard Steinbach, "Coexistence of Time-Triggered and Event-Triggered Traffic in Switched Full-Duplex Ethernet Networks," in *Junior Researcher Workshop on Real-Time Computing*, Nancy, France, March 2007.

[2] Joachim Hillebrand, Mehrnoush Rahmani, Richard Bogenberger, Eckehard Steinbach, "Coexistence of Time-Triggered and Event-Triggered Traffic in Switched Full-Duplex Ethernet Networks," in *IEEE Second International Symposium on Industrial Embedded Systems - SIES'07*, Lisbon, Portugal, July 2007.

[3] Mehrnoush Rahmani, Andrea Pettiti, Ernst Biersack, Eckehard Steinbach, and Joachim Hillebrand, "A Comparative Study of Network Transport Protocols for In-Vehicle Media Streaming," in *IEEE International Conference on Multimedia & Expo*, Hannover, Germany, June 2008.

[4] Mehrnoush Rahmani, Bernd Mueller-Rathgeber, Wolfgang Hintermaier, and Eckehard Steinbach, "Error Detection Capabilities of Automotive Network Technologies and Ethernet - A Comparative Study," in *IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, June 2007.

[5] Mehrnoush Rahmani, Holger Kloess, Wolfgang Hintermaier, and Eckehard Steinbach, "Real-Time Video Compression for Driver Assistance Camera Systems," in *The First Annual International Symposium on Vehicular Computing Systems*, Trinity College Dublin, Ireland, July 2008.

[6] Mehrnoush Rahmani, Joachim Hillebrand, Wolfgang Hintermaier, Eckehard Steinbach, and Richard Bogenberger, "A Novel Network Architecture for In-Vehicle Audio and Video Communication," in *IEEE Broadband Convergence Networks Workshop*, Munich, Germany, May 2007.

[7] Mehrnoush Rahmani, Ktawut Tappayuthpijarn, Benjamin Krebs, Eckehard Steinbach, and Richard Bogenberger, "Traffic Shaping for Resource-Efficient In-Vehicle Communication," *IEEE Transactions on Industrial Informatics*, 2009, Accepted for publication.

[8] Mehrnoush Rahmani, Martin Pfannenstein, Eckehard Steinbach, Giuseppe Giordano, and Ernst Biersack, "Wireless Media Streaming in IP-based In-Vehicle Networks," in *IEEE Vehicular Networking and Applications Workshop*, Dresden, Germany, June 2009.

[9] Mehrnoush Rahmani, Rainer Steffen, Ktawut Tappayuthpijarn, Giuseppe Giordano, Eckehard Steinbach, and Richard Bogenberger, "Performance Analysis of Different Network Topologies for In-Vehicle Audio and Video Communication," in *4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks (QoS-IP)*, Venice, Italy, February 2008.

[10] Mehrnoush Rahmani, Wolfgang Hintermaier, Andreas Laika, Holger Endt, and Eckehard Steinbach, "A Novel Network Design for Future IP-based Driver Assistance Camera Systems," in *IEEE International Conference on Networking, Sensing and Control*, Okayama, Japan, March 2009.

[11] Rainer Steffen, Richard Bogenberger, Daniel Herrscher, Joachim Hillebrand, Wolfgang Hintermaier, Mehrnoush Rahmani, and Andreas Winckler, "Vision of a Future In-Car Network Architecture based on the Internet Protocol," *IC@ST Magazine*, November 2008. [Online]. Available: http://icast-magazine.org/?page=articles

[12] Rainer Steffen, Richard Bogenberger, Joachim Hillebrand, Wolfgang Hintermaier, Mehrnoush Rahmani, and Andreas Winckler, "Design and Realization of an IP-based In-car Network Architecture," in *The First Annual International Symposium on Vehicular Computing Systems*, Trinity College Dublin, Ireland, July 2008.

General Publications

[13] A contribution from Pioneer, Gibson, BEC Labs America, Micrel, Samsung Electronics, Independent, Cirrus Logic, HP, JGG, Broadcom, EchoStar, "Audio/Video Bridging Task Group," 2008. [Online]. Available: http://www.ieee802.org/1/pages/avbridges.html

[14] Actel Corporation, "Developing AFDX Solutions," 2005. [Online]. Available: http://www.actel.com/documents/AFDX_Solutions_AN.pdf

[15] Adam Dunkels, "Full TCP/IP for 8 bit architecture," in *First International Conference on Mobile Systems, Applications and Services (MobiSys)*, San Franscisco, CA, USA, May 2003.

[16] Aidan Williams, "AVB L3 Transport SG," April 2007. [Online]. Available: http://grouper.ieee.org/groups/1722/contributions/AVB%20L3%20Transport%20SG.pdf

[17] Alberto Sangiovanni-Vincentelli, "Automotive Electronics: Trends and Challenges," Detroit, MI, October 2000.

[18] Andreas Grzemba, *MOST - Das Multimedia-Bussystem für den Einsatz im Automobil*. FRANZIS, 2007, vol. 2.

[19] Andrew S. Tanenbaum, *Computer Networks*, forth ed. Pearson Education International, 2003, pp. 400–415.

[20] Arizona State University, Aalborg University, and acticom GmbH, "YUV video Sequences," 2006. [Online]. Available: http://trace.eas.asu.edu/yuv/index.html

[21] Arne Caspari, "unicap library," 2006. [Online]. Available: http://www.unicap-imaging.org

[22] AUTOSAR Organization, "Automotive Open System Architecture," AUTOSAR Consortium. [Online]. Available: http://www.autosar.org

[23] Bernd Girod, "What's wrong with mean-squared error?" *Digital Images and Human Vision, MIT Press*, pp. 207–220, 1993.

[24] Berthold K.P. Horn and Brian G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–204, August 1981.

[25] Borko Furht and Joshua Greenberg and Raymond Westwater, *Motion Estimation Algorithms for Video Compression*, ser. Kluwer International Series in Engineering and Computer Science. Boston, MA, USA: Kluwer Academic Publisher, 1996, vol. 379.

[26] Changcheng Huang, Michael Devetsikiotis, Ioannis Lambadaris and A. Roger Kaye, "Modeling and Simulation of Self-Similar Variable Bit Rate Compressed Video: A Unified Approach," *ACM SIGCOMM*, 1995.

[27] Charles A. Poynton, "Chroma Subsampling Notation," March 2002. [Online]. Available: http://www.poynton.com/PDFs/Chroma_subsampling_notation.pdf

[28] Chiharu Ishii, Yoshie Sudo and Hiroshi Hashimoto, "An Image Conversion Algorithm from Fish Eye Image to Perspective Image for Human Eyes," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1009–1014, 2003.

[29] Communications and Remote Sensing Lab (TELE) at Univeriste´ Catholique de Louvain, "OpenJPEG Library," 2007. [Online]. Available: http://www.openjpeg.org/index.php?menu=doc

[30] Condor Engineering, "AFDX / ARINC 664 Tutorial (1500-049)," 2005. [Online]. Available: http://www.condoreng.com/support/downloads/tutorials/AFDXTutorial.pdf

[31] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1048 – 1056, August 1995.

[32] Dave Olsen, "IEEE 1722 AVB L2 Transport Protocol," November 2007. [Online]. Available: http://ieee802.org/1/files/public/docs2007/avb-dolsen-1722-status-1107.pdf

[33] David Meyer and Friedrich Leisch and Kurt Hornik, "The support vector machine under test," in *Neurocomputing*, vol. 55, 2003, pp. 169–186.

[34] Dinesh C. Verma, Hui Zhang and Domenico Ferrari, "Delay Jitter Control for Real-Time Communication in a Packet Switching Network," *In Proceedings of Tricom'91*, 1991.

[35] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A Framework for QoS-based Routing in the Internet. Request for comments RFC 2386," Internet Engineering Task Force (IETF), Tech. Rep., August 1998.

[36] Elektrobit Automotive Software Business Segment, “Elektrobit Automotive Software Creates Wideband Access to Autos,” March 2007. [Online]. Available: http://www.elektrobit.com//index.php?id=597&news_id=96&archive=

[37] EtherCAT Technology Group, “EtherCAT - Ethernet Control Automation Technology, Publicly available Specification,” International Electrotechnical Commission (IEC), 2004. [Online]. Available: http://webstore.iec.ch/webstore/webstore.nsf/artnum/034392

[38] ETHERNET Powerlink Standardization Group, “Real-time Ethernet Powerlink (EPL) IEC PAS 62408 , Publicly available Specification Pre-Standard,” 2005. [Online]. Available: http://webstore.iec.ch/webstore/webstore.nsf/artnum/034404

[39] Ferdy Hanssen, Pierre G. Jansen, Hans Scholten, Sape Mullender, “RTnet: A Distributed Real-Time Protocol for Broadcast-Capable Networks,” *Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS/ICNS 2005)*, April 2005.

[40] Fernando Pereira and Paulo Nunes, “Levels for MPEG-4 Visual Profiles,” February 2006. [Online]. Available: http://www.m4if.org/resources/profiles/index.html

[41] FFmpeg Multimedia System, 2006. [Online]. Available: http://ffmpeg.mplayerhq.hu

[42] Francoise Simonot-Lion, “In Car Embedded Electronic Architectures: How to Ensure their Safety,” *5th IFAC International Conference on Fieldbus Systems and their Applications, Aveiro, Portugal*, July 2003. [Online]. Available: http://www.loria.fr/publications/2003/A03-R-294/A03-R-294.ps

[43] Fridtjof Stein, “Efficient Computation of Optical Flow Using the Census Transform,” in *Deutsche Arbeitsgemeinschaft für Mustererkennung, DAGM’04*, 2004.

[44] Gary J. Sullivan and Pankaj Topiwala and Ajay Luthra, “The H.264/AVC Advanced Video Coding standard: Overview and introduction to the Fidelity Range Extensions,” in *SPIE Conference on Applications of Digital Image Processing XXVII*, A. G. Tescher, Ed., vol. 5558, Denver, CO, USA, August 2004, pp. 454–474.

[45] Gary J. Sullivan and Thomas Wiegand, “Video Compression—From Concepts to the H.264/AVC Standard,” in *Special Issue on Advances in Video Coding and Delivery*, ser. Proceedings of the IEEE, vol. 93, January 2005, pp. 18–31, invited Paper. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1369695

[46] Gregory K. Wallace, “The JPEG Still Picture Compression Standard,” *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, April 1991.

[47] Hans Weibel, Dominic Bechaz, “IEEE 1588 Implementation and Performance of Time Stamping Techniques,” *Conference on IEEE 1588*, September 2004. [Online]. Available: https://home.zhaw.ch/~wei/IEEE1588/PTP_Timestamping_Methods_Paper.pdf

[48] Hans-Georg Frischkorn, *Automotive Architecture Requirements*, BMW internal presentation, September 2005.

[49] Hermann Kopetz, Astrit Ademaj, Peter Grillinger and Klaus Steinhammer, “The Time-Triggered Ethernet (TTE) Design,” in *Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC’05)*, Seattle, Washington, USA, May 2005.

[50] Iain E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. West Sussex, England: John Wiley & Sons, Ltd., 2003. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1709988

[51] IEEE 802.3 Ethernet Working Group, “IEEE 802.3 CSMA/CD (ETHERNET),” 2008. [Online]. Available: http://www.ieee802.org/3/

[52] IEEE Microprocessor Standards Zone, “IEEE 1394 Overview,” 2006. [Online]. Available: http://standards.ieee.org/micro/1394overview.html

[53] IEEE Project 802, “IEEE 802.1p: Supplement to MAC Bridges: Traffic Calss Expediting and Dynamic Multicast Filtering,” Incorp. in IEEE Standard 802.1D, Part 3: Media Access Control (MAC) Bridges, 1998.

[54] IETF Zeroconf Working Group, “Zero Configuration Networking (Zeroconf),” 2003. [Online]. Available: http://www.zeroconf.org/

[55] Ingrid Daubechies, “The Wavelet Transform, Time-Frequency Localization and Signal Analysis,” in *IEEE Transactions on information theory*, vol. 36, no. 5, September 1990, pp. 961–1005.

[56] Institute of Electrical and Electronics Engineers, *Information Technology- Telecommunications and Information Exchange between Systems- Local and Metropolitan Area Networks- Specific Requirements- Part 3: Media Access Control (MAC) Bridges*, ANSI/IEEE Std 802.1D, 1998.

[57] ISO/IEC IS 10918–1, *Digital Compression and Coding of Continuous-Tone Still Images – Requirements and Guidelines*, Geneva, Switzerland, September 1992, also CCITT/ITU-T Recommendation T.81.

[58] ISO/IEC IS 13818–2, *Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video*, ITU-T Recommendation H.262 – ISO/IEC 13818-2 (MPEG-2), Geneva, Switzerland, November 1994, also ITU-T Recommendation H.262. (latest amendment: 2007). [Online]. Available: http://www.itu.int/rec/T-REC-H.262/en

[59] ISO/IEC IS 14496–10, *Coding of Audio visual Objects—Part 10: Advanced Video Coding*, Geneva, Switzerland, May 2003, also ITU-T Recommendation H.264. (vers. 2: 2004, vers. 3: 2005). [Online]. Available: http://www.itu.int/rec/T-REC-H.264/en

[60] ISO/IEC IS 14496–2, *Generic Coding of Audio-Visual Objects—Part 2: Visual*, Geneva, Switzerland, April 1999, (vers. 2: 2000, amendments: 2001,2002).

[61] ISO/IEC IS 15444–1, *JPEG 2000 Image Coding System—Part 1: Core Coding System*, Geneva, Switzerland, 2000, also ITU-T Recommendation T.800.

[62] ISO/IEC IS 15444–3, *JPEG 2000 Image Coding System—Part 3: Motion JPEG 2000*, Geneva, Switzerland, 2002.

[63] ITU-R Recommendation BT.601–5, *Encoding Parameters of Digital Television for Studios*, Recommendations of the ITU, Radiocommunication Assembly, 1995.

[64] Iwao Sasase, Alex Fung, "Hybrid local recovery scheme for reliable multicast using group-aided multicast scheme," *IEEE Wireless Communications and Networking Conference (WCNC), Hong Kong*, pp. 3478–3482, March 2007.

[65] J. Will Specks and Antal Rajnak, "LIN - Protocol, Development Tools, and Software Interfaces for Local Interconnect Networks in Vehicles," in *9th International Conference on Electronic Systems for Vehicles*, Baden-Baden, 2000. [Online]. Available: http://www.lin-subbus.org/frontend/kunden-pdf/w_specks_lin_baden-baden_paper.pdf#search=%22LIN%20-%20Protocol%2C%20Development%20Tools%2C%20and%20Software%20Interfaces%20for%20Local%20%22

[66] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt, "Performance of Optical Flow Techniques," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition'92*, June 1992, pp. 236–242.

[67] Jan Kiszka et al, "RTnet - A flexible Hard Real-Time Networking Framework," *10th IEEE Conference on Emerging Technologies and Factory Automation 2005, EFTA, Catania, Italy*, September 2005. [Online]. Available: www.rts.uni-hannover.de/rtnet/download/RTnet-ETFA05.pdf

[68] Jean-Philipppe Georges, Thierry Divoux and Eric Rondeau, "Strict Priority versus Weighted Fair Qeueing in Switched Ethernet Networks for Time Critical Applications," in *IEEE International Parallel and Distributed Processing Symposium*. Denver, Colorado, USA: Centre de Recherche en Automatique de Nancy (CNRS UMR 7039), April 2005.

[69] Jean-Yves Le Boudec, "Application of Network Calculus to Guaranteed Service Networks," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 1087–1096, 1998.

[70] Jeff Bier, "Introduction to video compression," *Video/Imaging Design Line*, April 2006. [Online]. Available: http://www.videsignline.com/howto/185301351

[71] Jeremiah Golston and Ajit Rao, "Video Compression: System Trade-Offs with H.264, VC-1 and Other Advanced CODECs," Dallas, TX, USA, August 2006. [Online]. Available: http://focus.ti.com/lit/ml/spry088/spry088.pdf

[72] John D. McCarthy and M. Angela Sasse and Dimitrios Miras, "Sharp or Smooth? Comparing the Effects of Quantization vs. Frame Rate for Streamed Video," in *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '04)*. New York, NY, USA: ACM Press, 2004, pp. 535–542.

[73] Jong-Suk Ahn, Seung-Wook Hong and John Heidemann, "An Adaptive FEC Code Control Algorithm for Mobile Wireless Sensor Networks," *Journal of Communications and Networks*, vol. 7, no. 4, pp. 489–499, 2005. [Online]. Available: http://www.isi.edu/~johnh/PAPERS/Ahn05a.html

[74] Jork Loeser, "Low-Latency Hard Real-Time Communication over Switched Ethernet," Dissertation, Technische Universitaet Dresden, Informatic Department, Dresden, Germany, January 2006. [Online]. Available: http://os.inf.tu-dresden.de/papers_ps/loeser-phd.pdf

[75] JPEG Committee and Elysium, Ltd. and 2KAN, "What is the Patent Situation with JPEG 2000?" 2006. [Online]. Available: http://www.jpeg.org/faq.phtml?action=show_answer&question_id=q3f042a68b1081

[76] Jérôme Grieu, "Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques," These de doctorat, Institut National Polytechnique de Toulouse, Toulouse, France, September 2004. [Online]. Available: http://ethesis.inp-toulouse.fr/archive/00000084/01/grieu.pdf

[77] Jörn Ostermann and Jan Bormans and Peter List and Detlev Marpe and Matthias Narroschke and Fernando Pereira and Thomas Stockhammer and Thomas Wedi, "Video Coding with H.264/AVC: Tools, Performance, and Complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, 2004. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1286980

[78] K. Onthriar and K. K. Loo and Z. Xue, "Performance Comparison of Emerging Dirac Video Codec with H.264/AVC," *International Conference on Digital Telecommunications (ICDT '06), Cote d'Azur, France*, p. 22, August 2006. [Online]. Available: https://docweb.lrz-muenchen.de/cgi-bin/doc/nph-webdoc.cgi/000110A/http/ieeexplore.ieee.org/search/srchabstract.jsp=3farnumber=3d1698469&isnumber=3d35811&punumber=3d11153&k2dockey=3d1698469@ieeecnfs&query=3d=2528=2528performance+comparison+of+emerging+dirac+video+codec=2529=253Cin=253Emetadata=2529&pos=3d0

[79] K. R. Rao and Patrick Yip, *Discrete cosine transform: algorithms, advantages, applications*. San Diego, CA, USA: Academic Press Professional, Inc., 1990.

[80] Kelson R. T. Aires and Andre M. Santana and Adelardo A. D. Medeiros, "Optical Flow using Color Information: Premliminary results," in *23rd ACM Symposium on Applied Computing*, March 2008.

[81] Laurent Boch, *MPEG-2 Profiles and Levels*, ISO/IEC, Geneva, Switzerland, July 1996. [Online]. Available: http://viswiz.gmd.de/DVP/Public/deliv/deliv.211/mpeg/pr@lv01.htm

[82] Leonard Kleinrock, "Queueing Systems: Computer Applications," *John Wiley and Sons, New York*, vol. 2, 1975.

[83] Leonidas Georgiadis, Roch Guérin, Vinod G. J. Peris and Rajendran Rajan, "Efficient Support of Delay and Rate Guarantees in an Internet," *ACM SIGCOMM*, vol. 26, pp. 106–116, New York, NY, USA, October 1996.

[84] Leonidas Georgiadis, Roch Guérin, Vinod G. J. Peris, Kumar N. Sivarajan, "Efficient Network QoS Provisioning Based on per Node Traffic Shaping," *IEEE/ACM Transactions on Networking*, vol. 4, no. 4, pp. 482–501, 1996.

[85] Lothar Pantel and Lars C. Wolf, “On the Impact of Delay on Real-Time Multiplayer Games,” in *NOSSDAV ’02: Proceedings of the 12th international workshop on network and operating systems support for digital audio and video*. New York, NY, USA: ACM Press, 2002, pp. 23–29.

[86] Mark W. Garrett, Walter Willinger, “Analysis, Modeling and Generation of Self-Similar VBR Video Traffic,” *ACM SIGCOMM*, 1994.

[87] Max Azarov, SMSC, “Approach to a Latency-Bound Ethernet,” IEEE 802.1 AVB group, 2006. [Online]. Available: http://www.ieee802.org/1/files/public/docs2006/avb-azarov-ApproachToLatencyBoundEthernet-060130.pdf

[88] Michael Horowitz and Anthony Joch and Faouzi Kossentini and Antti Hallapuro, “H.264/AVC baseline profile decoder complexity analysis,” vol. 13, no. 7, pp. 704–716, July 2003. [Online]. Available: http://ieeexplore.ieee.org/iel5/76/27384/01218201.pdf?tp=&isnumber=27384&arnumber=1218201

[89] Mohammad F. Alam, Mohammad Atiquzzaman, Mohammad A. Karim, “Traffic Shaping for MPEG Video Transmission over the Next Generation Internet,” *Computer Communications*, vol. 23, no. 14, pp. 1336–1348, August 2000.

[90] MOST Cooperation, “Gut unterhalten waehrend der Fahrt.” [Online]. Available: http://www.eue24.net/pi/index.php?StoryID=253&articleID=124534

[91] National Semiconductor, “News release,” May 2002. [Online]. Available: http://www.national.com/news/item/0,1735,759,00.html

[92] Navneet Dalal and Bill Triggs, “Histograms of Oriented Gradients for Human Detection,” *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA*, pp. 886–893, 2005.

[93] Nicolas Navet et al, “Trends in Automotive Communication Systems,” *Proceedings of IEEE*, vol. 93, no. 6, June 2005. [Online]. Available: http://www.loria.fr/~nnavet/publi/IEEE-Proc05.pdf

[94] Omar Javed and Mubarak Shah, “Tracking and Object Classification for Automated Surveillance,” *The seventh European Conference on Computer Vision, Copenhagen, Denmark*, May 2002.

[95] OMG (Object Management Group), “Smart Transducer Specification TTP/A,” 2002. [Online]. Available: ftp://ftp.omg.org/pub/docs/orbos/01-10-02.pdf

[96] OMNeT++ Community, “Omnet++ simulator.” [Online]. Available: http://www.omnetpp.org/

[97] On2 Technologies, “VP3 codec,” 2006. [Online]. Available: http://www.on2.com

[98] OVDA (Open DeviceNet Vendor Association), “EtherNet/IP.” [Online]. Available: http://www.ethernetip.de

[99] Pavel Korshunov, Wei Tsang Ooi, “Critical Video Quality for Distributed Automated Video Surveillance,” *ACM Multimedia*, November 2005.

[100] Peter M. Kuhn, *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation*. Norwell, MA, USA: Kluwer Academic Publishers, 1999.

[101] Peter Symes, *Digital Video Compression*. New York, NY, USA: The McGraw-Hill Companies, 2004. [Online]. Available: http://www.symes.tv

[102] Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale, "Real Time Application Interface," 2005. [Online]. Available: http://www.rtai.org

[103] R. Jain, "The Art of Computer Systems Performance Analysis," ISBN 0-471-50 336-34B4A5A, New York: Wiley, 1991.

[104] Ralf Steinmetz, "Human Perception of Jitter and Media Synchronization," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 1, pp. 61–72, January 1996.

[105] Rob Koenen, editor, "Overview of the MPEG-4 Standard (ISO/IEC 14496)," ISO/IEC, Geneva, Switzerland, March 2002. [Online]. Available: http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm

[106] Robert Bosch GmbH, "Controller Area Network," Robert Bosch GmbH. [Online]. Available: http://www.semiconductors.bosch.de/de/20/can/index.asp

[107] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, *An Architecture for Differentiated Services*, RFC 2475 ed., IETF, 1998.

[108] Sanaa Sharafeddine, "IP Network Planning for Realtime Services with Statistical QoS Guarantees," Dissertation, Technische Universität München, Munich, Germany, January 2005. [Online]. Available: http://deposit.ddb.de/cgi-bin/dokserv?idn=977700828

[109] Shih-Fu Chang and T. Sikora and A. Purl, "Overview of the MPEG-7 Standard," in *IEEE Transactions on Circuits And Systems for Video Technology*, vol. 11, no. 6, June 2001, pp. 696–702.

[110] Siemens AG, "IEC/PAS 62407 Real-time Ethernet PROFINET IO," International Electrotechnical Commission (IEC), 2005. [Online]. Available: http://webstore.iec.ch/webstore/webstore.nsf/artnum/034395

[111] Sorin Cocorada, "Experimental Evaluation of the Throughput of Unacknowledged UDP Traffic in an IEEE 802.11E/G WLAN," *RECENT*, vol. 8, no. 1, pp. 15–21, March 2007.

[112] ST Microelectronics, "User manual UM0469," November 2007.

[113] Stephan Wenger, "H.264/AVC over IP," vol. 13, no. 7, pp. 645–656, July 2003. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1218197

[114] Subhasis Saha, "Image Compression—from DCT to Wavelets: A Review," *Crossroads – The ACM Student Magazine, New York, NY, USA*, vol. 6, no. 3, pp. 12–21, 2000. [Online]. Available: http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html

[115] Syed Ali Khayam, "The Discrete Cosine Transform (DCT): Theory and Application," March 2003. [Online]. Available: http://www.egr.msu.edu/waves/people/Ali_files/DCT_TR802.pdf

[116] Test Sequences, 2007. [Online]. Available: ftp://ftp.ldv.e-technik.tu-muenchen.de/pub/test_sequences/601

[117] The Imaging Source, “Products,” 2006. [Online]. Available: http://www.theimagingsource.com/en/products

[118] Thomas A. Carlson and Hinze Hogendoorn and Frans A. J. Verstraten, “The Speed of Visual Attention: What Time is It?” *Journal of Vision*, vol. 6, no. 12, pp. 1406–1411, December 2006. [Online]. Available: http://journalofvision.org/6/12/6/

[119] Thomas Wiegand, Heiko Schwarz, Anthony Joch, Faouzi Kossentini, Gary J. Sullivan, “Rate-Constrained Coder Control and Comparison of Video Coding Standards,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688–703, July 2003. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1218200

[120] Tim Borer and Thomas Davies, “Dirac–Video Compression Using Open Standards,” BBC Research & Development, White Paper WHP 117, July 2005.

[121] Tinku Acharya and Ping-Sing Tsai, *JPEG2000 Standard for Image Compression*. Wiley-Interscience, John Wiley & Sons, Inc.,Hoboken, New Jersey, USA, 2005.

[122] Tor Skeie, Svein Johannessen, and Oyvind Holmeide, “Timeliness of Real-Time IP Communication in Switched Industrial Ethernet Networks,” *IEEE Transactions on Industrial Informatics*, vol. 2, no. 1, pp. 25–39, Februar 2006.

[123] Trista Pei-chun Chen and Tsuhan Chen, “Second-Generation Error Concealment for Video Transport over Error-Prone Channels,” *Wireless Communications and Mobile Computing*, vol. 2, no. 6, pp. 607–624, October 2002.

[124] TU Berlin, “MPEG-4 and H.263 Video Traces for Network Performance Evaluation.” [Online]. Available: http://www.tkn.tu-berlin.de/research/trace/trace.html

[125] P. N. Tudor, “MPEG-2 Video Compression,” December 1999. [Online]. Available: http://www.bbc.co.uk/rd/pubs/papers/paper_14/paper_14.shtml

[126] Via Licensing Corp., “Via Licensing,” 2006. [Online]. Available: http://www.vialicensing.com

[127] VideoLAN Project, “x264,” 2006. [Online]. Available: http://www.videolan.org/developers/x264.html

[128] Vincenzo Santoro, “Analysis and Implementation of Forward Error Correction Mechanisms for Multimedia Streaming in Wireless Networks,” Master’s thesis, Eurecom in co-operation with BMW Group Research and Technology, 2007.

[129] Werner Zimmermann, Ralf Schimdgall, *Bussysteme in der Fahrzeugtechnik*, Reinhard Dapper, Ed. Vieweg, 2006.

[130] Xiph.org Foundation., “Xiph.Org,” 2006. [Online]. Available: http://www.xiph.org

[131] Xiph.org Foundation, “Theora I Specification,” March 2006, (latest rev. Dec. 2006). [Online]. Available: http://www.theora.org

[132] Youngsoo Kim, Sunghyun Choi, Kyunghun Jang, and Hyosun Hwang, “Throughput Enhancement of IEEE 802.11 WLAN via Frame Aggregation,” *60th Vehicular Technology Conference (VTC)-Fall*, vol. 4, pp. 3030–3034, Los Angeles, CA, USA, September 2004.

[133] Yukihiro Arai and Masayuki Nakajima and Takeshi Agui, “A Fast DCT-SQ Scheme for Images,” *Transactions of the IEICE*, vol. E71, no. 11, pp. 1095–1097, November 1988. [Online]. Available: http://search.ieice.org/bin/summary.php?id=e71-e_11_1095&category=E&year=1988&lang=&abst=

[134] Zhou Wang and Alan C. Bovik and Ligang Lu, “Why is Image Quality Assessment so Difficult?” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ‘02)*, vol. 4, pp. 3313–3316, 2002. [Online]. Available: http://ieeexplore.ieee.org/xpl/absfree.jsp?arNumber=1004620

[135] Zhou Wang and Ligang Lu and Alan C. Bovik, “Video Quality Assessment based on Structural Distortion Measurement,” *Signal Processing: Image Communication, special issue on objective video quality metrics*, vol. 19, no. 2, pp. 121–132, 2004. [Online]. Available: http://www.cns.nyu.edu/~zwang/files/papers/vssim.pdf

[136] Charles A. Poynton, Ed., *A Technical Introduction to Digital Video*. New York, NY, USA: John Wiley & Sons, Inc., 1996.

[137] P. T. Jean-Yves Le Boudec, Ed., *Network Calculus - A Theory of Deterministic Queuing Systems for the Internet*. Springer Verlag - LNCS 2050, May 2004.

[138] “MPEG Licensing Authority, LLC,” 2006. [Online]. Available: http://www.mpegla.com

[139] *FlexRay, Requirements Specification, Version 2.1*, FlexRay Consortium, 2005.

[140] “IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements,” *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003))*, pp. 1–189, 2005.

[141] *MOST Media Oriented System Transport, Multimedia and Control Networking Technology, Rev. 2.4*, OASIS Silicon Systems, 2005. [Online]. Available: www.mostcooperation.com/downloads/Specifications

[142] *Residential Ethernet (RE) (a working paper)*, Draft 0.142 ed., A contribution from Pioneer, Gibson, BEC Labs America, Micrel, Samsung Electronics, Independent, Cirrus Logic, HP, JGG, Broadcom, EchoStar, November 2005.

[143] *Local Interconnect Network (LIN) Specification package*, Revision 2.0 ed., LIN Consortium, 2003. [Online]. Available: http://www.lin-subbus.org

[144] *CAN Specification, Version 2.0*, online, Robert Bosch GmbH, 1991, online. [Online]. Available: http://www.semiconductors.bosch.de/pdf/can2spec.pdf